# Socio-inspired Evolutionary Algorithms: A Unified Framework and Survey

**Laxmikant Sharma** ( lkrdkrishnan@gmail.com )
Dayalbagh Educational Institute Faculty of Science https://orcid.org/0000-0002-7597-5334
**Vasantha Lakshmi Chellapilla**
Dayalbagh Educational Institute Faculty of Science
**Patvardhan Chellapilla**
Dayalbagh Educational Institute Faculty of Engineering

**Research Article**

**Keywords:** Evolutionary Computation, Socio-inspired Evolutionary Algorithms, Socio-inspired Metaheuristic Algorithms, Soft Computing

**Posted Date:** September 27th, 2022

**DOI:** https://doi.org/10.21203/rs.3.rs-1665166/v1

# Socio-inspired Evolutionary Algorithms: A Unified Framework and Survey

Laxmikant Sharma[1*], Vasantha Lakshmi Chellapilla[1] and Patvardhan Chellapilla[2]

[1*]Department of Physics and Computer Science, Dayalbagh Educational Institute, Dayalbagh, Agra, 282005, Uttar Pradesh, India.

[2]Department of Electrical Engineering, Dayalbagh Educational Institute, Dayalbagh, Agra, 282005, Uttar Pradesh, India.

*Corresponding author(s). E-mail(s): lkrdkrishnan@gmail.com;
Contributing authors: vasanthalakshmi@dei.ac.in; cpatvardhan@dei.ac.in;

**Abstract**

Evolutionary computation has gone through vast and diverse research endeavors in the past few decades. Although the initial inspiration came from Darwin's ideas of biological evolution, the field has moved thereon to ideas from the collective intelligence of insects, birds, and fish, to name a few. A variety of algorithms have been proposed in the literature based on these ideas and have been shown to perform well in various applications. More recently, inspiration from human behavior and knowledge exchange and transformation has given rise to a new evolutionary computing paradigm. It is well recognized that human societies and problem-solving capabilities have evolved much faster than biological evolution. Many research endeavors have been reported in the literature inspired by diverse aspects of human societies with corresponding terminologies to describe the algorithm. These endeavors have resulted in a plethora of algorithms worded differently from each other, but the underlying mechanisms could be more or less similar, causing immense confusion for a new reader. This paper presents a generalized framework for these Socio-inspired Evolutionary Algorithms (SIEAs) or Socio-inspired Metaheuristic 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 of each SIEA enables 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. In that sense, 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 Metaheuristic 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 techniques constitute 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 non-linear, multi-modal, vast, dynamic, and complex, exact algorithms are not practically feasible to solve them. Heuristics and metaheuristics are typically employed in such cases due to their user-friendly implementation nature and schematic algorithmic framework and design. Heuristics are designed for a specific problem, whereas metaheuristics have broad applicability.

Several metaheuristics are nature-inspired algorithms. Evolutionary Algorithms (EAs) are one of the leading metaheuristics. 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 that are not well adapted to their environment do not survive long enough to reproduce or have fewer chances to reproduce than other 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 metaheuristic 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.

Fraser (1958) made some early contributions to this field. However, simulations of evolution using evolutionary algorithms and artificial life picked up with the work of Barricelli (1962). Artificial evolution became a widely recognized optimization method. Rechenberg (1970) used evolution strategies in solving complex engineering problems, and Holland (1992) popularized genetic algorithms. Increasing computer power has enabled practical applications, including the automatic evolution of computer programs (Koza, 1992). EAs now can solve multi-dimensional problems more efficiently than software produced by human designers and can optimize systems' design (Onwubolu & Babu, 2004). EAs have been successfully utilized to solve problems in various domains, including planning, design, simulation and identification, control, and classification (Beasley, 2000).

Akin to EAs, there are other classes of metaheuristics. 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 & Eberhart, 1995), Ant Colony Optimization (ACO) (Dorigo & Di Caro, 1999), and Artificial Bee Colony (ABC) optimization (Karaboga & Basturk, 2007) are some notable swarm-based algorithms. Physics-based algorithms and Chemistry-based algorithms are inspired by phenomena seen in Physical and Chemical Sciences, respectively (Biswas, Mishra, Tiwari, & Misra, 2013; Siddique & Adeli, 2017). Some good survey articles on bio-inspired metaheuristics 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, Moscato, 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, Gordon, & Mathias, 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 & 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 a belief space was introduced to exchange experiences of the candidate solutions. This cultural framework has been improved and hybridized many times. Some conspicuous algorithms inspired from this framework are Cultural Algorithm with Evolutionary Programming (Chung & Reynolds, 1998), Cultural Swarms (Iacoban, Reynolds, & Brewster, 2003), Cultured Differential Evolution (Becerra & Coello, 2006), Social Learning Optimization (Z.-Z. Liu, Chu, Song, Xue, & Lu, 2016) and others.

It is well-accepted that human societies and humans' problem-solving capabilities have evolved much faster than biological evolution (Bjorklund, Causey, & Periss, 2010). This evolution is due to humans' biological structure and social means of living and surviving. DNA studies indicate that a newborn child is not genetically much different from his ancestors that were born 40,000 years ago (Varki & Altheide, 2005), i.e., human's hardware part is substantially the same. However, socially, humans have left the caves and are now 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 means of living and surviving 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 metaheuristics dubbed Socio-inspired Evolutionary Algorithms (SIEAs) or Socio-inspired Metaheuristics (see Figure 1). Many research endeavors have been reported in the literature claiming inspiration from different social phenomena seen in human societies. These endeavors have resulted in a plethora of algorithms worded differently from each other, but 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. Furthermore, Khuat and Thi My Hanh (2016), and Kumar and Kulkarni (2019) surveyed many SIEAs. Khuat and Thi My Hanh (2016) provided a limited survey without describing several powerful SIEAs. In comparison, Kumar and Kulkarni (2019) provided independent detailed descriptions of each algorithm. However, these previously published studies are limited to individual surveys without any effort to put reported SIEAs in a common framework.

This paper proposes a common framework for describing the prominent SIEAs. Also, a survey on the state-of-the-art SIEAs is provided. For each SIEA, the pseudocode description provided indicates points of similarity and distinctiveness vis-à-vis 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 those algorithms. This description cuts through the diverse terminology of each algorithm to present it in an easily comprehensible manner.

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—section 2—presents an introduction to social phenomena and division of SIEAs according to well-known social phenomena classification. In section 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, section 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

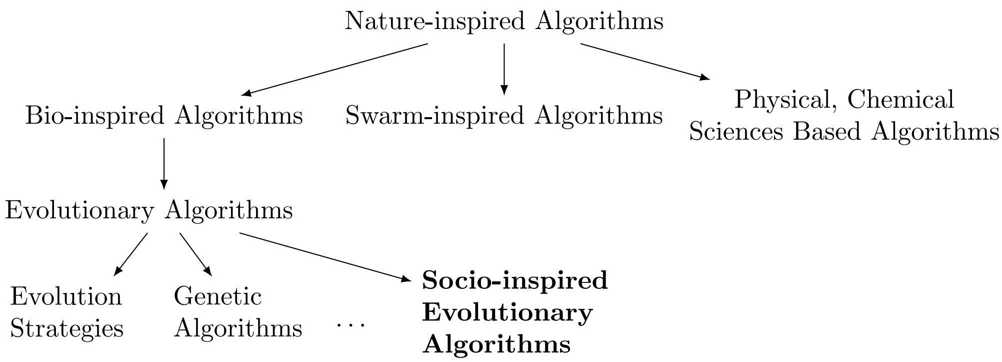


**Fig. 1** Taxonomy of SIEAs

and serial. These societies are built on the conventions of living in peace and working to fulfill everyone's need for food, clothes, and home. These basic needs of a human being are significantly achieved with various social activities or social actions. Interactions in humans lead to transfer of information. This information is usually utilized to generalize facts and for the benefit of society. There is no overarching optimizing principle seen in social interactions; however, they lead to robust society structures.

In order to improve their lifestyle, humans observe other similar or distinct species and change their behavior and actions accordingly. These observations also tend to influence people to do numerous actions of interest. Furthermore, humans self-observe their actions and try to improve in 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 & Hernández, 2009). Other examples of common social phenomena include collective behavior, self-organizing systems, social interaction, social coordination, social 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 some common phenomena in each of these categories.

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

# 3 Framework for SIEAs

Exploration and exploitation are two fundamental operations in optimization metaheuristics. 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. The term diversification is also used fro 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 metaheuristic (Bäck, Fogel, & Michalewicz, 1997). These operations are typically performed by selection, crossover, and mutation-like operators. It is not within this paper's scope to discuss the correspondence between the operator and its operation. Please refer Črepinšek, Liu, and Mernik (2013); Cuevas, Diaz, and Camarena (2021); Eiben and Schippers (1998) for details.

The essential idea in SIEAs is to divide the population into subpopulations, each

**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, 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 |

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

| SIEA | Social phenomena |
|---|---|
| **Cultural** | |
| Society and Civilization Algorithm (Ray & Liew, 2003) | Social interaction & cooperation |
| Soccer League Competition Algorithm (Moosavian, Roodsari, et al., 2013) | Social competition; Sports |
| Socio Evolution & Learning Optimization (Kumar, Kulkarni, & Satapathy, 2018) | Social learning |
| Nomadic People Optimizer (Salih & Alsewari, 2020) | Search for livable environment |
| **Economic** | |
| Imperialist Competitive Algorithm (Atashpaz-Gargari & Lucas, 2007b) | Imperialistic competition |
| **Moral and Juristic** | |
| — | — |
| **Political** | |
| Parliamentary Optimization (Borji, 2007; Borji & Hamidi, 2009) | Political competition |
| Group Leaders Optimization (Daskin & Kais, 2011b) | Self-existence of the group; Political influence |
| Election Campaign Optimization (Lv et al., 2010) | Political influence |
| Election Algorithm (H. Emami & Derakhshan, 2015) | Politics |
| Ideology Algorithm (Huan, Kulkarni, Kanesan, Huang, & Abraham, 2017) | Social competition |
| Political Optimizer (Askari, Younas, & Saeed, 2020) | Politics |

subpopulation searching in a different search sub-space, while at the same time maintaining the number of individuals in subpopulations commensurate with the importance of the search sub-space. In the beginning, 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 gradually 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 sub-classes/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 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 Figure 2 and Algorithm 1.

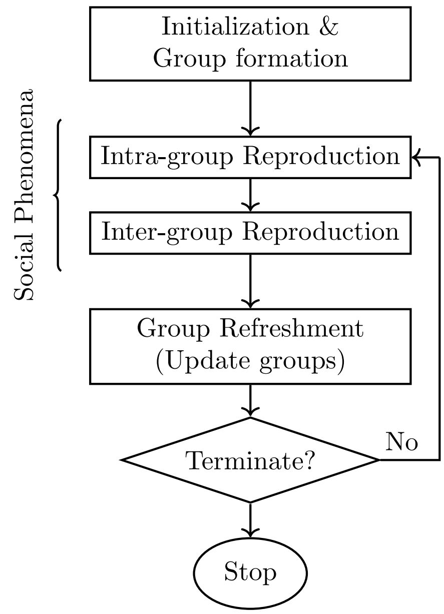


**Fig. 2** Flowchart of a generic SIEA

**Algorithm 1** Pseudocode of a generic SIEA

```
Initialize parameters and randomly generate
population of candidate solutions
Create logical groups of solutions
while termination condition not satisfied do
    (a) Intra-group reproduction
    (b) Inter-group reproduction
    Update groups
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., $N = \{I_1, I_2, \ldots, I_n\}$, 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., $\{I_1, I_2, \ldots, I_l\}$ in the first group, next $\{I_{l+1}, I_{l+2}, \ldots, I_{2l}\}$ individuals in the second group, $\ldots$, and last $\{I_{(m-1)l+1}, I_{(m-1)l+2}, \ldots, I_{ml}\}$ individuals in the $m^{\text{th}}$ group. The whole population of individuals can also be sorted according to their fitness prior to grouping to generate groups with phenotypically similar individuals.

   (B) Create subpopulations by selecting $l$ individuals randomly for the first group, and then selecting $l$ individuals randomly from the remaining population for the next group, and so on.

(ii) Generate $n$ individuals randomly i.e., $N = \{I_1, I_2, \ldots, I_n\}$. Evaluate each population individual $I_i$ according to the objective function. Select the best $m$ individuals to be the leaders of respective $m$ groups $\{M_1, M_2, \ldots, M_m\}$. 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 the corresponding fitness of the leaders, i.e., the number of individuals in a group $M_i$ equals $round\left\{\left|\frac{f(L_i)}{\sum f(L_i)}\right| \cdot (n-m)\right\}$, where $f(L_i)$ represents $i^{\text{th}}$ leader's fitness. Hence, a better leader will lead a bigger group possessing more individuals. 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 $\{M_1, M_2, \ldots, M_m\}$. 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 the corresponding fitness of the leaders, i.e., the number of individuals in a group $M_i$ equals $round\left\{\left|\frac{f(L_i)}{\sum f(L_i)}\right| \cdot (n-m)\right\}$.

(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 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 $\{M_1, M_2, \ldots, M_m\}$. Sequentially or randomly select an individual $I_i \in \{I_1, I_2, \ldots, I_{n-m}\}$ 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., $C_g \leftarrow I_i;\ g = 1, \ldots, k$ from which it has minimum Euclidean distance than from the other clusters, e.g., $D(I_i, C_g) = \min(D(I_i, C_1), \ldots, D(I_i, C_k))$. 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., $\max(D(I_i, C_1), \ldots, D(I_i, C_k)) > \text{avg}(D(I_i, C_1), \ldots, D(I_i, C_k))$.

The random selection of individuals while creating groups can enhance diversification. However, sequential selection and distance-based clustering methods can provide better intensification. The fitness-proportionate method while grouping 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, viz., 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.

(i) Individuals are evaluated (typically, to determine the leaders) and then grouping is performed.

(ii) 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.

(i) 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.

(ii) Groups are created using some distance-based method, and then the best of the evaluated individuals are announced as leaders.

(iii) Leaders are generated randomly in the initial phase of an algorithm, and then group members 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.

2. **Reproduction Mechanisms**: A reproductive mechanism used in socially motivated systems depicts the social convention of individual evolution. Mainly, imitation and the transfer of mannerism 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 for evolving themselves. Similarly, the global best individual influences

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

| # | Grouping Mechanism | | | Intensification | Diversification |
|---|---|---|---|---|---|
| | **Leader selection** | **Individual selection** | **Member count** | | |
| 1(a) | after grouping | sequential | equal | low | high |
| 1(b) | after grouping | random | equal | low | high |
| 2(a) | prior grouping | sequential | equal | low | moderate |
| 2(b) | prior grouping | sequential | unequal, proportionate | low | moderate |
| 3(a) | prior grouping | random | equal | low | high |
| 3(b) | prior grouping | random | unequal, proportionate | low | high |
| 4(a) | generate leaders | generate individuals | equal | high | low |
| 4(b) | generate leaders | generate individuals | unequal, proportionate | high | low |
| 5 | prior grouping | sequential/random | unequal, distance-based | high | low |
| 6 | after grouping (clustering) | sequential/random | unequal, distance-based | high | low |

the individual group leaders to follow him. Moreover, individuals can follow other better peers or use some self-improvement methods alongside. These reproductive mechanisms typically stipulate asexual versions of various recombination mechanisms used in the Darwinian evolutionary system. Reproduction in SIEAs occurs in two phases as follows.

(a) *Intra-group Reproduction*: Intra-group reproduction mechanisms utilized by SIEAs are as follows.
   (i) 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 towards the leader(s).
   (ii) The individuals follow better peer individual(s) to achieve a better place in the group.
   (iii) Random changes in the population members are made to enhance or maintain population diversity.
   (iv) 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 production can also take place in many ways as follows.
   (i) 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.
   (ii) The local leaders follow another leader(s).
   (iii) The local leaders go through self-observation and introduce a quantum of change in themselves using mutation-like operators.
   (iv) Individuals interact and follow the members of other groups.
   (v) Individuals switch their groups.

Following leaders or members can be modeled with a reinsertion operator or a weighted movement of members in the direction of leaders or other members. The quantum of change in each group member is an essential parameter in deciding how fast the member converges towards the leader(s) or other member(s). Too rapid convergence may destroy diversity in the group, resulting in a local optimum. However, too small changes increase computational cost. The crossover-like genetic operators may also be utilized for increased search power that contributes to exploration in the initial phases and exploitation towards the end of the algorithm.

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. If there is no improvement in a particular leader or member for some iterations, it may be replaced by a new randomly generated individual. Likewise, exploration may also be promoted by replacing the weak members of a group with new randomly generated individuals.

The sequence of intra- and inter-group reproduction phases can be implemented in an intertwined or reverse sequence manner. In the beginning, the groups must explore the subpopulation with a higher diversification rate, and 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 exploitation.

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.
   (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 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

## 4.1 Society and Civilization Algorithm (SCA)

Society and Civilization Algorithm (Ray & 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. The success of a society relies on the progress and success of its members. 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 the steps of SCA with those of the general framework is given in Table 4.

1. *Initialization and group 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 in parametric space 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 as leaders, and likewise, the best leaders become civilization leaders.
2. *Reproduction mechanisms*:
   (a) *Intra-group reproduction(s)*: Society leaders facilitate the improvement of other society individuals. This improvement is performed by the movement of individuals toward nearby society leaders in the parametric space. In effect, this simulates knowledge acquisition from the leader to improve individuals. In terms of EAs, this process implements exploitation.
   (b) *Inter-group reproduction(s)*: Society leaders themselves improve by moving or migrating toward the civilization leaders. The migration of society leaders expands promising regions in parametric space by receiving an increased individual count. In terms of EAs, this movement implements exploration. The newly generated improved individuals by intra- and inter-group improvement processes and the civilization leaders directly appear in the subsequent civilizations.

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

| **Steps in General Framework** | **Option(s) chosen in SCA** |
|---|---|
| 1A. Grouping mechanism | 6 |
| 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) |

3. *Group refreshment*: Society formation and leader selection are performed repeatedly at each subsequent iteration.
4. *Termination condition(s)*: The termination of SCA depends on the computational expense 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 originally proposed for constrained optimization problems. In SCA, for each individual, a constraint satisfaction vector is calculated and maintained throughout the run. A non-zero 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.

Talent Based Social Algorithm (Daneshyari & Yen, 2004) is an improved version of SCA. This algorithm proposes two new concepts of Liberty Rate and Talent to provide a new mechanism for moving individuals toward corresponding leaders. Liberty Rate measures society individuals' independence, defined as the ratio of society's average fitness and the average 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 & Thanushkodi, 2009), a prominent hybrid variation of SCA, integrates the self-experience concept from Particle Swarm Optimization (PSO). In CSO, a society individual

**Algorithm 2** Society and Civilization Algorithm

```
Generate a random population representing a civilization
Evaluate each individual according to objective function and satisfaction of constraints
repeat
    Build mutually exclusive societies using the clustering method
    Identify leaders in each society and create a leader set for each society
    Move each non-leader individual in the direction of the nearest leader and add this newly positioned individual to the new civilization
    Identify the civilization leader
    Move all the leaders but not the civilization leader in the direction of the nearest best leader and add these newly positioned leaders to the new civilization
    Add the civilization leader to the new civilization
until given civilization count
```

explores a 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 et al., 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 while 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 while the losing team members self-inspect their strategies and are substituted often. In imitation, fixed players in a winning team imitate both the SP and the SSP in the league to improve their future activities. In provocation, substitute players are promoted to fixed players to prove their performance equal to the average performance level of the fixed players in their team. 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 he thinks may benefit his goal. In substitution, new substitutes are tested in a match for their performance.

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

1. *Initialization and group 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 winner and loser teams are declared based upon 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

**Table 5** Hybrid Versions of SCA

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

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

| **Steps in General Framework** | **Option(s) chosen in SLC** |
|---|---|
| 1A. Grouping mechanism | 1(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) |

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.

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 pseudo-code for SLC is provided in Algorithm 3.

Diversity-team soccer league competition algorithm (DSLC) (Qiao, Dao, Pan, Chu, & Nguyen, 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 hybrid versions of SLC.

**Algorithm 3** Soccer League Competition Algorithm

```
Initialize the problem and algorithm parameters (Number of teams, players, fixed players, substitutes)
Generate random players
Evaluate players fitness
repeat
    Form the teams and allocate sorted players to them sequentially
    Start the league competition
    Find the losing and winning teams
    Apply corresponding operators in both the teams
until maximum number of seasons
```

## 4.3 Socio Evolution & 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.

**Table 7** Hybrid Versions of SLC

| Hybrid Version | Application | Publication |
|---|---|---|
| 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) |

Each member of a family 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 stuff from other parents. This interaction and learning from others helps in improving individual performance.

The algorithmic steps for SELO are outlined as follows, and the correspondence of the steps of SELO with those of the general framework is provided in Table 8.

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

| Steps in General Framework | Option(s) chosen in SELO |
|---|---|
| 1A. Grouping mechanism | 4(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) |

1. *Initialization and group 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 at last on 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, then parent intervention is required; a behavior correction operator is utilized to overcome such a situation.
3. *Group refreshment*: Global best society member is redetermined at the beginning of each iteration.
4. *Termination condition(s)*: The algorithm terminates when the maximum number of learning attempts are performed, or the behavior of families converges. The convergence of a family represents the saturation of all the family members to a single position, and at termination, any of these can be reported as the best solution found. As a rule, the best family in society contains the best solution(s).

The pseudo-code for SELO is provided in Algorithm 4.

## 4.4 Nomadic People Optimizer (NPO)

Nomadic People Optimizer (Salih & Alsewari, 2020) is inspired by the behavior of Nomads in the search for required food and water. NPO

**Algorithm 4** Socio Evolution and Learning Optimization Algorithm

1: Initialize population and families (parents and kids)
2: **repeat**
3: Find the best family and society member
4: Each parent of every family decides to randomly follow the behavior of a parent from one of the other families
5: Kids decide to follow either their parents or their siblings or kids from other families
6: **until** maximum number of learning attempts, or behavior of families converges

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 locations for the families' survival and distribution pattern at the location. The Sheikh selects a few normal families to explore the surrounding regions to find suitable locations. Whenever a better location is found, Sheikh moves the entire clan there.

The algorithmic steps for NPO are outlined as follows, and the correspondence of the steps of NPO with those of the general framework is provided in Table 9.

1. *Initialization and group 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, Zain, Yusuf, & Udin, 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 global best leader is determined at the beginning of inter-group improvement processes.
4. *Termination condition(s)*: NPO terminates after a maximum number of iterations have been completed.

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

| **Steps in General Framework** | **Option(s) chosen in NPO** |
|---|---|
| 1A. Grouping mechanism | 4(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 pseudo-code for NPO is provided in Algorithm 5, and Table 10 lists hybrid versions of NPO.

**Algorithm 5** Nomadic People Optimizer

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

**Table 10** Hybrid Versions of NPO

| Hybrid Version | Application | Publication |
|---|---|---|
| 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, Hashemzadeh, Rajabioun, & Lucas, 2008; Atashpaz-Gargari & Lucas, 2007b) 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 an imperialistic competition process and become more powerful. An empire with no colonies suffers collapse or is defeated by a better/best empire in this imperialistic competition.

The algorithmic steps for ICA are outlined as follows, and the correspondence of the steps of ICA with those of the general framework is provided in Table 11.

1. *Initialization and group 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 as imperialists, and the rest are termed colonies. The imperialists overcome nearby weaker countries in the proportion of their cost to form empires.
2. *Reproduction mechanisms*:
   (a) *Intra-group reproduction(s)*: Imperialists impose their characteristics on the possessed colonies. Colonies move toward their imperialist in objective space to simulate this process. In effect, this is imitating 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, and weaker ones get weaker and, at last, collapse into better empires.
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 a strong empire after the imperialistic competition.
4. *Termination condition(s)*: The weaker colonies and empires are collapsed 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 discriminated on the basis of their power, calculated using the cost of their imperialist and colonies. Upon termination, the imperialist of the best empire holds the best solution.

The pseudo-code for ICA is provided in Algorithm 6.

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

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

| Steps in General Framework | Option(s) chosen in ICA |
|---|---|
| 1A. Grouping mechanism | 3(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) |

**Algorithm 6** Imperialist Competitive Algorithm

```
1: Generate some random solutions (countries)
2: Initialize the empires
3: repeat
4:     Assimilate and revolute empires: move the colonies towards 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 to possess it
8:     Eliminate empires having no colonies
9: until only one empire remains
```

viz., Discrete Binary ICA (DB-ICA) Nozarian, Soltanpoora, and Jahanb (2012), Discrete ICA (DICA) H. 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 & Khaled, 2014). The broad applicability of ICA requires 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 & 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 and 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 outlined as follows, and the correspondence of the steps of POA with those of the general framework is provided in Table 13.

1. *Initialization and group 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)*: The regular political party members are biased in favor of 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.

**Table 12** Hybrid Versions of ICA

| Hybrid Version | Application | Publication |
|---|---|---|
| ICA + EA | Global optimization | Ramezani and Lotfi (2013) |
| ICA + PSO | Optimal reactive power dispatch of power systems | Mehdinejad, Mohammadi-Ivatloo, Dadashzadeh-Bonab, and Zare (2016) |
| ICA + ABC | Sentiment classification | Osmani, Mohasefi, and Gharehchopogh (2022) |

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

| Steps in General Framework | Option(s) chosen in POA |
|---|---|
| 1A. Grouping mechanism | 1(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) |

3. *Group refreshment*: Candidate members in each group are reassigned after intra-group improvement. Some of the best groups are merged and a number of 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 and the best among them is the best solution in each iteration.

The pseudo-code for POA is provided in Algorithm 7.

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.

**Algorithm 7** Parliamentary Optimization Algorithm

```
Generate random population
Equally divide the entire population into some groups
Select some highly fit individuals in each group as candidate members
repeat
    Bias regular members of each group towards the candidates of the same group
    Reassign candidate members
    Compute each group's power
    Select some best groups and merge them with some predefined probability
    Delete some groups with a predefined probability
until only one political group remains
```

## 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.

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, and the correspondence of the steps of ECO with those of the general framework is provided in Table 15.

1. *Initialization and group 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 voter increases, and after a range limit of support region, it reduces to zero. The support regions can be

**Table 14** Hybrid Versions of POA

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

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

| Steps in General Framework | Option(s) chosen in ECO |
|---|---|
| 1A. Grouping mechanism | 4(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) |

overlapping, i.e., a voter can be affected by one or more candidates depending upon their support ranges.

2. *Reproduction mechanisms*:
   (a) *Intra-group reproduction(s)*: Voters affected by multiple candidates distribute their support in proportion of their effect. In this process, a voter's prestige can become better than a candidate, and 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.

The pseudo-code for ECO is provided in Algorithm 8.

**Algorithm 8** Election Campaign Optimization

Generate candidates, calculate their prestige and define their ranges accordingly
Generate voters
**repeat**
    Calculate the effect of candidates on voters
    Calculate the prestige and support of the voters
    Find the new support focus and ranges of the candidates
    Substitute the candidate with the voter having better prestige than that of the candidate, if any
**until** maximum number of iterations

A comparative study verifying the good performance of ECO on constrained optimization problems is reported by Q. Xie et al. (2010). A parameter design and performance study on the ECO algorithm is presented by L. Zhang, Lv, Wang, Cheng, and Luo (2011).

## 4.8 Group Leaders Optimization Algorithm (GLOA or GLA)

Group Leaders Optimization Algorithm (Daskin & 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 gain better characteristics.

The algorithmic steps for GLOA are outlined as follows, and the correspondence of the steps of GLOA with those of the general framework is provided in Table 16.

1. *Initialization and groups formation*: In this algorithm, a few groups are generated

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

| Steps in General Framework | Option(s) chosen in GLOA |
|---|---|
| 1A. Grouping mechanism | 1(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) |

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 are used to terminate the algorithm. The leader in each group represents the best local solution, while the best of these leaders represents the best found solution so far.

The pseudo-code for GLOA is provided in Algorithm 9.

Xiang, Hu, Yu, and Wu (2014) proposed a Pareto-GLA to solve multi-objective optimization algorithms. Table 17 lists some hybrid versions of GLOA.

**Algorithm 9** Group Leaders Optimization Algorithm

```
Initialize some groups, each having equal number of individuals
Evaluate the fitness of each individual
repeat
    Determine the leader for each group
    Generate new individuals using crossover operator between an individual and the group leader and mutate the generated individual
    Influence individuals from other group individuals selected randomly
until maximum number of iterations
```

## 4.9 Election Algorithm (EA)

The Election Algorithm (H. Emami & Derakhshan, 2015) is inspired by the presidential election 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, either a candidate or a voter, participates in an election. 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 the steps of EA with those of the general framework is provided in Table 18.

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.

**Table 17** Hybrid Versions of GLOA

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

After that, various electoral parties are formed according to the same interests, beliefs, and ideas to participate in the election, using the clustering method.

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 creates a global impact on individuals 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.

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

| **Steps in General Framework** | **Option(s) chosen in EA** |
|---|---|
| 1A. Grouping mechanism | 5 |
| 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

```
Generate an initial population of individuals
Evaluate each individual
Create electoral parties
repeat
    for candidate size do
        Candidates advertise their plans and enhance their positions within the party by learning new ideas
        Candidates aim to win over supporters from other parties
        Collate candidates if they have similar ideas
        Re-evaluate the eligibility of candidates
    end for
until maximum number of iterations
```

The pseudo-code for the Election Algorithm is provided in Algorithm 10.

A new election algorithm based on assistance in distributed systems (Zargarnataj, 2007) is an improvement over this algorithm. H. Emami (2019) proposed an improved Election Algorithm by modifying party formation step and introducing chaotic positive advertisement and migration operator. A modified version of Election Algorithm to solve random $k$ satisfiability problem is provided by Abubakar, Sabri, Masanawa, and

Yusuf (2020); Sathasivam, Mansor, Kasihmuddin, Abubakar, et al. (2020).

A comparison between the election algorithm and election campaign optimization algorithm is presented in (Abubakar & Sathasivam, 2020). Abubakar, Sathasivam, and Alzaeemi (2020) reported a significant enhancement in the Election Algorithm on incorporating negative campaign strategy. Table 19 lists the hybrid versions of 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 the self-interest and competitive behavior of political party members that they exhibit 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, and the correspondence of the steps of IA with those of the general framework is provided in Table 20.

1. *Initialization and groups formation*: In IA, the population of randomly generated possible solutions is 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 other individuals 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 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 out 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 and/or the maximum number of iterations is reached. The best among all local party leaders provides the global best solution.

The pseudo-code for IA is provided in Algorithm 11.

**Algorithm 11** Ideology Algorithm

```
Initialize population
Divide the population sequentially into equal-sized political parties
Evaluate each individual
repeat
    Rank individuals in each party
    Search in the neighborhood of each party best, each second party best, and the global best members
    Worst individuals switch their parties due to a predefined condition
    Ordinary individuals introspect and follow other local party individuals
    Update party individuals
until no significant change over iterations and/or the maximum number of iterations
```

**Table 19** Hybrid Versions of EA

| Hybrid Version | Application | Publication |
|---|---|---|
| EA + Kriging Method | Groundwater monitoring network design | Kavusi, Khashei Siuki, and Dastourani (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 | S. Emami, Emami, and Parsa (2022) |

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

| Steps in General Framework | Option(s) chosen in IA |
|---|---|
| 1A. Grouping mechanism | 1(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) |

## 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, and at the same time, 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 past experience to win over the voters, and 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.

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, respectively, depending upon the difference between 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 found solution.

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

| Steps in General Framework | Option(s) chosen in PO |
|---|---|
| 1A. Grouping mechanism | 1(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) |

The pseudo-code for PO is provided in Algorithm 12.

**Algorithm 12** Political Optimizer

```
Initialize the population and divide it into equal-sized parties sequentially
Election: evaluate the fitness of each party member
Government formation: compute the sets of all party leaders and constituency winners
repeat
    Election campaign: update party members and candidates according to their corresponding party leaders and constituency winners, respectively
    Party switching: exchange members between parties
    Election and government formation
    Parliamentary affairs: constituency winners are updated based upon another randomly selected constituency winner
until the maximum number of iterations
```

Manita and Korbaa (2020) proposed a binary version of PO using eight transfer functions (Mirjalili & 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 improving its position-updating mechanism and demonstrated its applicability on benchmark complex 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 some hybrid versions of PO.

# 5 Application Areas

In this section, an exhaustive search is performed, and the results are tabulated in Table 23, which lists the various reported applications of the state-of-the-art SIEAs. From a study of the analysis of Table 23, the following salient points can be summarized.

1. Almost all SIEAs have been applied to solve various real-function optimization problems.
2. Imperialist Competitive Algorithm and Election Campaign Optimization have been prevalent 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.

**Table 22** Hybrid Versions of PO

| Hybrid Version | Application | Publication |
|---|---|---|
| PO + Random Vector Functional Link (RVFL) | Metal cutting process response prediction | Elsheikh, Abd Elaziz, Das, Muthuramalingam, and Lu (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, Sudabattula, and Velamuri (2021) |
| PO + Voltage Stability Index (VSI) | Capacitor Allocation problem in Distribution Systems | Mani, Varma, Krishna, Khan, and Sudabattula (2021) |
| PO + Shannon entropy function | Flying Ad Hoc Networks Clustering | Asaamoning, Mendes, and Magaia (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 problems | Dong, Zou, Li, and Wang (2022) |
| PO + Dragonfly Algorithm | Soil Moisture and Heat Level Prediction in IoT | Muppidi, PG, et al. (2022) |
| PO + Multi-verse Optimizer | Tumor Classification and Survival Prediction of Brain Tumor Patients with MRI | Rajeswari, Neelima, Maram, and Angadi (2022) |
| PO + Sine Cosine Algorithm | Tweet Hashtag Recommendation | Banbhrani, Xu, Liu, and Lin (2021) |
| PO + Sailfish optimization algorithm (SOA) | Optimal DeepMRSeg based tumor segmentation | Neelima, Chigurukota, Maram, and Girirajan (2022) |
| PO + Tunicate Swarm Algorithm | UAV (Unmanned Aerial Vehicle) network secure communication | Sangeetha Francelin, Daniel, and Velliangiri (2022) |
| PO + Competitive Multi-verse Optimization enabled Deep Neuro fuzzy network (PFCMVO enabled DNFN) | Student performance estimation in spark environment | Jyoti Baruah and Baruah (2022) |

# 6 Conclusions and Future directions

Evolutionary computation has gone through 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 several animals and insects. More recently, inspiration from social phenomena, e.g., human behavior exchange and knowledge transfer, has ushered in a new evolutionary computing paradigm entitled Socio-inspired Evolutionary Algorithms (SIEAs). It is accepted that the evolution through human social phenomena and humans' problem-solving tactics is much faster than biological evolution alone. Numerous evolutionary algorithms inspired by these social phenomena have been proposed in the literature employing a variety of terminologies used in the domain from which inspiration is derived, e.g., elections, societal and imperial colonization, to name a few. This diverse terminology has created a situation in which some algorithms appear to be entirely different but are very similar in matters of significant detail. Anyone trying to understand the area must sift through this terminological maze to arrive at a meaningful understanding.

**Table 23** Applications of SIEAs

| SIEA | Applications | Publication |
|---|---|---|
| SCA | Single objective constrained engineering design optimization problems | Akhtar, Tai, and Ray (2002); Ray and Liew (2003) |
| | Combined heat and power economic dispatch problems | Sharma, Sharma, and Sachdev (2018) |
| SLC | Systems of nonlinear equations | Moosavian et al. (2013) |
| | Optimal design of water distribution networks in urban areas | Moosavian and Kasaee Roodsari (2014) |
| | Knapsack problem | Moosavian (2015) |
| | Set covering problem | Jaramillo, Crawford, Soto, Misra, et al. (2016); Jaramillo, Crawford, Soto, Villablanca, et al. (2016); Jaramillo, Gómez, et al. (2016) |
| | 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, Naseri, Saeedi Nezhad, Etebari Ghasbeh, and Moghadas Nejad (2021) |
| | Pavement maintenance planning of large-scale transportation networks considering energy consumption | Golroo, Fani, and Naseri (2021) |
| | Load flow analysis of power systems | Srilakshmi, Babu, Venkatesan, and Palanivelu (2022) |
| SELO | 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, Al-Anbarri, and Hannun (2020a, 2020b, 2020c) |
| | Zoning map for drought prediction using integrated machine learning models | Mohamadi et al. (2020) |
| | Test list generation for interaction testing in IoT | Alsewari (2021) |
| ICA | Nash equilibrium point achievement | Rajabioun, Atashpaz-Gargari, and Lucas (2008) |
| | Off-line scheduling problem with rejection | Jolai, Sangari, and Babaie (2010) |
| | Hybrid flow-shop scheduling | Behnamian and Zandieh (2011) |
| | Mixed-model U-line balancing and sequencing problem | Lian, Zhang, Gao, and Shao (2012) |
| | PID controller designing | Atashpaz-Gargari, Hashemzadeh, and Lucas (2008); Atashpaz Gargari et al. (2008); Atashpaz-Gargari and Lucas (2007a) |
| | MIMO PIID controller designing | Atashpaz-Gargari et al. (2008) |
| | Materials property characterization from sharp indentation test | Biabangard-Oskouyi, Atashpaz-Gargari, Soltani, and Lucas (2009) |

| | | |
|---|---|---|
| | Integrated product mix-outsourcing problem | Nazari-Shirkouhi, Eivazy, Ghodsi, Rezaie, and Atashpaz-Gargari (2010) |
| | Optimum skeletal structures designing | Kaveh and Talatahari (2010) |
| | Linear induction motor designing | Lucas, Nasiri-Gheidari, and Tootoonchian (2010) |
| | Optimizing the free convection heat transfer in a vertical cavity with flow diverters | Karami, Yousefi, Ghashghaei, and Rezaei (2011) |
| | Optimal simultaneous coordinated tuning of damping controller | Bijami and Marnani (2012) |
| | Plate fin heat exchanger designing | Yousefi, Darus, and Mohammadi (2012) |
| | Adhesive-bonded fiber glass strip optimization | Mozafari, Abdi, and Ayob (2012) |
| | Robust PID controller for load-frequency control of power systems | Shabani, Vahidi, and Ebrahimpour (2013) |
| | Sliding mode controller designing | Jalali, Piltan, Keshtgar, and Jalali (2013) |
| | Fractional order PID controller design for LFC in electric power systems | Taher, Fini, and Aliabadi (2014) |
| | Non-convex economic dispatch problem | Bijami, Jadidoleslam, Ebrahimi, Askari, and Farsangi (2014) |
| | Ground vibration production in quarry blasting | Hajihassani, Armaghani, Marto, and Mohamad (2015) |
| | Harmonic minimization in multilevel inverters | Etesami, Farokhnia, and Fathi (2015) |
| POA | 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) |
| | Web pages classification | Kiziloluk and Ozer (2017) |
| | Community structure identification in social networks | Shakya, Shaik, Singh, Sinha, and Biswas (2020) |
| ECO | 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, Deyuan, Siyuan, Shaoming, and Zeyu (2010a) |
| | Tuning digital PID controllers for discrete-time systems | Wenge, Deyuan, Siyuan, Shaoming, and Zeyu (2010b) |
| | Multi-peak optimization problems | Wenge, Zhiyong, et al. (2010) |
| | Pressure vessel design problems | H. Zhang, Lv, Cheng, Luo, and Zhang (2011) |
| | Generation expansion planning in deregulated power systems | Shayeghi and Esmaeli (2013) |
| | Re-use reverse logistics system model of disused electric appliances | Q.H. Xie, Zhang, Lv, Cheng, Huang, and Cai (2013) |
| | Quantity prediction and optimal disposal capacity of waste electric appliances | Q.H. Xie, Zhang, Lv, Cheng, Lin, and Yang (2013) |
| | Rotated image matching | Zeng, Lv, and Xie (2015) |
| | Unit commitment problems with wind power effect | Alizadeh, Akbarimajd, and Ghadimi (2016) |
| | Feature points matching | Mei, Lv, and Chen (2016) |
| | Multilevel Thresholding segmentation technique | TU and LV (2016) |

| | | |
|---|---|---|
| | Fault diagnosis | Q. Xie, Zhang, Lv, and Cheng (2016a) |
| | Image comparison | Q. Xie, Zhang, Lv, and Cheng (2016b) |
| | Image matching | Q. Xie, Zhang, Lv, and Cheng (2016c) |
| | Marbling stone slab image segmentation | Q. Xie, Zhang, Lv, and Cheng (2016d) |
| | Public transport scheme optimization | Q. Xie, Zhang, Lv, and Cheng (2016e) |
| | Video tracking | Q. Xie, Zhang, Lv, and Cheng (2016f) |
| | Stereo matching | Q.H. Xie, Zhang, Lv, and Cheng (2016) |
| | Detecting maximum inscribed rectangle | Q.-H. Xie, Zhang, Lv, and Cheng (2017) |
| GLOA | Decomposing unitary matrix into a proper-minimum cost quantum gate sequence | Cao, Daskin, Frankel, and Kais (2012); Daskin (2014); Daskin and Kais (2011a) |
| | Job scheduling in a grid computing system | Pooranian, Shojafar, Abawajy, and Singhal (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 | M. Liu, Yi, Wen, and Song (2017) |
| | Economic load dispatch problems | Roy, Bhattacharjee, and Bhattacharya (2017); Shah, Bhattacharjee, and Godwal (2020) |
| | Low-cost circuit approximations for the Hamiltonian simulation | Dutta et al. (2018) |
| EA | Benchmark Functions | H. Emami and Derakhshan (2015) |
| | Groundwater level prediction | Choopan, Emami, and Kheiri Ghooje Bigloo (2020); H. Emami and Emami (2019); S. Emami, Choopan, and Parsa (2018); S. Emami, Emami, Choopan, Parsa, and Jahandideh (2020) |
| | Suspended sediment load prediction of rivers | H. Emami, Emami, and Heydari (2019); S. Emami and Parsa (2022a) |
| | Random $k$ satisfiability in Hopfield neural network | Abubakar, Masanawa, Yusuf, and Boaku (2021); Abubakar, Sabri, et al. (2020); Bazuhair et al. (2021); Sathasivam et al. (2020) |
| | Crop cultivation pattern optimization | Choopan and Emami (2020) |
| | Optimal Economic Water Allocation | S. Emami and Choopan (2020) |
| | Concrete Dams cracks evaluation | S. Emami, Parsa, and Emami (2021) |
| | Concrete Gravity Dam Section Optimization | S. Emami and Parsa (2022b) |
| ID | Unconstrained Benchmark Functions | Huan et al. (2017) |
| PO | PI Controller design | Devarapalli, Lakshmi, and Prasad (2020) |
| | Proton exchange membrane fuel cell (PEMFC) parameters estimation problem | Diab, Tolba, El-Magd, Zaky, and El-Rifaie (2020) |
| | Solar photovoltaic cell unknown parameters identification | Premkumar, Sowmya, Jangir, and Kumar (2020) |

| | |
|---|---|
| Levelized Cost of Hybrid Wind-Solar-Diesel-Battery System Optimization | Singh, Pandit, and Srivastava (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, SUDABATTULA, and Velamuri (2021) |
| Concentric circular antenna arrays (CCAAs) radiation properties improvement | Durmus and Kurban (2021) |
| Structural design optimization for buckling-restrained braces (BRBs) | Hoseini, Parastesh, Hajirasouliha, and Ferdowsi (2021) |
| Color aerial image multilevel thresholding | Kurban, Durmus, and Karakose (2021) |
| Energy-management system for a microgrid installation | Suresh et al. (2021) |
| Integration of DGs in the distribution grids | Tolba and Tulsky (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, Rezk, Abdelkareem, and Olabi (2022) |

This paper presents a generalized framework for SIEAs. The presented 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. A survey of various SIEAs is also provided to highlight the working of these algorithms and their improved hybrid versions. The algorithmic descriptions and pseudocodes provided help in understanding 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.

Furthermore, Table 24 provides the utilization count of all the framework components in various published SIEAs. Based on the data provided in Table 24, the following points emerge.

1. The highlighted (in bold) mechanisms serve as most popular choices in SIEAs. However, the best choice of mechanisms for a new SIEA is dependent on the fact that the combination of all the mechanisms balances the exploration and exploitation requirements perfectly.
2. Some framework components have contrasting mechanisms that cannot be utilized simultaneously. However, in the case of the other mechanisms, the chosen ones can be collected to propose what might work as a good SIEA. Combinations that have not been attempted yet might be put together to develop new and promising approaches.

The SIEA proposed above can be demonstrated on some benchmark combinatorial and real-valued function optimization problems in the future. Since the different SIEAs appear to have different strengths and search power, an effort 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. However, No Free Lunch Theorem (Wolpert, Macready, et al., 1995, 1997)

**Table 24** SIEAs general framework's steps utilization counts

| Step | Mechanism utilization count |
|---|---|
| 1A. Grouping mechanism | **1(a) - 5**, 1(b) - 0, 2(a) - 0, 2(b) - 0, 3(a) - 0, 3(b) - 1, 4(a) - 3, 4(b) - 0, 5 - 1, 6 - 1. |
| 1B. Evaluation | (a) - 3, **(b) - 8**. |
| 1C. Leader selection | (a) - 3, **(b) - 6**, (c) - 2. |
| 2A. Intra-group reproduction | **(a) - 9**, (b) - 2, (c) - 6, (d) - 1. |
| 2B. Inter-group reproduction | (a) - 3, (b) - 2, (c) - 1, **(d) - 7**, (e) - 3. |
| 3. Group refreshment | **(a) - 9**, (b) - 1, (c) - 2, (d) - 2. |
| 4. Termination criterion | (a) - 2, **(b) - 10**, (c) - 1. |

prohibits the development of an algorithm that does better than every other algorithm on every problem.

In most cases, the SIEA proposed and published reports better results than traditional EAs. A rigorous study that compares the performance of various SIEAs on a fair platform and diverse set of applications is conspicuous by its absence. Such a study is needed to clarify the relative strengths of the various SIEAs and the respective domains in which each is better than the others.

## References

Abdelhamid, M., Kamel, S., Mohamed, M.A., Aljohani, M., Rahmann, C., Mosaad, M.I. (2020). Political optimization algorithm for optimal coordination of directional overcurrent relays. *2020 ieee electric power and energy conference (epec)* (pp. 1–7).

Abdolrazzagh-Nezhad, M., & Adibiyan, 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, M.L. (2021). Hopfield type of artificial neural network via election algorithm as heuristic search method for random boolean ksatisfiability. *International Journal of Computing and Digital Systems*.

Abubakar, H., Masanawa, A.S., Yusuf, S., Boaku, G. (2021). Optimal representation to high order random boolean ksatisability via election algorithm as heuristic search approach in hopeld neural networks. *Journal of the Nigerian Society of Physical Sciences*, 201–208.

Abubakar, H., Sabri, S.R.M., Masanawa, S.A., Yusuf, S. (2020). Modified election algorithm in hopfield neural network for optimal random k satisfiability representation. *International Journal for Simulation and Multidisciplinary Design Optimization*, *11*, 16.

Abubakar, H., & Sathasivam, S. (2020). Comparing election algorithm and election campaign optimization algorithm. *Aip conference proceedings* (Vol. 2266, p. 040006).

Abubakar, H., Sathasivam, S., Alzaeemi, S.A. (2020). Effect of negative campaign strategy of election algorithm in solving optimization problem. *Journal of Quality Measurement and Analysis JQMA*, *16*(2), 171–181.

Akhtar, S., Tai, K., Ray, T. (2002). A socio-behavioural simulation model for engineering design optimization. *Engineering Optimization*, *34*(4), 341–354.

Alizadeh, A., Akbarimajd, A., Ghadimi, N. (2016). *Election campaign optimization in unit commitment problem with wind power effect.*

Alsewari, A.A. (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. *International Journal of Computer Networks and Applications*, *2*(1), 12–19.

Anderson, J.C. (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. (2021a). Improved political optimizer for complex landscapes and engineering optimization problems. *Expert Systems with Applications*, *182*, 115178.

Askari, Q., & Younas, I. (2021b). Political optimizer based feedforward neural network for classification and function approximation. *Neural Processing Letters*, *53*(1), 429–458.

Askari, Q., Younas, I., Saeed, M. (2020). Political optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowledge-Based Systems*, *195*, 105709.

Atashpaz-Gargari, E., Hashemzadeh, F., Lucas, C. (2008). Designing mimo piid controller using colonial competitive algorithm: applied to distillation column process. *Evolutionary computation, 2008. cec 2008.(ieee world congress on computational intelligence). ieee 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. *International Journal of Intelligent Computing and Cybernetics*, *1*(3), 337–355.

Atashpaz-Gargari, E., & Lucas, C. (2007a). Designing an optimal pid controller using imperialist competitive algorithm. *First joint congress on fuzzy and intelligent systems ferdowsi university of mashhad, iran* (pp. 29–31).

Atashpaz-Gargari, E., & Lucas, C. (2007b, Sept). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. *2007 ieee congress on evolutionary computation* (pp. 4661–4667). 10.1109/cec.2007.4425083

Awad, R. (2021). Sizing optimization of truss structures using the political optimizer (po) algorithm. *Structures* (Vol. 33, pp. 4871–4894).

Bäck, T., Fogel, D.B., Michalewicz, Z. (1997). *Handbook of evolutionary computation.* CRC Press.

Banbhrani, S.K., 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.

Barricelli, N.A. (1962). Numerical testing of evolution theories. *Acta Biotheoretica*, *16*(1), 69–98.

Basetti, V., Rangarajan, S.S., Shiva, C.K., Pulluri, H., Kumar, R., Collins, R.E., 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, M.M., Jamaludin, S.Z.M., Zamri, N.E., Kasihmuddin, M.S.M., Mansor, M., Alway, A., . . . others (2021). Novel hopfield neural network model with election algorithm for random 3 satisfiability. *Processes*, *9*(8), 1292.

Beasley, D. (2000). Possible applications of evolutionary computation. *Evolutionary computation*, *1*, 4–19.

Becerra, R.L., & Coello, C.A.C. (2006). Cultured differential evolution for constrained optimization. *Computer Methods in Applied Mechanics and Engineering*, *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 Systems with Applications*, *38*(12), 14490–14498.

Bi, L., Zuo, Y., Tao, F., Liao, T.W., Liu, Z. (2017). Energy-aware material selection for product with multicomponent under cloud environment. *Journal of Computing and Information Science in Engineering*, *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. *International Journal of Engineering Simulation*, *10*(1), 11–12.

Bijami, E., Jadidoleslam, M., Ebrahimi, A., Askari, J., Farsangi, M.M. (2014). Implementation of imperialist competitive algorithm to solve non-convex economic dispatch problem. *Journal of the Chinese Institute of Engineers*, *37*(2), 232–242.

Bijami, E., & Marnani, J.A. (2012). Imperialist competitive algorithm for optimal simultaneous coordinated tuning of damping controller. *International Journal on Technical and Physical Problems of Engineering (IJTPE)*, 34–41.

Biswas, A., Mishra, K., Tiwari, S., Misra, A. (2013). Physics-inspired optimization algorithms: a survey. *Journal of Optimization*, *2013*.

Bjorklund, D.F., Causey, K., Periss, V. (2010). The evolution and development of human social cognition. *Mind the gap* (pp. 351–371). Springer.

Bonyadi, M.R., Azghadi, S., Hosseini, H.S. (2007). Solving traveling salesman problem using combinational evolutionary algorithm. *Ifip international conference on artificial intelligence applications and innovations* (pp. 37–44).

Borji, A. (2007). A new global optimization algorithm inspired by parliamentary political competitions. A. Gelbukh & Á.F. Kuri Morales (Eds.), *Micai 2007: Advances in artificial intelligence* (pp. 61–71). Berlin, Heidelberg: Springer Berlin Heidelberg.

Borji, A., & Hamidi, M. (2009). A new approach to global optimization motivated by parliamentary political competitions. *International Journal of Innovative Computing, Information and Control*, *5*(6), 1643–1653.

Brindha, G.J., & Gopi, E. (2019). Maximizing profits in crop planning using socio evolution and learning optimization. *Socio-cultural inspired metaheuristics* (pp. 151–174). Springer.

Cao, Y., Daskin, A., Frankel, S., Kais, S. (2012). Quantum circuit design for solving linear systems of equations. *Molecular Physics*, *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. *Journal of Applied Research in Water and Wastewater*, *7*(1), 48–56.

Choopan, Y., Emami, S., Kheiri Ghooje Bigloo, M. (2020). Evaluating election, imperialist competitive algorithms and artificial neural network method in investigating the groundwater level of reshtkhar plain. *Amirkabir Journal of Civil Engineering*, *52*(6), 1333–1246.

Chung, C.-J., & Reynolds, R.G. (1998). Caep: An evolution-based tool for real-valued function optimization using cultural algorithms. *International Journal on Artificial Intelligence Tools*, *7*(03), 239–291.

Črepinšek, M., Liu, S.-H., Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. *ACM computing surveys (CSUR)*, *45*(3), 1–33.

Cuevas, E., Diaz, P., Camarena, O. (2021). Experimental analysis between exploration and exploitation. *Metaheuristic computation: A performance perspective* (pp. 249–269). Springer.

Daneshyari, M., & Yen, G.G. (2004). Talent based social algorithm for optimization. *Proceedings of the 2004 congress on evolutionary computation (ieee cat. no. 04th8753)* (Vol. 1, pp. 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 Computing and Informatics Journal*, *3*(2), 231–246.

Daskin, A. (2014). *Quantum circuit design methods and applications* (Unpublished doctoral dissertation). Purdue University.

Daskin, A., & Kais, S. (2011a). Decomposition of unitary matrices for finding quantum circuits: application to molecular hamiltonians. *The Journal of chemical physics*, *134*(14), 144112.

Daskin, A., & Kais, S. (2011b). Group leaders optimization algorithm. *Molecular Physics*, *109*(March 2011), 761–772.

10.1080/00268976.2011.552444

de Marcos, L., García, A., García, E., Martínez, J.J., Gutiérrez, J.A., Barchino, R., ... Otón, S. (2010). An adaptation of the parliamentary metaheuristic for permutation constraint satisfaction. *Ieee congress on evolutionary computation* (pp. 1–8).

Devarapalli, R., Lakshmi, N.J.N., Prasad, U. (2020). Application of a novel political optimization in optimal parameter design of pi controller for the bldc motor speed control. *2020 international conference on emerging frontiers in electrical and electronic technologies (icefeet)* (pp. 1–6).

Dharavat, N., Sudabattula, S.K., Velamuri, S. (2021). Integration of distributed generation and electric vehicles in a distribution network using political optimizer. *2021 4th international conference on recent developments in control, automation &*

*power engineering (rdcape)* (pp. 521–525).

Dharavat, N., SUDABATTULA, S.K., Velamuri, S. (2021). Optimal allocation of multiple distributed generators and shunt capacitors in a distribution system using political optimization algorithm. *International Journal of Renewable Energy Research (IJRER)*, *11*(4), 1478–1488.

Diab, A.A.Z., Tolba, M.A., El-Magd, A.G.A., Zaky, M.M., El-Rifaie, A.M. (2020). Fuel cell parameters estimation via marine predators and political optimizers. *IEEE Access*, *8*, 166998–167018.

Dogan, A. (2021a). 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. (2021b). Optimum sitting and sizing of wts, pvs, esss and evcss using hybrid soccer league competition-pattern search algorithm. *Engineering Science and Technology, an International Journal*, *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 meta-heuristic. *Proceedings of the 1999 congress on evolutionary computation-cec99 (cat. no. 99th8406)* (Vol. 2, pp. 1470–1477).

Durmus, A., & Kurban, R. (2021). Optimal synthesis of concentric circular antenna arrays using political optimizer. *IETE Journal of Research*, 1–10.

Dutta, S., Suau, A., Dutta, S., Roy, S., Behera, B.K., Panigrahi, P.K. (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 wsns (wireless sensor networks). *Wireless Personal Communications*, *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 Journal of Civil Engineering*, *53*(10), 1–1.

Eiben, A.E., & Schippers, C.A. (1998). On evolutionary exploration and exploitation. *Fundamenta Informaticae*, *35*(1-4), 35–50.

Elsheikh, A.H., Abd Elaziz, M., Das, S.R., 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. *Journal of Manufacturing Processes*, *67*, 562–578.

Emami, H. (2019). Chaotic election algorithm. *Computing and Informatics*, *38*(6), 1444–1478.

Emami, H., & Derakhshan, F. (2015, jul). 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 journal of Ecohydrology*, *6*(1), 177–190.

Emami, H., Emami, S., Heydari, S. (2019). Prediction suspended sediment load of river using meta-heuristic algorithms. *Iranian Journal of Irrigation & 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 Journal of Civil Engineering*, *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 journal of Ecohydrology*, *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. *Advances in Environmental Technology*, *6*(2), 99–110.

Emami, S., Emami, H., Parsa, J. (2022). Performance of vertically cosine shape weir using artificial intelligence. *Modeling Earth Systems and Environment*, *8*(1), 787–798.

Emami, S., & Parsa, J. (2022a). Optimal design of river groynes using meta-heuristic models. *Environment and Water Engineering*, *8*(1), 146–160.

Emami, S., & Parsa, J. (2022b). Optimization of concrete gravity dam section using new election meta-heuristic algorithm. *Journal of Structural and Construction Engineering*, *8*(12), 164–183.

Emami, S., Parsa, J., Emami, H. (2021). Evaluating cracks in concrete dams using meta-heuristic algorithms and artificial neural networks. *Journal of Structural and Construction Engineering*, *8*(Special Issue 2), 194–215.

Etesami, M., Farokhnia, N., Fathi, S.H. (2015). Colonial competitive algorithm development toward harmonic minimization in multilevel inverters. *IEEE Transactions on Industrial Informatics*, *11*(2), 459–466.

Eusuff, M.M., & Lansey, K.E. (2003). Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources planning and management*, *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. *Journal of Bionic Engineering*, *17*(3), 611–631.

Fathy, A., Ferahtia, S., Rezk, H., Yousri, D., Abdelkareem, M.A., 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 sofc optimal parameters for static and dynamic models. *Energy*, *238*, 122031.

Ferahtia, S., Rezk, H., Abdelkareem, M.A., Olabi, A. (2022). Optimal techno-economic energy management strategy for building's microgrids based bald eagle search optimization algorithm. *Applied Energy*, *306*, 118069.

Fogel, D.B. (1997). The advantages of evolutionary computation. *Biocomputing and emergent computation: Proceedings of bcec97* (pp. 1–11). World Scientific Press.

Fraser, A.S. (1958, jan). Monte carlo analyses of genetic models. *Nature*, *181*, 208–209.

10.1038/181208a0

Giddings, F.H. (1914). *Inductive sociology: A syllabus of methods, analyses and classifications, and provisionally formulated laws.* Macmillan.

Goldberg, D.E. (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 Journal of Civil Engineering*, *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. *International Journal of Intelligent Systems*, *37*(2), 1273–1298.

Hajihassani, M., Armaghani, D.J., Marto, A., Mohamad, E.T. (2015). Ground vibration prediction in quarry blasting through an artificial neural network optimized by imperialist competitive algorithm. *Bulletin of Engineering Geology and the Environment*, *74*(3), 873–886.

Hayes, E.C. (1911). The classification of social phenomena. *American Journal of Sociology*, *17*(1), 90–118.

He, C., & Zhang, X. (2010). Application of the election campaign optimization algorithm to engineering optimum design problems. *2010 international symposium on computational intelligence and design* (Vol. 2, pp. 34–37).

Holland, J.H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence.* Cambridge, MA, USA: MIT Press.

Hoseini, S.M., Parastesh, H., Hajirasouliha, I., Ferdowsi, A. (2021). Structural design optimization of all-steel buckling-restrained braces using intelligent optimizers. *International Journal of Steel Structures*, *21*(6), 2055–2070.

Hosseini, S., & Khaled, A.A. (2014). A survey on the imperialist competitive algorithm metaheuristic: Implementation in engineering domain and directions for future research. *Applied Soft Computing*, *24*, 1078–1094.

Hosseini-Hemati, S., Karimi, S., Shaeisi, G.H. (2021). How does the civilized gravitational search algorithm solve the optimal dg placement? *Journal of Energy Management and Technology*, *5*(3), 8–20.

Huan, T.T., Kulkarni, A.J., Kanesan, J., Huang, C.J., Abraham, A. (2017, dec). Ideology algorithm: a socio-inspired optimization methodology. *Neural Computing and Applications*, *28*(S1), 845–876.

Iacoban, R., Reynolds, R.G., Brewster, J. (2003). Cultural swarms: modeling the impact of culture on social interaction and problem solving. *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. *International Journal of Intelligent Systems and Applications*, *5*(7), 50.

Jaramillo, A., Crawford, B., Soto, R., Misra, S., Olguín, E., Rubio, Á.G., ... Villablanca, S.M. (2016). An approach to solve the set covering problem with the soccer league competition algorithm. *International conference on computational science and its applications* (pp. 373–385).

Jaramillo, A., Crawford, B., Soto, R., Villablanca, S.M., Rubio, Á.G., Salas, J., Olguín, E. (2016). Solving the set covering problem with the soccer league competition algorithm. *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. *2016 11th iberian conference on information systems and technologies (cisti)* (pp. 1–4).

Jolai, F., Sangari, M.S., Babaie, M. (2010). Pareto simulated annealing and colonial competitive algorithm to solve an offline scheduling problem with rejection. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *224*(7), 1119–1131.

Jyoti Baruah, A., & Baruah, S. (2022). Pfcmvo: Political fractional competitive multi-verse optimization enabled deep neuro fuzzy network for student performance estimation in spark environment. *International Journal of Modeling, Simulation, and Scientific Computing*, 2250050.

Kamaruzaman, A.F., Zain, A.M., Yusuf, S.M., Udin, A. (2013). Levy flight algorithm for optimization problems-a literature review. *Applied mechanics and materials*, *421*, 496–501.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, *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. *International Journal of Modeling and Optimization*, *1*(4), 289.

Kaveh, A., & Talatahari, S. (2010). Optimum design of skeletal structures using imperialist competitive algorithm. *Computers & Structures*, *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 Resources Management*, *34*(8), 2503–2516.

Kennedy, J., & Eberhart, R. (1995, Nov). Particle swarm optimization. *Proceedings of icnn'95 - international conference on neural networks* (Vol. 4, pp. 1942–1948 vol.4). 10.1109/icnn.1995.488968

Khan, A.H., Ahmed, S., Bera, S.K., Mirjalili, S., Oliva, D., Sarkar, R. (2022). Enhancing the contrast of the grey-scale image based on meta-heuristic optimization algorithm. *Soft Computing*, 1–23.

Khuat, T., & Thi My Hanh, L. (2016, 07). A survey on human social phenomena inspired algorithms. *International Journal* *of Computer Science and Information Security*, *14*, 76–81.

Kiziloluk, S., & Alatas, B. (2015). Automatic mining of numerical classification rules with parliamentary optimization algorithm. *Advances in Electrical and Computer Engineering*, *15*(4), 17–25.

Kiziloluk, S., & Ozer, A.B. (2017). Web pages classification with parliamentary optimization algorithm. *International Journal of Software Engineering and Knowledge Engineering*, *27*(03), 499–513.

Kiziloluk, S., & Özer, A.B. (2019). Hybrid parliamentary optimization and big bang-big crunch algorithm for global optimization. *Turkish Journal of Electrical Engineering & Computer Sciences*, *27*(3), 1954–1969.

Koza, J.R. (1992). *Genetic programming: On the programming of computers by means of natural selection*. Cambridge, MA, USA: MIT Press.

Kumar, M., & Kulkarni, A. (2019, 01). Socio-inspired optimization metaheuristics: A review. In (pp. 241–265). Springer. 10.1007/978-981-13-6569-0_12

Kumar, M., Kulkarni, A., Satapathy, S. (2018, 11). Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology. *Future Generation Computer Systems*, *81*, 252–272.

Kurban, R., Durmus, A., Karakose, E. (2021). A comparison of novel metaheuristic algorithms on color aerial image multilevel thresholding. *Engineering Applications of Artificial Intelligence*, *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. *International Journal of Production Research*, *50*(18), 5117–5131.

Liu, M., Yi, S., Wen, P., Song, H. (2017). Disruption management for predictable new job arrivals in cloud manufacturing. *Journal of Intelligent Systems*, *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. *Information Sciences*, *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 Conversion and Management*, *51*(7), 1407–1411.

Lv, W., He, C., Li, D., Cheng, S., Luo, S., Zhang, X. (2010). Election campaign optimization algorithm. *Procedia Computer Science*, *1*(1), 1377–1386.

Mani, V., Varma, M.D., Krishna, K.V., Khan, Z., Sudabattula, S.K. (2021). Hybrid approach to solve capacitor allocation problem in distribution system using political optimizer algorithm. *International Journal of Engineering Research and Applications*, *11*(5), 28–35.

Manita, G., & Korbaa, O. (2020). Binary political optimizer for feature selection using gene expression data. *Computational Intelligence and Neuroscience*, *2020*.

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. *International*

*Journal of Electrical Power & Energy Systems*, *83*, 104–116.

Mei, W.-b., Lv, W.-g., Chen, K. (2016). Research on feature points matching based on election campaign optimization algorithm. *Mechanical & Electrical Engineering Technology*, 04.

Mirhosseini, M., & Nezamabadi-pour, H. (2018). Bica: a binary imperialist competitive algorithm and its application in cbir systems. *International Journal of Machine Learning and Cybernetics*, *9*(12), 2043–2057.

Mirjalili, S., & Lewis, A. (2013). S-shaped versus v-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, *9*, 1–14.

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & operations research*, *24*(11), 1097–1100.

Mohamadi, S., Sammen, S.S., 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. *Natural Hazards*, *104*(1), 537–579.

Mohammed, A.Q., Al-Anbarri, K.A., Hannun, R.M. (2020a). Introducing newly developed nomadic people optimizer (npo) algorithm to find optimal sizing of a hybrid renewable energy. *Iop conference series: Materials science and engineering* (Vol. 928, p. 022052).

Mohammed, A.Q., Al-Anbarri, K.A., Hannun, R.M. (2020b). Multi-objective npo minimizing the total cost and co 2 emissions for a stand-alone hybrid energy system. *International conference of reliable information and communication technology* (pp. 351–363).

Mohammed, A.Q., Al-Anbarri, K.A., Hannun, R.M. (2020c). 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 and Evolutionary Computation*, *20*, 14–22.

Moosavian, N., & Kasaee Roodsari, B. (2014). Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks. *Swarm and Evolutionary Computation*, *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. *International Journal of Engineering*, *30*(7), 926–936.

Moosavian, N., Roodsari, B.K., et al. (2013). Soccer league competition algorithm, a new method for solving systems of nonlinear equations. *International Journal of Intelligence Science*, *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 Technology*, *1*, 194–198.

Muppidi, S., PG, O.P., et al. (2022). Dragonfly political optimizer algorithm-based rider deep long short-term memory for soil moisture and heat level prediction in iot. *The Computer Journal*.

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 Systems with Applications*, *37*(12), 7615–7626.

Neelima, G., Chigurukota, D.R., Maram, B., Girirajan, B. (2022). Optimal deepmrseg based tumor segmentation with gan for brain tumor classification. *Biomedical Signal Processing and 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 and Evolutionary Computation*, *2*, 1–14.

Norman, M.G., Moscato, P., et al. (1991). A competitive and cooperative approach to complex combinatorial search. *Proceedings of the 20th informatics and operations research meeting* (pp. 3–15).

Nozarian, S., Soltanpoora, H., Jahanb, M.V. (2012). A binary model on the basis of imperialist competitive algorithm in order to solve the problem of knapsack 1-0. *Proceedings of the international conference on system engineering and modeling* (pp. 130–135).

Onwubolu, G.C., & Babu, B.V. (2004). Introduction. In *New optimization techniques in engineering* (pp. 1–12). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-540-39930-8_1 10.1007/978-3-540-39930-8_1

Osmani, A., Mohasefi, J.B., Gharehchopogh, F.S. (2022). Sentiment classification using two effective optimization methods derived from the artificial bee colony optimization and imperialist competitive algorithm. *The Computer Journal*, *65*(1), 18–66.

Pei, J., Song, Q., Liao, B., Liu, X., Pardalos, P.M. (2021). Parallel-machine serial-batching scheduling with release times under the effects of position-dependent learning and time-dependent deterioration. *Annals of Operations Research*, *298*(1), 407–444.

Pekel, E., & Soner Kara, S. (2017). Passenger flow prediction based on newly adopted algorithms. *Applied Artificial Intelligence*, *31*(1), 64–79.

Pooranian, Z., Shojafar, M., Abawajy, J.H., Singhal, M. (2013). Gloa: a new job scheduling algorithm for grid computing. *International Journal of Interactive Multimedia and Artificial Intelligence*, *2*(1), 59–64.

Premkumar, M., Sowmya, R., Jangir, P., Kumar, J.S. (2020). A new and reliable objective functions for extracting the unknown parameters of solar photovoltaic cell using political optimizer algorithm. *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. *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. *International Journal of Pattern Recognition and Artificial Intelligence*, 2252006.

Ramezani, F., & Lotfi, S. (2013). Social-based algorithm (sba). *Applied Soft Computing*, *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, 09). Society and civilization: An optimization algorithm based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation*, *7*(4), 386–396.

Rechenberg, I. (1970). *Optimierung technischer systeme nach prinzipien der biologischen evolution* (Unpublished doctoral dissertation). Technische Universität, Fakultät für Maschinenwissenschaft.

Reynolds, R.G. (1994). An introduction to cultural algorithms. *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. *International Journal of Energy Optimization and Engineering (IJEOE)*, *6*(1), 66–85.

Salih, S.Q., & Alsewari, A.A. (2020). A new algorithm for normal and large-scale optimization problems: Nomadic people optimizer. *Neural Computing and Applications*, *32*(14), 10359–10386.

Sangeetha Francelin, V.F., Daniel, J., Velliangiri, S. (2022). Intelligent agent and optimization-based deep residual network to secure communication in uav network. *International Journal of Intelligent Systems*.

Sathasivam, S., Mansor, M., Kasihmuddin, M.S.M., Abubakar, H., et al. (2020). Election algorithm for random k satisfiability in the hopfield neural network. *Processes*, *8*(5), 568.

Selvakumar, A.I., & 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, 12). Effect of group formation on performance, task management and social loafing. *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 transactions*, *52*(1), 88–95.

Shah, K., Bhattacharjee, K., Godwal, S. (2020). Solving of economical load dispatch using efficient group leader optimization technique. *Technologies for sustainable* *development: Proceedings of the 7th nirma university international conference on engineering (nuicone 2019), november 21-22, 2019, ahmedabad, india* (p. 181).

Shakya, H.K., 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. *International Journal of Scientific and Technical Advancements*, *4*(1), 109–112.

Shayeghi, H., & Esmaeli, H. (2013). Eco based solution for generation expansion planning in deregulated power systems. *Electrical and Electronic Engineering*, *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 levelized cost of hybrid wind-solar-diesel-battery system using political optimizer. *2020 ieee first international conference on smart technologies for power, energy and control (stpec)* (pp. 1–6).

Srilakshmi, K., Babu, P.R., Venkatesan, Y., Palanivelu, A. (2022). Soccer league optimization for load flow analysis of power systems. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, *35*(2), e2965.

Suresh, V., Jasinski, M., Leonowicz, Z., Kaczorowska, D., Reddy K, H., et al. (2021). Political-optimizer-based energy-management system for microgrids. *Electronics*, *10*(24), 3119.

Taher, S.A., Fini, M.H., Aliabadi, S.F. (2014). Fractional order pid controller design for lfc in electric power systems using imperialist competitive algorithm. *Ain Shams Engineering Journal*, *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 Annals*, *65*(1), 9–12.

Tizhoosh, H.R. (2005). Opposition-based learning: a new scheme for machine intelligence. *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, M.A., & Tulsky, V.N. (2021). Integration of dgs optimally to enhance the voltage profile and stability index of distribution grid. *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. *Mechanical & Electrical Engineering Technology*, 03.

Varki, A., & Altheide, T.K. (2005). Comparing the human and chimpanzee genomes: searching for needles in a haystack. *Genome research*, *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. *International Journal of Production Research*, *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. *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. (2010). Election campaign optimization algorithm for multi-peak optimization problems. *2010 third international conference on information and computing* (Vol. 3, pp. 154–157).

Whitley, D., Gordon, V.S., Mathias, K. (1994). Lamarckian evolution, the baldwin effect and function optimization. *International conference on parallel problem solving from nature* (pp. 5–15).

Wolpert, D.H., Macready, W.G., et al. (1995). *No free lunch theorems for search* (Tech. Rep.). Technical Report SFI-TR-95-02-010, Santa Fe Institute.

Wolpert, D.H., Macready, W.G., et al. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, *1*(1), 67–82.

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. *Central European Journal of Operations Research*, *22*(4), 663–685.

Xie, Q., Lv, W., Liu, Z., Zhang, X., Luo, S., Cheng, S. (2010). Constrained optimization with election campaign algorithm. *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. *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. *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. *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. *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. *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. *2016 6th international conference on mechatronics, computer and education informationization (mcei 2016).*

Xie, Q.H., Zhang, X.W., Lv, W.G., Cheng, S.Y. (2016). Stereo matching based on election campaign algorithm. *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. *2016 6th international conference on advanced design and manufacturing engineering (icadme 2016)* (pp. 815–820).

Xie, Q.H., Zhang, X.W., Lv, W.G., Cheng, S.Y., Huang, H.X., Cai, S.D. (2013). Research on re-use reverse logistics network of disused electric appliances based on election campaign algorithm. *Applied mechanics and materials* (Vol. 397, pp. 1121–1128).

Xie, Q.H., Zhang, X.W., Lv, W.G., Cheng, S.Y., Lin, C.T., Yang, Y.J. (2013). Quantity prediction and optimal disposal capacity of waste electric appliances based on election campaign algorithm. *Applied mechanics and materials* (Vol. 411, pp. 2395–2399).

Xu, B., Heidari, A.A., Kuang, F., Zhang, S., Chen, H., Cai, Z. (2022). Performance optimization of photovoltaic systems: Reassessment of political optimization with a quantum nelder-mead functionality. *Solar Energy*, *234*, 39–63.

Yıldız, B.S., Pholdee, N., Bureerat, S., Erdaş, M.U., Yıldız, A.R., Sait, S.M. (2021). Comparision of the political optimization algorithm, the archimedes optimization algorithm and the levy flight algorithm for design optimization in industry. *Materials Testing*, *63*(4), 356–359.

Yousefi, M., Darus, A., Mohammadi, H. (2012). An imperialist competitive algorithm for optimal design of plate-fin heat exchangers. *International Journal of Heat and Mass Transfer*, *55*(11), 3178–3185.

Yousri, D., Abd Elaziz, M., Oliva, D., Abualigah, L., Al-qaness, M.A., Ewees, A.A. (2020). Reliable applied objective for identifying simple and detailed photovoltaic models using modern metaheuristics: Comparative study. *Energy Conversion and Management*, *223*, 113279.

Zargarnataj, M. (2007). New election algorithm based on assistant in distributed systems. *Computer systems and applications, 2007. 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. *Mechanical & Electrical Engineering Technology*, 09.

Zhang, H., Lv, W.G., Cheng, S.Y., Luo, S.M., Zhang, X.W. (2011). Election campaign optimization algorithm for design of pressure vessel. *Advanced materials research* (Vol. 308, pp. 15–20).

Zhang, L., Lv, W., Wang, C., Cheng, S., Luo, S. (2011). Parameter design and performance study on election campaign optimization algorithm. *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.

# Statements and Declarations

**Funding:** The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.
**Conflict of interests:** The authors have no relevant financial or non-financial interests to disclose.
**Author contributions:** All authors contributed equally.
**Data availability:** This manuscript has no associated data.