Future Generation Computer Systems 81 (2018) 252–272

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology

Meeta Kumar [a], Anand J. Kulkarni [a,b,*], Suresh Chandra Satapathy [c]

[a] Symbiosis Institute of Technology, Symbiosis International University, Pune, MH 412 115, India
[b] Odette School of Business, University of Windsor, 401 Sunset Avenue, Windsor, ON N9B3P4, Canada
[c] Department of Computer Science and Engineering, PVP Siddhartha Institute of Technology, Vijayawada, AP, India

ARTICLE INFO

Article history:
Received 31 July 2017
Received in revised form 29 September 2017
Accepted 29 October 2017
Available online 15 November 2017

Keywords:
Cultural evolution
Evolutionary algorithm
Metaheuristic
Nature-inspired computing
Socio-inspired optimization
Unconstrained optimization

ABSTRACT

The paper proposes a novel metaheuristic Socio Evolution & Learning Optimization Algorithm (SELO) inspired by the social learning behaviour of humans organized as families in a societal setup. This population based stochastic methodology can be categorized under the very recent and upcoming class of optimization algorithms—the socio-inspired algorithms. It is the social tendency of humans to adapt to mannerisms and behaviours of other individuals through observation. SELO mimics the socio-evolution and learning of parents and children constituting a family. Individuals organized as family groups (parents and children) interact with one another and other distinct families to attain some individual goals. In the process, these family individuals learn from one another as well as from individuals from other families in the society. This helps them to evolve, improve their intelligence and collectively achieve shared goals. The proposed optimization algorithm models this de-centralized learning which may result in the overall improvement of each individual's behaviour and associated goals and ultimately the entire societal system. SELO shows good performance on finding the global optimum solution for the unconstrained optimization problems. The problem solving success of SELO is evaluated using 50 well-known boundary-constrained benchmark test problems. The paper compares the results of SELO with few other population based evolutionary algorithms which are popular across scientific and real-world applications. SELO's performance is also compared to another very recent socio-inspired methodology—the Ideology algorithm. Results indicate that SELO demonstrates comparable performance to other comparison algorithms. This gives ground to the authors to further establish the effectiveness of this metaheuristic by solving purposeful and real world problems.

1. Introduction and motivation

Optimization methods play a vital role in solving engineering problems. The exact optimization methods or deterministic methods may not be computationally efficient in solving complex non-linear and multimodal problems that exist in most real-world applications [1,2]. In the past few decades researchers have resorted to a number of methodologies inspired from biological and natural systems have been proposed for solving complex optimization problems. By far the majority of nature-inspired algorithms [3] are based on certain characteristics of biological system [4,5]. Recent literature [6,7] advocate that the largest number of nature-inspired algorithms is categorized as bio-inspired algorithms. Closer to the bio-inspired algorithms are the swarm based algorithms [8,9] which seek inspiration from the collective or swarm behaviour exhibited by a number of animal species. The collective intelligence as exhibited by a swarm towards achievement of shared goal forms the key idea behind swarm-based algorithms. A large number of homogeneous agents in an environment function through mutual cooperation to achieve desired goal. Another popular category is the Evolutionary Algorithms (EA) which may be categorized under bio-inspired algorithms and seek inspiration from the processes of biological evolution and natural selection. Few examples include genetic algorithms [8], ant colony optimization [10] based on the interaction of ants, fish school search [11] based on the collective behaviour of fishes in order to survive and many others. Many algorithms find motivation from physical and chemical systems too, as reviewed by Fister Jr. et al. [5] and Biswas et al. [12]. A few examples include, but not limited to Simulated Annealing [13–15] and Harmony Search [16,17] which find their base in physical processes; while Chemical Reaction Optimization [18] is an example of chemical-inspired methodology. A very upcoming and emerging

* Corresponding author at: Odette School of Business, University of Windsor, 401 Sunset Avenue, Windsor, ON N9B3P4, Canada.
E-mail addresses: meeta.kumar@sitpune.edu.in (M. Kumar), anand.kulkarni@sitpune.edu.in, kulk0003@uwindsor.ca (A.J. Kulkarni), sureshsatapathy@ieee.org (S.C. Satapathy).

https://doi.org/10.1016/j.future.2017.10.052
0167-739X/

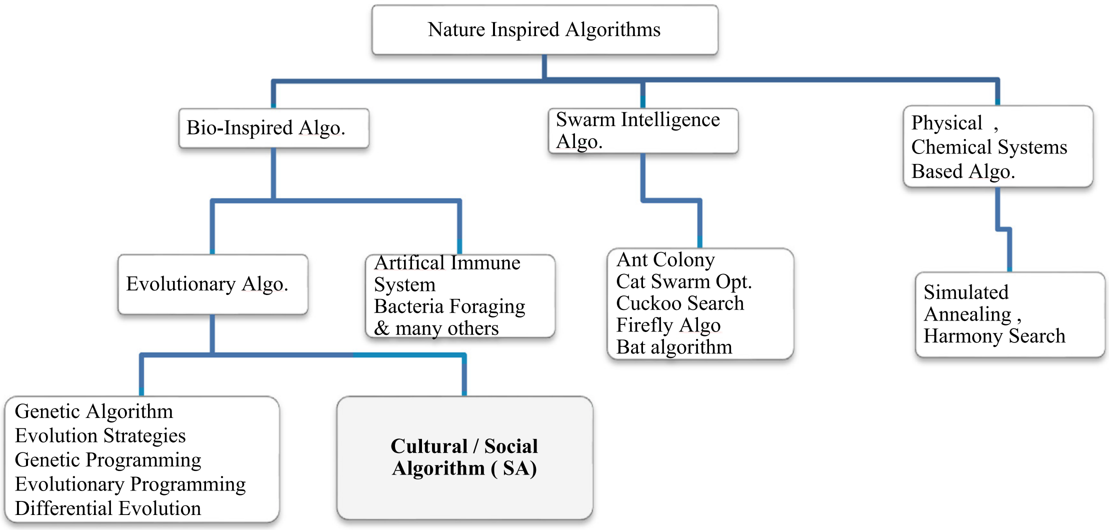

Fig. 1. Broad classification of nature-inspired algorithms.

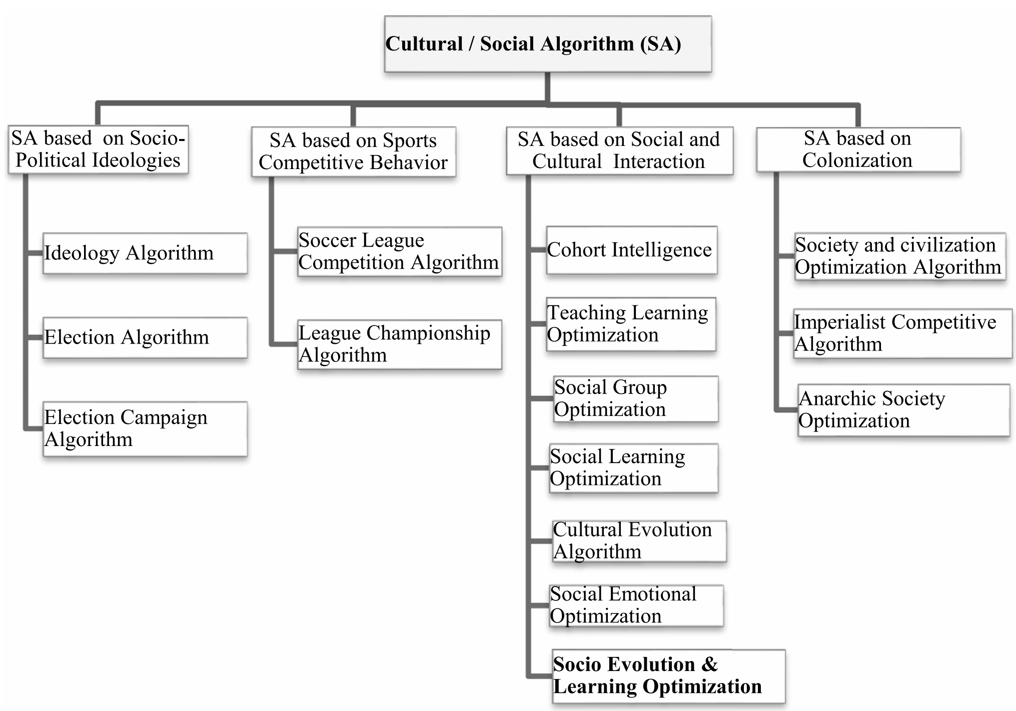

Fig. 2. Broad classification of socio-inspired algorithms.

Table 1
A list of Socio-Inspired Algorithms.

Algorithm	Author	Year, Reference	Algorithm	Author	Year, Reference
Ideology Algorithm	Huan et al.	2016, [54]	Teaching Learning Based Optimization	Rao et al.	2012, [47]
Social Learning Optimization	Liu et al.	2016, [52]	Anarchic Society Optimization	Ahmadi	2011, [46]
Social Group Optimization	Satapathy et al.	2016, [53]	Social Emotional Optimization Algorithm	Xu et al.	2010, [43]
Election Algorithm	Emami et al.	2015, [51]	Election Campaign Optimization Algorithm	Lv et al.	2010, [44]
Soccer League Competition Algorithm	Moosavian et al.	2014, [21]	League Championship Algorithm	Kashan	2009, [40]
Cultural Evolution Algorithm	Kuo et al.	2013, [69]	Imperialist Competitive Algorithm	Atashpaz et al.	2007, [38]
Cohort Intelligence	Kulkarni et al.	2013, [70]	Society And Civilization	Ray et al.	2003, [37]

branch of EA is the Socio-Inspired class of algorithms, which take motivation from the social and cultural interactions seen in human behaviour. A detailed review of this class of algorithms is presented in the further section. Thus the existing nature-inspired algorithms may be briefly categorized into following broad or major categories: swarm intelligence based, bio-inspired, physical /chemical

Table 2
Benchmark functions used in the experiments (U—unimodal, M—multimodal, S—separable, N—non-separable, low and up are limitations of search space).

Fnc#	Name	Type	Low	Up	Dim	Fnc#	Name	Type	Low	Up	Dim
F1	Foxholes	MS	−65.536	65.536	2	F26	Michalewicz2	MS	0	3.1416	2
F2	Goldstein–Price	MN	−2	2	2	F27	Michalewicz5	MS	0	3.1416	5
F3	Penalized	MN	−50	50	30	F28	Michalewicz10	MS	0	3.1416	10
F4	Penalized2	MN	−50	50	30	F29	Perm	MN	−4	4	4
F5	Ackley	MN	−32	32	30	F30	Powell	UN	−4	5	24
F6	Beale	UN	−4.5	4.5	5	F31	Powersum	MN	0	4	4
F7	Bohachevsky1	MS	−100	100	2	F32	Quartic	US	−1.28	1.28	30
F8	Bohachevsky2	MN	−100	100	2	F33	Rastrigin	MS	−5.12	5.12	30
F9	Bohachevsky3	MN	−100	100	2	F34	Rosenbrock	UN	−30	30	30
F10	Booth	MS	−10	10	2	F35	Schaffer	MN	−100	100	2
F11	Branin	MS	−5	10	2	F36	Schwefel	MS	−500	500	30
F12	Colville	UN	−10	10	4	F37	Schwefel_1_2	UN	−100	100	30
F13	Dixon–Price	UN	−10	10	30	F38	Schwefel_2_22	UN	−10	10	30
F14	Easom	UN	−100	100	2	F39	Shekel10	MN	0	10	4
F15	Fletcher	MN	−3.1416	3.1416	2	F40	Shekel5	MN	0	10	4
F16	Fletcher	MN	−3.1416	3.1416	5	F41	Shekel7	MN	0	10	4
F17	Fletcher	MN	−3.1416	3.1416	10	F42	Shubert	MN	−10	10	2
F18	Griewank	MN	−600	600	30	F43	Six-hump camelback	MN	−5	5	2
F19	Hartman3	MN	0	1	3	F44	Sphere2	US	−100	100	30
F20	Hartman6	MN	0	1	6	F45	Step2	US	−100	100	30
F21	Kowalik	MN	−5	5	4	F46	Stepint	US	−5.12	5.12	5
F22	Langermann2	MN	0	10	2	F47	Sumsquares	US	−10	10	30
F23	Langermann5	MN	0	10	5	F48	Trid6	UN	−36	36	6
F24	Langermann10	MN	0	10	10	F49	Trid10	UN	−100	100	10
F25	Matyas	UN	−10	10	2	F50	Zakharov	UN	−5	10	10

inspired, and others (like social, cultural algorithms). Fig. 1 details the broad classification of these algorithms. The authors' further attempt to sub-classify the Social Algorithms in Fig. 2 based on the broad ideas from which they draw inspiration. A list of Socio Inspired algorithms is also presented in Table 1. All these methods have gained popularity because of their use of simple rules for searching for optimal solutions to complex and real-world computational problems. Metaheuristics [19] is a term used to refer to such general algorithmic framework which seek inspiration from various phenomenon observed in nature. Metaheuristic strategies combine rules and a certain degree of randomness to find optimal (or near optimal) solutions to problems and thus can be applied to a variety of optimization problems. They are general approximate strategies [2] and can be adapted to solve a wide variety of optimization problems with little changes to their general algorithmic framework. The popularity of metaheuristics in solving real world optimization problems can be attributed to their following characteristics: simple design owing to their closeness to natural concepts, less problem specific, faster problem solving ability and ability to scale large dimension problems. The development of these metaheuristics has been a thrust area in the field of computational intelligence for decades now, with newer class of algorithms fast evolving, with well-accepted algorithms like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Algorithms and socio algorithms like Cohort Intelligence among others.

Originally proposed by Kennedy and Eberhart in 1995, PSO [20] is based on the behaviour exhibited by flocking birds or a school of fish. In PSO [22], swarm of birds/fish representing the candidate solutions travel through the sample space driven by their own and best performances of their neighbours. Although popular, PSO suffers from getting stuck in local minima and premature convergence for certain complex problems. The comprehensive learning PSO (CLPSO) [23] and PSO2011 [24,25] are the improved variants of the standard PSO partially overcoming this limitation. CLPSO maintains the diversity of the swarm by allowing a particle to update its velocity by using previous best positions of other particles. Each dimension of a particle seeks learning from a different particle. The EA draw inspiration from the 'survival of the fittest' analogy as stated in the theory of natural selection by Charles Darwin. EA's encompass the very popular GA and the others such as Evolution Strategies, Genetic Programming, Evolutionary Programming and Differential Evolution; as elaborated in the article by Nanda and Panda [6]. Covariance matrix adaptation evolution strategy (CMAES) [26,27] is a powerful EA for real-valued optimization based on self-adaptation in evolution strategies. The artificial bee colony algorithm (ABC) [28,29] is an optimizer based on the collaborative and intelligent behaviour of a swarm of honey bees. It mimics the foraging behaviour of swarm of bees which comprises of employed bees, onlooker and scout bees. It is a powerful metaheuristic which effectively employs four different selection processes: a global selection process, local selection process carried out by employed and onlooker bees, a local greedy selection process to memorize the best candidate solution and a random selection process carried out by the scouts. Differential evolution (DE) algorithm, proposed by Storn and Price [30] is a powerful yet simple population-based optimization technique mimicking the basic rules of genetics, yet different from the basic Genetic algorithm. It employs [31] five mutation and two crossover strategies. The Adaptive DE algorithm (JDE) [32] and the Self-Adaptive Differential Evolution algorithm (SADE) [33,34] are modified versions of the DE. The JDE provides a modified DE algorithm with self-tuned control parameters. In SADE, both, the control parameters as well as the learning strategies of DE are self-adapted during the evolution phase. Another optimization methodology is the Backtracking Search Algorithm (BSA) [35] exploits certain genetic operators such as selection, mutation and crossover. However it uses a random mutation approach where an individual from the previous population is retained, which helps it to seek important learning from the experiences of the previous population. Also, the crossover strategy of BSA is more complex than the crossover strategies used across the various derivative forms of GA. Occasionally new metaheuristics are introduced which use a novel metaphor as guide for solving optimization problems. Socio-inspired algorithms are one such very recent and upcoming class of optimization algorithms, which use the idea of simulating and mimicking social learning of humans (or social evolution). The notion of Cultural Algorithms [36] was first introduced by Reynolds as early as 1994 which states that individuals evolve much faster through cultural evolution than through biological or genetic evolution alone. Humans adapt to mannerisms and behaviours by observing/imitating other individuals which helps them improve their intelligence quickly and achieve shared goals. The tendency to cooperate and function together as a cohesive group adds to their collective intelligence. This idea has formed the ground for many a researchers to formalize some recent socio-inspired algorithms like Society and Civilization Optimization algorithm (SCO), Imperialist Competitive algorithm (ICA), League Championship algorithm (LCA), Social Emotional Optimization algorithm (SEOA), Election Campaign Optimization algorithm (ECO), Anarchic Society Optimization algorithm (ASO), Teaching–learning-based optimization (TLBO), Cultural Evolution algorithm (CEA), Cohort Intelligence (CI), Soccer League Competition Algorithm (SLC), Election Algorithm, Social learning optimization (SLO), Social Group Optimization (SGO) and Ideology Algorithm (IA) etc.

SCO [37] is inspired from human social behaviour seen among society individuals. The individuals in a society interact with one another to improve their overall behaviour and a cooperative interaction among such societies represents a civilization. The ICA [38,39] was originally proposed in 2007, simulates the socio-political behaviours seen across imperialist nations which compete to take possession of weaker colonies or empires. This imperialist competition finally results in strengthening the power of stronger and successful imperialist empires; whilst the weaker empires collapse gradually finally leading to a state of convergence. LCA established by [40–42] is derived from the competition amongst teams seen in league matches. Artificial teams (representing solutions) compete across weeks (representing iterations) based on a league schedule and a strong team (with higher fitness value) gradually emerges as the winner at the end of the playing season (stopping condition). SEOA [43] is a swarm based socio-inspired metaheuristic which simulates a virtual individual person who wishes to achieve a higher status in the society and his decisions are guided by his emotion. An emotional index determines his current behaviour and the society determines whether his current behaviour is better or worse, thus affecting this emotion index value (parameter). The authors Lv et al. [44] introduced the ECO algorithm based on the mannerisms of political candidates during an election campaign. The voters are inspired to vote for a candidate with better prestige (better function value) and finally a stronger election candidate wins the highest supports from voters [45]. Another optimization algorithm which seeks inspiration from a very commonly observed human behaviour of being greedy and disorderly to achieve their goals is ASO algorithm [46]. Members behave anarchically to find better solutions in the solution space. TLBO [47] an optimization method mimics the influence of a teacher on the learning outcome of it students. The methodology works in two parts: the teacher phase and the learner phase, where the learners increase their knowledge (improve solution quality) by imbibing knowledge from the teacher as well through interaction with their peer learners. Kuo and Lin presented a framework for Cultural Evolution Algorithm in 2013. They state that a species learns or evolves through one of these modes: group consensus, individual learning, innovative learning and self-improvement and in their research mathematically model each of these learning modes. CI [48,49] a successful socio-inspired metaheuristic mimics the self-learning behaviour exhibited by candidates in a group, where the candidates cooperate and compete with one another to achieve some individual goal. SLC [21,50] has been effectively applied to solve discrete and continuous optimization problems and takes inspiration from competitive behaviour seen among teams and players in soccer league matches. Election Algorithm [51], yet another optimization and search algorithm, tries to mimic the political elections comprising the electoral parties, the candidates and the voters. SLO [52] which proposes an optimization methodology where individuals evolve at different levels: one is genetic evolution and the other is cultural evolution through imitation and observation. This cultural evolution also in turn influences the genetic evolution in future generations to come and helps accelerate human intelligence. A very recent algorithm, the Social Group Optimization [53] takes motivation from the human behaviour exhibited when trying to collectively attempt to solve a complex task at hand. This mannerism of getting influenced by a better person; modifying his behaviour accordingly helps address complex problems. Another innovative algorithm IA [54] is inspired from the idea how certain beliefs become the guide for individuals in a society to achieve their goals. IA elicits this idea through a political scenario where individuals follow their political ideologies and compete with members of their political party as well as with leaders of other political parties in their will to excel. It can thus be seen from the literature that a number of popular and efficient optimization algorithms have been proposed and will continue to develop; however in agreement with 'no free lunch in optimization' [55,56] it is true that one particular optimization algorithm may be more suited to some optimization problem better than others. It underscores that a universal algorithm/method does not exist for all optimization problems and that on an average will perform equally well. Typically for a defined problem with specific objective functions there will be a class of optimization algorithms that will outperform some others. The major task at hand is only to identify these better performing algorithms which are specialized to the structure of specific optimization problem under consideration. Therefore, it can be stated that developing new optimization algorithms will be always essential and significant and gives the authors' a ground that there will be always a scope to develop newer prospective algorithms that could be well-suited to some specific class of optimization tasks and may as well surpass a few other already existing algorithms for solving some specific optimization problems [57].

In the current research, the authors propose a novel socio-inspired optimization methodology referred to as Social Evolution & Learning Optimization algorithm. The metaheuristic is motivated and draws inspiration from the social behaviour exhibited by individuals in a family which is a part of a human societal setup. A family represents an elementary social group in a society typically consisting of parents and their children and a society can be visualized as a multi-agent setup of different families coexisting together. According to Goldsmith [58] 'a family in its various forms is the basis of all human societies and social structures'. Each family member can be thought of as an individual agent in a family, making its own behavioural choices inspired by observing and learning from others. The evolution of parents and children of a family is based on learning from one another as well as from other families. This de-centralized learning may result in the overall improvement of each agent's behaviour and associated goals and ultimately the entire societal system. The proposed optimization algorithm models the above rationale of 'decentralized learning behaviour' of the family individuals who collectively evolve their family. Thus it may be stated that SELO takes inspiration and mimics natural and social tendency of humans organized as family groups, where individual family members (parents and kids) interact with one another and other distinct families to achieve some individual/shared goals. A kid is genetically similar to its parent and exhibits similar behaviour (fitness); however may later be influenced by the social behaviour of his peers. Unlike traditional GA's, the proposed methodology provides for a two-way evolution; allowing for a two way system of learning. In SELO, evolution and learning takes place at both: social level and population level (for every individual).

In a multi-agent environment [59], the algorithm tries to simulate the behaviour of agents (family members) who work in a coordinated way, optimizing their local utilities and contributing the maximum towards optimization of the global objective *(betterment of the society as a whole*). According to Hechter and Horne

[60] 'people must coordinate their actions and cooperate to attain common goals'. SELO follows this very notion and uses intra and intersociety interactions observed within human society to converge to an optimal solution of the optimization problem at hand. In the context of human society, one family unit may be inspired or motivated to follow and pursue the qualities of the individuals of the other families, which in-turn, may result in the improvement of its own qualities and the associated behaviour. The parents from a family, living in a societal setup may be influenced by the positive/negative qualities and behaviour of the parents from other families. Also, all the individuals of a family typically may exhibit similar behaviour and have similar choices and likings. Qualities of the parents which construct their social behaviours is typically assimilated or followed by their children also. So during the process of socialization, every member of the family tries to advance its own behaviour by either observing its own family or other families. When a family member attempts to follow a given behaviour characterized by certain qualities, it often adopts such qualities in a manner that may improve its own behaviour and associated goals. This socio-behavioural model using SELO enabled to solve a multitude of optimization test problems with diverse properties so that the usefulness of the proposed method can be truly tested in an unbiased way. The authors compare the problem solving success and robustness of the proposed methodology with some popular and well-established metaheuristics [61] like PSO *(its variants PSO2011 and CLPSO*), CMAES, ABC and BSA as well with newer and promising socio-inspired methodology like IA. This gives a clear picture of how the proposed algorithms SELO fairs at solving numerical optimization problems in comparison to varied classes of other evolutionary algorithms.

The remainder of the paper is organized as follows. Section 2 discusses the exact methodology and algorithmic framework for the proposed technique and how the idea can be applied to solving optimization problems. Section 3 tabulates the computational results obtained and discusses the performance of SELO and other comparator algorithms. Finally, Section 4 concludes the paper and suggests future directions for researchers.

In this way, parents and kid from a family may learn from one another which may help the entire society evolve and improve.

2. Methodology : Socio evolution & learning optimization Algorithm (SELO)

The proposed iterative algorithm is population based, which initially starts its search and optimization process with a population of solutions. Akin to other population-based designs, SELO attempts to direct the population of possible solutions towards the more promising areas of the solution space in search for optimal solution. In the context of SELO, the behaviour of an individual belonging to a family represents each such solution. Each family comprises of individuals or family members, who can either, be a parent or a kid. Thus members of the immediate family include two parents, and sons and/or daughters (referred to as kids in the algorithm). Thus population comprises a set of multiple families.

This social framework comprising families and a cluster of families (a society) forms the basis of our proposed socio-inspired SELO. In the context of an optimization problem, the objective function represents the behaviour $f(\mathbf{x})$ of a family individual defined as$f(\mathbf{x}) = f(x_1, \ldots, x_n, \ldots, x_N)$, and the variables $\mathbf{x} = (x_1, \ldots, x_n, \ldots, x_N)$ represent the qualities of each family individual. We present certain characteristics observed in a societal setup of human families to describe the artificial Socio Evolution & Learning Algorithm which will be explained in the further sections of the paper.

- An individual represents a basic element of a society.
- A certain group of individuals coexist as a family.
- A society can be thought of as a group of individuals or group of families involved in continual social interaction.
- Every family in the population (considering a typical setup) comprises of individuals or family members, who are either a parent or a kid. Thus members of the immediate family may include two parents (grownups), and their offspring (sons and/or daughters).
- The grownups or elders follow behaviours and get influenced (evolve) by the mannerisms of other individuals, during the course of social interaction with other families.
- Families provide initial socialization for children that shape their attitudes, values and behaviours.
- A kid is genetically similar to its parents and will imbibe their behaviour in the earlier years and exhibits similar fitness; however at behaviour level may later evolve differently inspired by social behaviour of others (peers).
- In the later or their growing years, children are greatly influenced by their peers and assimilate a lot of peer behaviour in social settings in order to be accepted by their peers [62].

Thus every human (an elder or a child) constantly learns through observation and imitation. This idea is strongly supported through the social learning theory proposed by Bandura [63] and Bandura and Walters [64] that human behaviour is greatly due to genetic as well as physiological factors and is achieved primarily through learning. The acquired experience environment and cultural influence greatly impacts his behaviour. Various social researches and studies have also found that parents influence at-home behaviour of children [65], and peers influence behaviour outside home or their social behaviour. This social learning accumulated from his parents and his peers guide an individual's genetic evolution. As characterized from above, the SELO algorithm will begin with the creation of initial population in some way, e.g. through randomization, with some initial number of families; every family consisting of two parents and kids. A set of parents from each of the initialized families generates their set of behaviour randomly in close neighbourhood of each other. Initially, every kid belonging to a family generates its behaviour function correspondingly in the neighbourhood of their parents. The algorithm then progresses iteratively where members from each family may decide to follow the behaviour of its own family or other families. In the current study, the choice to follow is decided through a roulette wheel approach [66,67] thus giving a fair chance to every behaviour in the population to get selected purely based on its quality. A roulette wheel selection (RWS) operator is used to recommend a fitter or better behaviour to either a parent or a kid. The parent or the kid may then choose to follow this behaviour or fitness (detailed in step 4 of the algorithm). RWS is a fitness proportionate selection where the selection mechanism abandons none of the individuals in the population and every individual gets a fair opportunity to get selected, also ensuring some level of diversity in the population of solutions. Each individual is assigned a share in the roulette wheel which is proportional to its fitness value [68]. Then probabilistically roulette wheel returns the selected individual, with more probability that fitter individuals may be chosen as compared to individuals with poor fitness. This selection algorithm thus provides a zero bias. It also helps the algorithm jump out of possible local minima.

The concept of SELO is represented in Fig. 3. Certain fitter individuals (with better behaviour) are selected based on a fitness function evaluation which guides other individuals to gradually improve their behaviour. Knowledge of fitter individuals thus guides other individuals in the society across iterations through an influence or 'follow' function. The population could be assumed to become successful when for a considerable number of learning

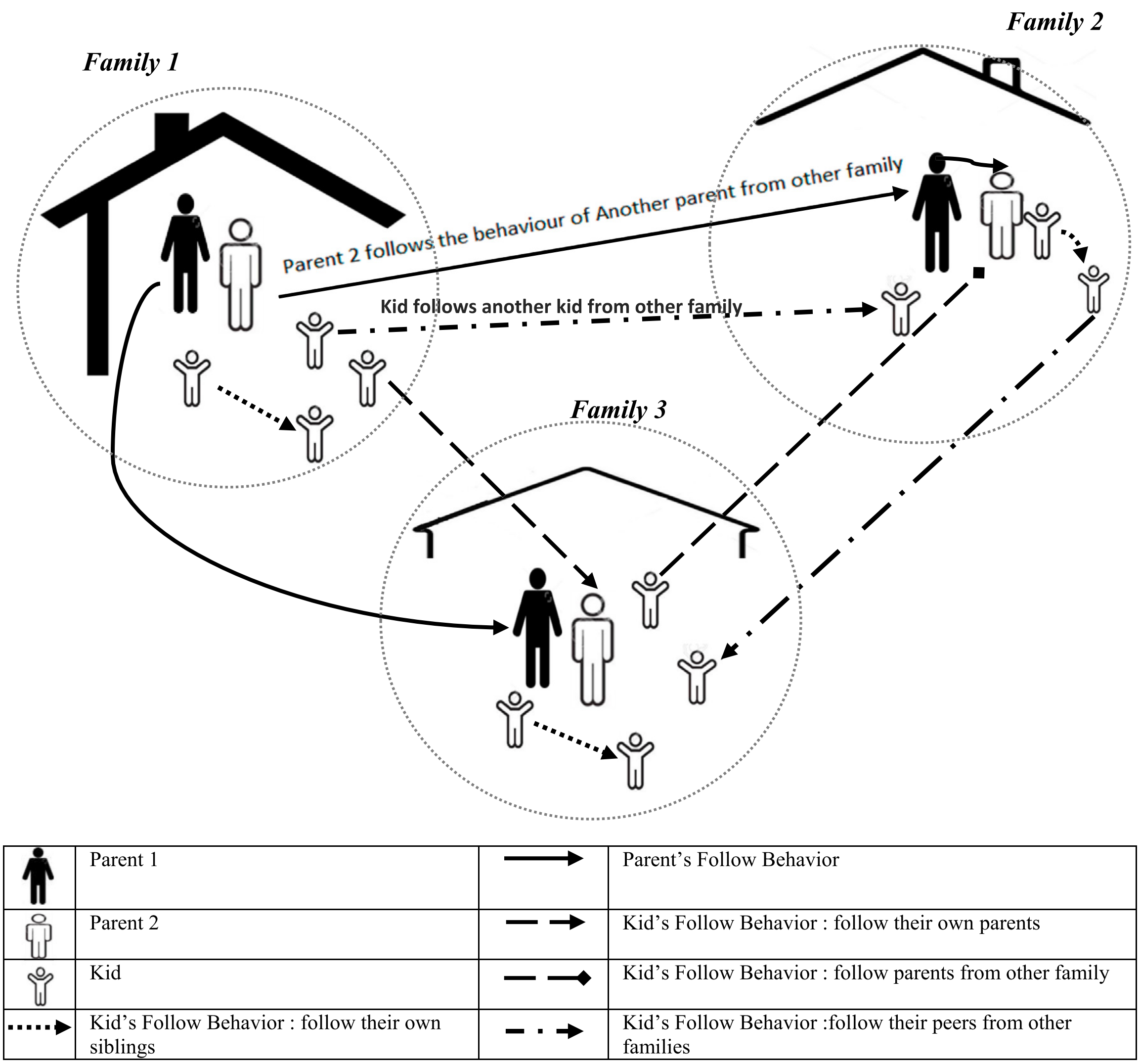

Fig. 3. Schematic representation of SELO.

attempts the families' behaviour saturates to the same behaviour and there would be no improvement in the behaviour of every individual member in the families. The basic steps and mathematical formulation of the algorithm are presented in Section 2.1.

2.1. Steps and flowchart of the SELO Algorithm

Consider a generalized unconstrained problem (in minimization sense) as follows:

Minimize $f(\mathbf{x}) = f(x_1, \ldots, x_n, \ldots, x_N)$

Subject to $\varphi_n^{lower} \le x_n \le \varphi_n^{upper}$, $n = 1, \ldots, N$

The SELO procedure begins with initialization of number of families M where every family $F_i, (i = 1, \ldots, M)$ consists of two parents $F_i^{P_1}, F_i^{P_2}$ and kids $F_i^{K_j}$ $(i = 1, \ldots, M), (j = 1, \ldots, O)$, i.e. $F_i \in \left\{F_i^{P_1}, F_i^{P_2}, F_i^{K_1}, \ldots, F_i^{K_j}, \ldots, F_i^{K_O}\right\}$. Every individual $s \in \{P_1, P_2, K_1, \ldots, K_j, \ldots, K_O\}$ of each family $\boldsymbol{F}_i$, $(i = 1, \ldots, M)$ generates variables/qualities, $y = \left(x_1^{F_i^s}, \ldots, x_n^{F_i^s}, \ldots, x_N^{F_i^s}\right)$, $n = 1, \ldots, N$ which makes its behaviour $f(\mathbf{x}^{F_i^s})$. The individual behaviour of every individual s of each family $\boldsymbol{F}_i$, $(i = 1, \ldots, M)$ is generally being observed by every individual from the same as well as other families. This urges every family individual to follow the behaviour better than its current behaviour. More specifically, s may follow $\widehat{s}$ iff $f\left(\mathbf{x}^{F_i^{\widehat{s}}}\right) < f\left(\mathbf{x}^{F_i^s}\right)$, $s \neq \widehat{s}$, s and $\widehat{s} \in \{P_1, P_2, K_1, \ldots, K_j, \ldots, K_O\}$.

In addition, initialize convergence parameter ε, sampling interval reduction factor $r \in [0, 1]$, parent_follow_probability $r_p \in [0, 1]$ to a significantly large value and peer_follow_probability $r_k \in [0, 1]$ to a small value, learning iteration counter c, learning attempt counter $l = l_{max}$ which repeats the 'c' iterative trials till required learning behaviour converges. Tuning the parameters $r_p \in [0, 1]$ and $r_k \in [0, 1]$ allows the choice of whether the kids follow the behaviour of their parents or their peers. Values of parameters r_p, r_k will be reduced across iterations which signify that the kids gradually tend to get influenced or follow the behaviour of their compeers (i.e. their own siblings or kids from other families).

Step 1 (Initialize families and objective function evaluation)

Every parent $F_i^{P_1}$ and $F_i^{P_2}$ associated with family F_i $(i = 1, \ldots, M)$ randomly generates its set of qualities/variables $\mathbf{x}^{F_i^{P_1}} = \left(x_1^{F_i^{P_1}} \ldots x_n^{F_i^{P_1}}, \ldots, x_N^{F_i^{P_1}}\right)$ and $\mathbf{x}^{F_i^{P_2}} = \left(x_1^{F_i^{P_2}} \ldots x_n^{F_i^{P_2}}, \ldots, x_N^{F_i^{P_2}}\right)$ in the close neighbourhood of each other, from within the corresponding sampling interval $\left[\varphi_n^{lower}, \varphi_n^{upper}\right]$, $n = 1, \ldots, N$ and further evaluates associated objective function/behaviour $f\left(\mathbf{x}^{F_i^{P_1}}\right)$

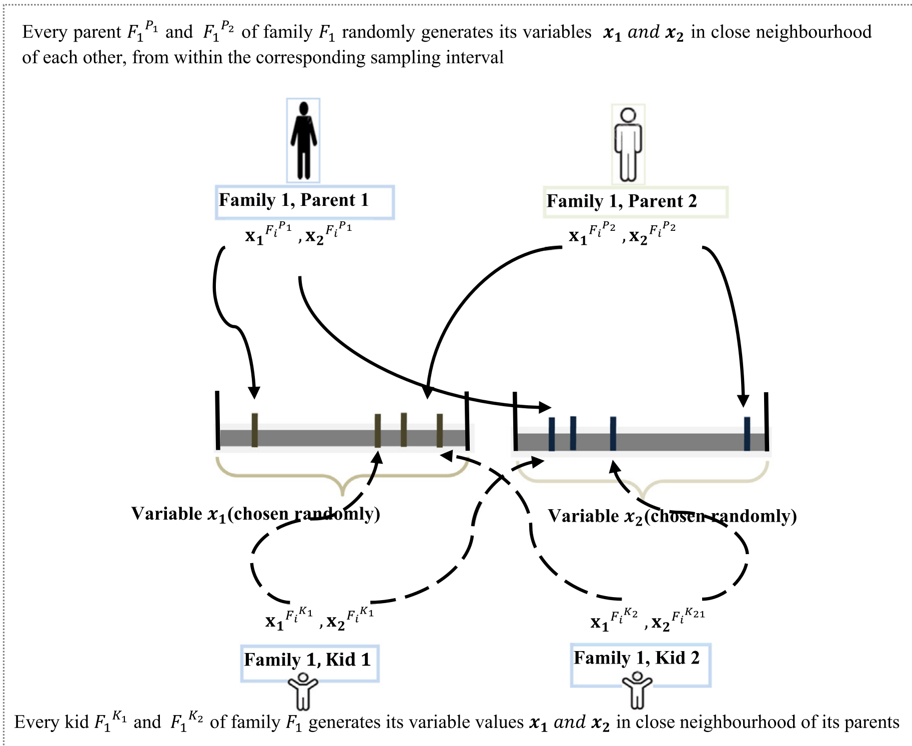

Fig. 4. Generation of variable values for a family.

and $f\left(\mathbf{x}^{F_i^{P_2}}\right)$, respectively. '$N$' is the dimension of the optimization problem being solved by proposed algorithm.

Every kid $F_i^{K_j}$ ($i = 1, \ldots, M$ and $j = 1, \ldots, O$) randomly generates its set of qualities/variables $\mathbf{x}^{F_i^{K_j}} = x_1^{F_i^{K_j}} \ldots x_n^{F_i^{K_j}}, \ldots, x_N^{F_i^{K_j}}$ in the close neighbourhood of one of its parents and evaluates associated objective function/behaviour $f\left(\mathbf{x}^{F_i^{K_j}}\right)$. This behaviour is illustrated as a schematic in the Fig. 4, where a single family is represented comprising of two parents with their two kids. Each member of the family generates its set of variables (x_1 and x_2) which denotes their qualities. It should be noted that this generation of initial qualities for every individual is simulated using the 'rand' function in MATLAB which generates random numbers whose elements are uniformly distributed in the interval (0, 1).

Step 2 *(Parent follow behaviour/parent influence function)*

Every parent $F_i^{P_1}$ and $F_i^{P_2}$ associated with family F_i ($i = 1, \ldots, M$) decides to follow corresponding behaviour $f^*(x^{F_i^{P[\#]}})$ of one of the parent from certain family F_i, ($i = 1, \ldots, M$) and associated qualities $x^{F_i^{P[\#]}} = x_1^{F_i^{P[\#]}}, \ldots, x_n^{F_i^{P[\#]}}, \ldots, x_N^{F_i^{P[\#]}}$. Thus every parent may follow certain behaviour and the behaviour to follow is selected based on roulette wheel selection approach [68] and the subscript '#' indicates that the behaviour is selected at random by family member and not known in advance. Following certain behaviour implies that the current sampling space/interval of every variable associated with the parent are updated to the close neighbourhood of the individual it follows. A parent may attempt to improve its behaviour by either following the behaviour of another parent through an influence function or through a self-contemplation operator. The self-contemplation operator (described in step4) here refers to searching in the close neighbourhood of its own current sampling space and solution; which may result in self-improvement in its own behaviour. Thus each parent of every family updates the current sampling interval associated with its every variable $\mathbf{x}^{F_i^{P[\#]}} = \left(x_1^{F_i^{P[\#]}} \ldots x_n^{F_i^{P[\#]}}, \ldots, x_N^{F_i^{P[\#]}}\right)$, $n = 1, \ldots, N$ to the close neighbourhood of the best solution $f^*(\mathbf{x}^{F_i^{P[\#]}})$ this particular parent chooses to follow.

Step 3 *(Kid follow behaviour/kid influence function)*

Every kid $F_i^{K_j}$ ($i = 1, \ldots, M$ and $j = 1, \ldots, O$) associated with every Family F_i ($i = 1, \ldots, M$) may choose to follow his siblings or peers or may learn from the behaviours of elders in the societal setup. These elders may be his parents; or in some cases, a kid may be influenced by the qualities of other parents or grownups from some other family. Children will often replicate what they see, and not always what their parents ask them to do. The above behaviour exhibited by kids is modelled using the Kid follow/influence function as follows:

Members from a better behaving family have greater influence on their kids' basic behavioural values. Thus a kid from the global best family may mimic his parents or his siblings which in turn may lead to improvement in his own behaviour. This observation in a social setting has been mapped using the probabilistic parameter r_p, where r_p is the parent_follow_probability and a kid may choose to follow his parents with a probability r_p. Every kid $F_i^{K_j}$ ($i = 1, \ldots, M$ and $j = 1, \ldots, O$) from the global best family generates a random number $rand_{j'} \in [0, 1]$ and probabilistically performs the following:

If $rand_{j'} \leq r_p$ then the kid $F_i^{K_j}$ decides to follow the corresponding behaviour of either of its parents, else it may choose to follow behaviour of one of its own sibling and using a roulette wheel approach decides to follow the corresponding behaviour $f^*(\mathbf{x}^{F_i^{K[\#]}})$ and associated qualities $\mathbf{x}^{F_i^{K[\#]}} = x_1^{F_i^{K[\#]}}, \ldots, x_n^{F_i^{K[\#]}}, \ldots, x_N^{F_i^{K[\#]}}$ of either of his siblings. Thus every kid may follow certain behaviour and the superscript indicates that the behaviour is selected at random by family member and not known in advance, the selection being based on roulette wheel selection approach.

Some other kids may be largely influenced by their peers and may choose to follow other kids with a probability r_k. The parameter, kid_follow_probability r_k is significantly smaller in initial iterations and increases as iterations progress simulating that the peer influence on the children grows as they age. Every other kid $F_i^{K_j}$ ($i = 1, \ldots, M$ and $j = 1, \ldots, O$) from other families randomly chooses to follow behaviour from within one of the other families and generates a random number $\text{rand}_j \in [0, 1]$ and probabilistically performs the following:

If $rand_j \leq r_k$ then kid may choose to imbibe the behaviour of any one of its peers using a roulette wheel approach and decides to follow the corresponding behaviour $f^*(x^{F_i^{K[\#]}})$ and associated qualities $x^{F_i^{K[\#]}} = x_1^{F_i^{K[\#]}}, \ldots, x_n^{F_i^{K[\#]}}, \ldots, x_N^{F_i^{K[\#]}}$ of the chosen peer, else the kid $F_i^{K_j}$ randomly chooses to follow the behaviour of one of the randomly chosen parents or elder from a selected family in the society using a roulette wheel approach.

Step 4 *(Sampling Interval Updation)*

Following or mimicking certain behaviour (in steps 2 and 3) implies that the current sampling interval of every variable associated with every parent or kid is updated to the close neighbourhood of the individual it follows. A parent may decide to follow the behaviour of another grown up from another family based on roulette wheel selection approach (influence function) and calculates the updated objective function (or the associated updated behaviour). Thus every individual of each family updates the sampling interval associated with its every variable (refer to Eqs. (1) and (2))

Every parent $F_i^{P_1}$ and $F_i^{P_2}$ associated with family F_i ($i = 1, \ldots, M$) samples variable values from within the updated sampling interval $\boldsymbol{\varphi}_n^{F_i^{P[\#]}}$, $n = 1, \ldots, N$ associated with every variable $\mathbf{x}_n^{F_i^{P[\#]}}$, $n = 1, \ldots, N$ as follows:

$$\varphi_n^{F_i^{P[\#]}} \in \left[x_n^{F_i^{P[\#]}} - \left(\frac{\left\| \varphi_n^{F_i} \right\|}{2} \right), x_n^{F_i^{P[\#]}} + \left(\frac{\left\| \varphi_n^{F_i} \right\|}{2} \right) \right],$$
$$\text{where } \varphi_n^{F_i} = \left(\left\| \varphi_n^{F_i} \right\| \right) \times r \tag{1}$$

A parent may seek to improve his own behaviour too through self-contemplation i.e. they may not choose to follow another individual each time (as was recommended by the roulette wheel selection operator). This behaviour is simulated through the self-contemplation operator in the current study where if the new solution is worse than the one which parent already had, then the parent/ grownup tries to improve themselves through self-introspection. The self-contemplation operator thus refers to searching in the close neighbourhood of its own current sampling space and solution; which may result in self-improvement in its own behaviour. Self-introspection may guide the search towards a better behaviour. In context of the algorithm, this operator provides a provision to exploit the local neighbourhood of the current solution of an individual, which might be much better than the solutions/behaviour of other parents. This operator simulates the self-help and self-improvement behaviour of an individual exhibited in real life.

Similarly, every kid $F_i^{K_j}$ ($i = 1, \ldots, M$ and $j = 1, \ldots, O$) samples variable values from within the updated sampling interval $\boldsymbol{\varphi}_n^{F_i^{K[\#]}}$ associated with every variable $\mathbf{x}_n^{F_i^{K[\#]}}$, $n = 1, \ldots, N$ to its local neighbourhood and further generate the updated objective function as follows:

$$\varphi_n^{F_i^{K[\#]}} \in \left[x_n^{F_i^{K[\#]}} - \left(\frac{\left\| \varphi_n^{F_i} \right\|}{2} \right), x_n^{F_i^{K[\#]}} + \left(\frac{\left\| \varphi_n^{F_i} \right\|}{2} \right) \right],$$
$$\text{where } \varphi_n^{F_i} = \left(\left\| \varphi_n^{F_i} \right\| \right) \times r \tag{2}$$

Every kid adapts certain behaviour to follow and it is likely that the behaviour it chooses is worse than its current behaviour i.e. in the optimization terminology the newly generated solution may be worse than its current solution. In real life to tackle such situations, parents' intervention is needed who guide the children to correct this unacceptable behaviour (this feature is simulated by behaviour correction operator in the current study). Peer influence on kids becomes greater as they progress through their childhood and parents attempt to correct the deteriorating behaviour of their kids. If the behaviour chosen to be followed by a kid has resulted in its own behaviour to worsen, then the parents intervene. They supervise the adapted behaviour of their kid and if this new behaviour is inferior then parents help their children make better behaviour choices. This process, which is performed by the Behaviour Correction operator in SELO algorithm, is described as:

The kid may sample values from any of the three sampling intervals: $\varphi_n^{F_i^{K_j}}$ which was formed using the kids own current behaviour, $\varphi_n^{F_i^{P_1}}$ formed using current behaviour of parent $F_i^{P_1}$ of the kid or $\varphi_n^{F_i^{P_2}}$ formed using current behaviour of parent $F_i^{P_2}$ of the kid. It then calculates the associated solution or behaviour. Then one of the better behaviours from these three will be selected based on their fitness proportions through the roulette wheel selection. The Behaviour Correction operator as enforced by the parents will try to improve the behaviour of the kid.

Step 4 thus results in every family $F_i (i = 1, \ldots, M)$ being available with, in all, $(2 + O)$ updated behaviours (2 parents and 'O' kids associated with each family $f\left(\mathbf{x}^{F_i^{P_1}}\right), f\left(\mathbf{x}^{F_i^{P_2}}\right), f\left(\mathbf{x}^{F_i^{K_j}}\right), j = 1, \ldots, O$). The best behaviour can be represented as $\boldsymbol{F}_{Min}^{M} = \{min\,(F_i)\}\,(i = 1, \ldots, M)$.

Step 5 *(Exploitation)*

Similar to other evolutionary algorithms, the SELO algorithm continues its iterations while exploiting better solutions until certain stopping criteria are met. The multi-agent algorithm continues to look within the neighbourhood of previously visited solutions till there is no significant improvement in the independent behaviour of every family, and the difference between the individual behaviours is not very significant for successive considerable number of iterations c i.e. if for every family F_i, $(i = 1, \ldots, M)$:

3 $\left\| \max\left(\mathbf{F}^{M}\right)^{c} - \max\left(\mathbf{F}^{M}\right)^{c-1} \right\| \leq \varepsilon$, AND

4 $\left\| \min\left(\mathbf{F}^{M}\right)^{c} - \min\left(\mathbf{F}^{M}\right)^{c-1} \right\| \leq \varepsilon$, AND

5 $\left\| \max\left(\mathbf{F}^{M}\right)^{c} - \min\left(\mathbf{F}^{M}\right)^{c-1} \right\| \leq \varepsilon$

Values of the probabilistic parameters r_p, r_k are also iteratively reduced to some minimum value (Fig. 5). If the families have converged in the current iteration then continue to step 6, else go back to step 2.

Step 6 *(Convergence and further search)*

Once the behaviours of family members are converged (satisfying the conditions in step 5), the values of family_follow_probability $r_p \in [0, 1]$ and peer_follow_probability $r_k \in [0, 1]$ are reset to their initial values so that the parents' help could be sought. The steps 2 through 5 are then repeated again for considerable number of learning attempts l.

The SELO algorithm is considered to be converged when the following convergence conditions are satisfied else continue to step 1

(a) maximum number of learning attempts l_{max} is reached, OR
(b) family members' are converged and families' behaviours saturate (satisfying the conditions in Step 5)

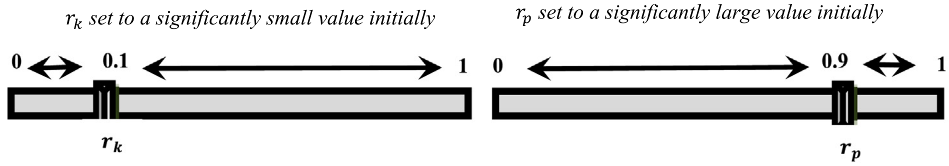

Fig. 5. Representation of peer_follow_probability and family_follow_probability.

Accept any of the M behaviours from current set of behaviours in the families F_i, $(i = 1, \ldots, M)$ as the final objective function $\boldsymbol{F}^M_{Min}$ value as the final solution and stop. The formulation of the SELO methodology is explained below in detail with the algorithmic flowchart in Fig. 6.

3. Results and discussion

This section presents the benchmark test problems used and the results and findings in order to evaluate the performance of the proposed Social algorithm and how well it performs on finding the global optimum solution for the unconstrained problems. The section also discusses the control parameters, precision and the stopping criterion used for testing the optimization algorithm along with tabulated results. The performance of SELO is compared to other widely used population based algorithms like PSO, CMAES, ABC, JDE, CLPSO, SADE, and BSA since these algorithms are popular across scientific and real-world applications. The current algorithm is also compared to Ideology Algorithm (IA) which is a recently proposed socio-inspired metaheuristic with promising results. In the experiment, the computation for each of the algorithms is executed in MATLAB R2013a on a Windows Platform with a T6400, 4 GHz Intel Core-2 Duo processor with 4 GB RAM.

3.1. Benchmark test problems

To compare the relative success of the proposed algorithm with other classic population-based optimization techniques an empirical study based on a set of 50 benchmark functions (f1 − f50) [71,72] is conducted (refer Table 2). These set of test functions include a multitude of problems with varying problem complexity levels such as unimodal, multimodal, separable and non-separable [29,73,74]. Multimodal functions are used to validate the capability of an algorithm to escape from local minimum, which tests the explorations capability of an algorithm. Another measure of difficulty is the separability of test functions. Broadly speaking, separable functions are easier to optimize, as compared non-separable functions, because each variable of a function is independent of the other variables. A good blend of these test functions have been used in the current test bed with diverse properties such as modality and separability, which proves useful in estimating and validating various characteristics of optimization algorithms like convergence rate, precision and the general ability of exploration and exploitation to find optimal solutions while jumping out of local minima. All benchmark test problems are divided into four categories such as US, MS, UN, MN, and its range, formulation, characteristics and the dimensions of these problems are listed in the Table 2.

3.2. Stopping criterion

Each experiment has been repeated 30 times on each function with a random initial seed value each time. Values below $1E^{-16}$ are considered to be zero; considering an arithmetic precision of 10^{-16}. Thus the algorithm is assumed to have converged or terminated based on the following stopping criteria:

Table 3
Initial value of relevant control parameters for SELO.

Control parameter	Initial values
Maximum number of iterations (*Imax*)	70000
Initial number of families created (M)	03
Number of parents in each family (P)	02
Number of children in each family (O)	03
parent_follow_probability r_p	0.999
follow_prob_factor_ownparent	0.999
peer_follow_probability r_k	0.1
follow_prob_factor_otherkids	0.9991
sampling interval reduction factor r	0.95000 to 0.99995

- If the absolute value of the objective function is less than 10^{-16}
- If the algorithm is unable to improve the quality of the found solution over a period of few preceding iterations
- Maximum number of iterations or maximum number of functions evaluations is reached

3.3. Control parameters

Table 3 summarizes the values of various control parameters used for the experimentation of proposed algorithm SELO.

The values of the various control parameters have been chosen based on initial trial runs carried out on the algorithm. The parameters mentioned above have been retained as is for most of the test problems; however for few large dimension or complex problems these parameters have been tuned to other values based on a few initial trial runs to obtain the optimal(near optimal) solutions.

3.4. Analysis of results

In the study, each of the benchmark problems were solved 30 times and the global minimum and the running time for each independent run or trial of the algorithm was recorded. Nature-inspired algorithms always have some randomness owing to their stochastic nature. Thus solutions of the algorithm in the population will be different each time the program is run; each time arriving at better or inferior solutions than they may have arrived at during their search for newer solutions to a certain problem. To take into account this case, it is therefore required that the overall success of a metaheuristic at solving an optimization function be considered based on its performance across a series of trial or runs. The performance and problem solving success of optimization algorithms are compared by using statistical measures. One algorithm may be compared to another with the help of statistical measures to compare their correctness, algorithmic accuracy and computational complexities. The simple statistical values like mean solution (*mean*), standard deviation of the mean solution (*S.D.*), best solution (*Best*) produced and the average running time (*R.T.*) by the algorithms were recorded. The detailed results are presented in Table 4. To validate the performance and potential of the proposed approach, the results are then compared with some of the other popular metaheuristic algorithms like PSO, CMAES,

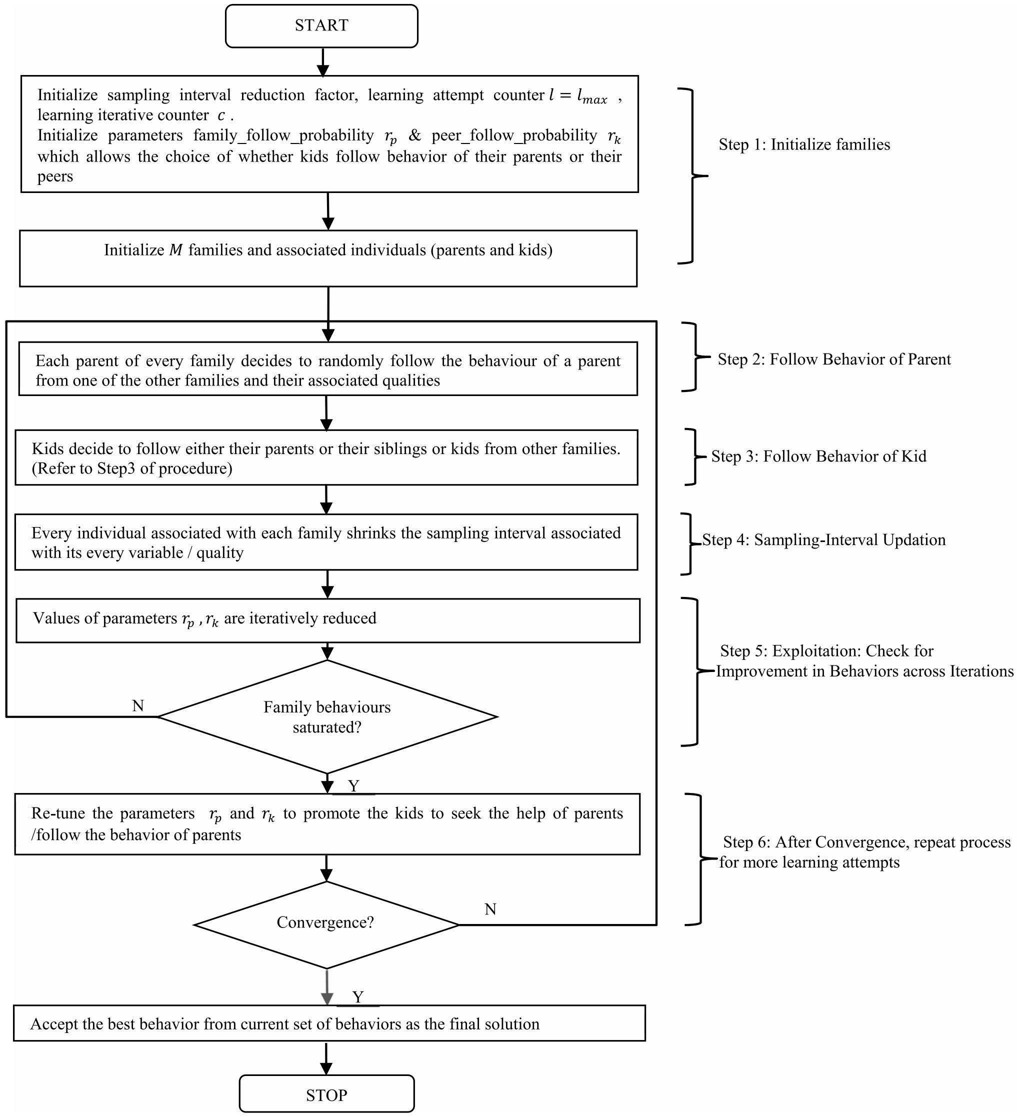

Fig. 6. SELO Algorithm flowchart.

ABC, JDE, CLPSO, SADE BSA and IA. Table 5 shows the performance comparison and ranks of the algorithms in solving the 50 benchmark functions. These functions provide reasonably difficult test environments for the EA's.

In Table 5, the first column (*optimal*) shows the accepted global minimum or the optimal solution to a particular optimization function ($f(x)$). The algorithms have been ranked based on their mean solutions i.e. relative ranking based on the affectivity of the global solution attained by the algorithm for a particular $f(x)$. For each of the functions, the second row indicates the rank of each of comparator algorithms on the specified function. The last row of Table 5 provides the overall ranking point (*in parentheses*) and consequently the final rank of each of the algorithm. The observations from the table signify that the proposed meta-heuristic SELO could reach the global optimum in 35 out of 50(70%) problems during each of the runs. This record for PSO, CMAES, ABC, JDE, CLPSO, SADE, BSA and IA is 22 out of 50(44%), 11 out of 50(22%), 24 out of 50(48%), 20 out of 50(40%), 20 out of 50(40%), 31 out of 50(62%), 36 out of 50(72%) and 21 out of 50(42%) respectively. It can be inferred from the current experiments and analysis that SELO and BSA are most promising in attaining the global optimum/near optimum solutions. SELO and BSA show comparable performances while other algorithms under comparison show relatively mediocre performances in reaching the best minimal solutions to the test bed of simulation optimization problems. The performances of ABC, JDE, CLPSO and IA are rather equal. It is however worth noting that all algorithms under comparison are well-established and have reached close to the acceptable solutions for a number of test problems in the experiments; but have not been able to rank 1 and reach the best solution or the global optimum in a number of the test problems under consideration. Thus SELO and BSA prove to be the most powerful in reaching the optimum maximum number of times, under the current study.

In assessing the overall capability of any optimization algorithm, it is important to also analyse the types of the test problems that the algorithm solves most successfully. If the overall ranking points are to be considered based on general performance then BSA, SADE, ABC and SELO record the most consistent and reliable performances out of the all the 9 algorithms, where BSA shows a superior performance over all others. CLPSO, JDE, IA, PSO and CMAES can be ranked as average performers based on their overall ranking and problem solving ability across all the 50 benchmark problems. Among all algorithms CMAES exhibits the poorest performance recording the last place in ranking on

Table 4
Statistical solutions obtained by PSO, CMAES, ABC, JDE, CLPSO, SADE, BSA, IA and proposed SELO in the test (Mean = Mean solution; S.D. = Standard-deviation of mean solution; Best = Best solution; R.T. = Mean Run Time in seconds).

$f(x)$	Statistic	PSO2011	CMAES	ABC	JDE	CLPSO	SADE	BSA	IA	SELO
F1	Mean	1.3316029264876300	10.0748846367972000	0.9980038377944500	1.0641405484285200	1.8209961275956800	0.9980038377944500	0.9980038377944500	0.9980038690000000	0.9980038538690870
	S.D.	0.9455237994690700	8.0277365400340800	0.0000000000000001	0.3622456829347420	1.6979175079427900	0.0000000000000000	0.0000000000000000	0.0000000000000035	0.0000013769725269
	Best	0.9980038377944500	0.9980038377944500	0.9980038377944500	0.9980038377944500	0.9980038377944500	0.9980038377944500	0.9980038377944500	0.9980038685998520	0.9980038383022720
	R.T.	72.527	44.788	64.976	51.101	61.650	66.633	38.125	43.535	1.750
F2	Mean	2.9999999999999200	21.8999999999995000	3.0000000465423000	2.9999999999999200	3.0000000000000700	2.9999999999999200	2.9999999999999200	3.0240147900000000	3.0013971187248700
	S.D.	0.0000000000000013	32.6088098948516000	0.0000002350442161	0.0000000000000013	0.0000000000007941	0.0000000000000020	0.0000000000000011	0.0787814840000000	0.0018936009191261
	Best	2.9999999999999200	2.9999999999999200	2.9999999999999200	2.9999999999999200	2.9999999999999200	2.9999999999999200	2.9999999999999200	3.0029461118668700	3.0000021202023800
	R.T.	17.892	24.361	16.624	7.224	24.784	28.699	7.692	41.343	28.909
F3	Mean	0.1278728062391630	0.0241892995662904	0.0000000000000004	0.0034556340083499	0.0000000000000000	0.0034556340083499	0.0000000000000000	0.3536752140000000	0.2899597890213580
	S.D.	0.2772792346028400	0.0802240262581864	0.0000000000000001	0.0189272869685522	0.0000000000000000	0.0189272869685522	0.0000000000000000	1.4205454130000000	0.0159272187796787
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000003	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0014898619035614	0.2497905224307240
	R.T.	139.555	5.851	84.416	9.492	38.484	15.992	18.922	34.494	59.260
F4	Mean	0.0043949463343535	0.0003662455278628	0.0000000000000004	0.0007324910557256	0.0000000000000000	0.0440448539086004	0.0000000000000000	0.0179485820000000	2.3720510573781100
	S.D.	0.0054747064090174	0.0020600937719584	0.0000000000000001	0.0027875840585535	0.0000000000000000	0.2227372747439610	0.0000000000000000	0.0526650620000000	0.1531241868389090
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000003	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000165491	2.0664368584658500
	R.T.	126.507	6.158	113.937	14.367	48.667	33.019	24.309	322.808	37.105
F5	Mean	1.5214322973725000	11.7040011684582000	0.0000000000000340	0.0811017056422860	0.1863456353861950	0.7915368220335460	0.0000000000000105	0.0000000000000009	0.0000000000000000
	S.D.	0.6617570384662600	9.7201961540865200	0.0000000000000035	0.3176012689149320	0.4389839299322230	0.7561593402959740	0.0000000000000034	0.0000000000000000	0.0000000000000002
	Best	0.0000000000000080	0.0000000000000080	0.0000000000000293	0.0000000000000044	0.0000000000000080	0.0000000000000044	0.0000000000000080	0.0000000000000009	0.0000000000000000
	R.T.	63.039	3.144	23.293	11.016	45.734	40.914	14.396	49.458	1.120
F6	Mean	0.0000000041922968	0.2540232169641050	0.0000000000000028	0.0000000000000000	0.0000444354499943	0.0000000000000000	0.0000000000000000	0.0082236060000000	0.0000997928359263
	S.D.	0.0000000139615552	0.3653844307786430	0.0000000000000030	0.0000000000000000	0.0001015919507724	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0001311815541321
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000005	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0082236059357692	0.0000007530509495
	R.T.	32.409	4.455	22.367	1.279	125.839	4.544	0.962	50.246	23.876
F7	Mean	0.0000000000000000	0.0622354533647150	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	S.D.	0.0000000000000000	0.1345061339146580	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	R.T.	16.956	6.845	1.832	1.141	2.926	4.409	0.825	38.506	0.723
F8	Mean	0.0000000000000000	0.0072771062590204	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	S.D.	0.0000000000000000	0.0398583525142753	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	R.T.	17.039	2.174	1.804	1.139	2.891	4.417	0.824	39.023	1.442
F9	Mean	0.0000000000000000	0.0001048363065820	0.0000000000000006	0.0000000000000000	0.0001933464326398	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	S.D.	0.0000000000000000	0.0005742120996051	0.0000000000000003	0.0000000000000000	0.0000846531630676	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000001
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000001	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	R.T.	17.136	2.127	21.713	1.129	33.307	4.303	0.829	40.896	3.028
F10	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0006005122443674	0.0000000000000000	0.0000000000000000	0.8346587090000000	0.0000000000000000
	S.D.	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0029861918862801	0.0000000000000000	0.0000000000000000	0.0000000000000005	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.8346587086917530	0.0000000000000000
	R.T.	17.072	1.375	22.395	1.099	28.508	4.371	0.790	39.978	7.373
F11	Mean	0.3978873577297380	0.6372170283279430	0.3978873577297380	0.3978873577297380	0.3978873577297390	0.3978873577297380	0.3978873577297380	0.4156431270000000	0.3978943993817670
	S.D.	0.0000000000000000	0.7302632173480510	0.0000000000000000	0.0000000000000000	0.0000000000000049	0.0000000000000000	0.0000000000000000	0.0406451050000000	0.0003536060523484
	Best	0.3978873577297380	0.3978873577297380	0.3978873577297380	0.3978873577297380	0.3978873577297380	0.3978873577297380	0.3978873577297380	0.4012748152492080	0.3978822494361650
	R.T.	17.049	24.643	10.941	6.814	17.283	27.981	5.450	40.099	22.72
F12	Mean	0.0000000000000000	0.0000000000000000	0.0715675060725970	0.0000000000000000	0.1593872502094070	0.0000000000000000	0.0000000000000000	0.0014898620000000	3.6688019971758100
	S.D.	0.0000000000000000	0.0000000000000000	0.0579425013417103	0.0000000000000000	0.6678482786713720	0.0000000000000000	0.0000000000000000	0.0000000000000000	1.7577708967227600
	Best	0.0000000000000000	0.0000000000000000	0.0013425253994745	0.0000000000000000	0.0000094069599934	0.0000000000000000	0.0000000000000000	0.0082029783984983	0.8388908577815620
	R.T.	44.065	1.548	21.487	1.251	166.965	4.405	2.460	48.067	47.028
F13	Mean	0.6666666666666750	0.6666666666666670	0.0000000000000038	0.6666666666666670	0.0023282133668190	0.6666666666666670	0.6444444444444440	0.2528116640000000	0.9737369841168760
	S.D.	0.0000000000000022	0.0000000000000000	0.0000000000000012	0.0000000000000002	0.0051792840882291	0.0000000000000000	0.1217161238900370	0.0000000006509080	0.0054869670667257
	Best	0.6666666666666720	0.6666666666666670	0.0000000000000021	0.6666666666666670	0.0000120708732167	0.6666666666666670	0.0000000000000000	0.2528116633611470	0.9541730938494050
	R.T.	167.094	3.719	37.604	18.689	216.261	47.833	21.192	67.463	38.897
F14	Mean	−1.0000000000000000	−0.1000000000000000	−1.0000000000000000	−1.0000000000000000	−1.0000000000000000	−1.0000000000000000	−1.0000000000000000	−0.9997989620000000	0.0000000000000000
	S.D.	0.0000000000000000	0.3051285766293650	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000167151	0.1083854312160620
	Best	−1.0000000000000000	−1.0000000000000000	−1.0000000000000000	−1.0000000000000000	−1.0000000000000000	−1.0000000000000000	−1.0000000000000000	−0.9997989624626810	−0.5936514558196160
	R.T.	16.633	3.606	13.629	6.918	16.910	28.739	5.451	39.685	0.0129476459522760
F15	Mean	0.0000000000000000	1028.3930784026900000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	S.D.	0.0000000000000000	1298.1521820113500000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	R.T.	27.859	15.541	40.030	2.852	4.030	6.020	2.067	38.867	0.004
F16	Mean	48.7465164446927000	1680.3460230073400000	0.0218688498331872	0.9443728655432830	81.7751618148164000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	S.D.	88.8658510972991000	2447.7484859066000000	0.0418409568792831	2.8815514827061600	379.9241117377270000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000016	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	R.T.	95.352	11.947	44.572	4.719	162.941	5.763	7.781	48.262	1.350

(*continued on next page*)

Table 4 (*continued*)

$f(x)$	Statistic	PSO2011	CMAES	ABC	JDE	CLPSO	SADE	BSA	IA	SELO
F17	Mean	918.9518492782850000	12340.2283326398000000	11.0681496253548000	713.7226974626920000	0.8530843976878610	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	S.D.	1652.4810858411400000	22367.1698875802000000	9.8810950146557100	1710.0713074301200000	2.9208253191698800	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.3274654777056860	0.0000000000000000	0.0016957837829822	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	R.T.	271.222	7.631	43.329	16.105	268.894	168.310	33.044	69.060	0.808
F18	Mean	0.0068943694819713	0.0011498935321349	0.0000000000000000	0.0048193578543185	0.0000000000000000	0.0226359326967139	0.0004930693556077	0.0000000000000000	0.0000000000000000
	S.D.	0.0080565201649587	0.0036449413521107	0.0000000000000001	0.0133238235582874	0.0000000000000000	0.0283874287215679	0.0018764355751644	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	R.T.	73.895	2.647	19.073	6.914	14.864	25.858	5.753	2.717	0.792
F19	Mean	−3.8627821478207500	−3.7243887744664700	−3.8627821478207500	−3.8627821478207500	−3.8627821478207500	−3.8627821478207500	−3.8627821478207500	−3.8596352620000000	−2.2922815000937700
	S.D.	0.0000000000000027	0.5407823545193820	0.0000000000000024	0.0000000000000027	0.0000000000000027	0.0000000000000027	0.0000000000000027	0.0033967610000000	0.5795350381767260
	Best	−3.8627821478207600	−3.8627821478207600	−3.8627821478207600	−3.8627821478207600	−3.8627821478207600	−3.8627821478207600	−3.8627821478207600	−3.8613076574052300	−3.5841184056629400
	R.T.	19.280	21.881	12.613	7.509	17.504	24.804	6.009	46.167	32.987
F20	Mean	−3.3180320675402500	−3.2942534432762600	−3.3219951715842400	−3.2982165473202600	−3.3219951715842400	−3.3140689634962500	−3.3219951715842400	−2.5710247593206100	−1.1719158908829300
	S.D.	0.0217068148263721	0.0511458075926848	0.0000000000000014	0.0483702518391572	0.0000000000000013	0.0301641516823498	0.0000000000000013	0.0000000000000009	0.0003690446342091
	Best	−3.3219951715842400	−3.3219951715842400	−3.3219951715842400	−3.3219951715842400	−3.3219951715842400	−3.3219951715842400	−3.3219951715842400	−2.5710247593206100	−1.1727699585993300
	R.T.	26.209	7.333	13.562	8.008	20.099	33.719	6.822	59.083	14.864
F21	Mean	0.0003074859878056	0.0064830287538208	0.0004414866359626	0.0003685318137604	0.0003100479704151	0.0003074859878056	0.0003074859878056	0.0016993410000000	0.0003493601571991
	S.D.	0.0000000000000000	0.0148565973286009	0.0000568392289725	0.0002323173367683	0.0000059843325073	0.0000000000000000	0.0000000000000000	0.0000013058400000	0.0000226057336871
	Best	0.0003074859878056	0.0003074859878056	0.0003230956007045	0.0003074859878056	0.0003074859941292	0.0003074859878056	0.0003074859878056	0.0016989914552560	0.0003226283751593
	R.T.	84.471	13.864	20.255	7.806	156.095	45.443	11.722	48.920	15.969
F22	Mean	−1.0809384421344400	−0.7323679641701760	−1.0809384421344400	−1.0764280762657400	−1.0202940450426400	−1.0809384421344400	−1.0809384421344400	−1.4315374190000000	−1.0835400071766800
	S.D.	0.0000000000000006	0.4136688304155380	0.0000000000000008	0.0247042912888477	0.1190811583120530	0.0000000000000005	0.0000000000000005	0.0000000000000009	0.5277882902242550
	Best	−1.0809384421344400	−1.0809384421344400	−1.0809384421344400	−1.0809384421344400	−1.0809384421344400	−1.0809384421344400	−1.0809384421344400	−1.4315374193830000	−2.1933014645645000
	R.T.	27.372	32.311	27.546	19.673	52.853	36.659	21.421	34.714	1.757
F23	Mean	−1.3891992200744600	−0.5235864386288060	−1.4999990070800800	−1.3431399432579700	−1.4765972735526500	−1.4999992233525000	−1.4821658762555300	−1.5000000000000000	−1.4999998390866700
	S.D.	0.2257194403158630	0.2585330714077300	0.0000008440502079	0.2680292304904580	0.1281777579497830	0.0000000000000009	0.0976772648082733	0.0000000000000000	0.0000000818646069
	Best	−1.4999992233524900	−0.7977041047646610	−1.4999992233524900	−1.4999992233524900	−1.4999992233524900	−1.4999992233524900	−1.4999992233524900	−1.5000000000000000	−1.4999999590992100
	R.T.	33.809	17.940	37.986	20.333	42.488	36.037	18.930	41.848	4.708
F24	Mean	−0.9166206788680230	−0.3105071678265780	−0.8406348096500680	−0.882715279883576	−0.9431432797743700	−1.2765515661973800	−1.3127183561646500	−1.5000000000000000	−1.4999991427332700
	S.D.	0.3917752367440500	0.2080317241440800	0.2000966365984320	0.3882445165494030	0.3184175870987750	0.3599594108130040	0.3158807699946290	0.0000000000000000	0.0000003717669841
	Best	−1.5000000000003800	−0.7976938356122860	−1.4999926800631400	−1.5000000000003800	−1.5000000000003800	−1.5000000000003800	−1.5000000000003800	−1.5000000000000000	−1.4999999303979900
	R.T.	110.798	8.835	38.470	21.599	124.609	47.171	35.358	54.651	17.794
F25	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000004	0.0000000000000000	0.0000041787372626	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	S.D.	0.0000000000000000	0.0000000000000000	0.0000000000000003	0.0000000000000000	0.0000161643637543	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000001	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	R.T.	25.358	1.340	19.689	1.142	31.632	4.090	0.813	35.662	1.8469296606031600
F26	Mean	−1.8210436836776800	−1.7829268228561700	−1.8210436836776800	−1.8210436836776800	−1.8210436836776800	−1.8210436836776800	−1.8210436836776800	−1.8203821100000000	−1.8166465888521900
	S.D.	0.0000000000000009	0.1450583631808370	0.0000000000000009	0.0000000000000009	0.0000000000000009	0.0000000000000009	0.0000000000000009	0.0000000000000014	0.0072804985619476
	Best	−1.8210436836776800	−1.8210436836776800	−1.8210436836776800	−1.8210436836776800	−1.8210436836776800	−1.8210436836776800	−1.8210436836776800	−1.8203821095139300	−1.8106292157333700
	R.T.	19.154	26.249	17.228	9.663	18.091	28.453	7.472	34.891	1.199
F27	Mean	−4.6565646397053900	−4.1008953007033700	−4.6934684519571100	−4.6893456932617100	−4.6920941990586400	−4.6884965299983800	−4.6934684519571100	−3.2820108350000000	−3.3591408962129900
	S.D.	0.0557021530063238	0.4951250481844850	0.0000000000000009	0.0125797149251589	0.0075270931220834	0.0272323381095561	0.0000000000000008	0.0000000000000023	0.2009584117455920
	Best	−4.6934684519571100	−4.6934684519571100	−4.6934684519571100	−4.6934684519571100	−4.6934684519571100	−4.6934684519571100	−4.6934684519571100	−3.2820108345268900	−3.9631157953194900
	R.T.	38.651	10.956	17.663	14.915	25.843	38.446	11.971	45.085	23.801
F28	Mean	−8.9717330307549300	−7.6193507368464700	−9.6601517156413500	−9.6397230986132500	−9.6400278592589600	−9.6572038232921700	−9.6601517156413500	−6.2086254390000000	−3.9793838974626000
	S.D.	0.4927013165009220	0.7904830398850970	0.0000000000000008	0.0393668145094111	0.0437935551332868	0.0105890022905617	0.0000000000000007	0.0000000000000027	0.0005104314209355
	Best	−9.5777818097208200	−9.1383975057875100	−9.6601517156413500	−9.6601517156413500	−9.6601517156413500	−9.6601517156413500	−9.6601517156413500	−6.2086254392105500	−3.9806353395021300
	R.T.	144.093	6.959	27.051	20.803	32.801	46.395	22.250	71.652	26.425
F29	Mean	0.0119687224560441	0.0788734736114700	0.0838440014038032	0.0154105130055856	0.0198686590210374	0.0140272066690658	0.0007283694780796	1.3116221610000000	2.0169277899221400
	S.D.	0.0385628598040034	0.1426911799629180	0.0778327303965192	0.0308963906374663	0.0613698943155661	0.0328868042987376	0.0014793717464195	0.5590904820000000	1.2374893392409200
	Best	0.0000044608370213	0.0000000000000000	0.0129834451730589	0.0000000000000000	0.0000175219764526	0.0000000000000000	0.0000000000000000	1.0960146962658900	0.3208703882956150
	R.T.	359.039	17.056	60.216	35.044	316.817	92.412	191.881	34.697	14.519
F30	Mean	0.0000130718912008	0.0000000000000000	0.0002604330013462	0.0000000000000001	0.0458769685199585	0.0000002733806735	0.0000000028443186	0.0000000000000000	0.0000000000000000
	S.D.	0.0000014288348929	0.0000000000000000	0.0000394921919294	0.0000000000000002	0.0620254411839524	0.0000001788830279	0.0000000033308990	0.0000000000000000	0.0000000000000000
	Best	0.0000095067504097	0.0000000000000000	0.0001682411286088	0.0000000000000000	0.0005277712020642	0.0000000944121661	0.0000000004769768	0.0000000000000000	0.0000000000000000
	R.T.	567.704	14.535	215.722	194.117	252.779	360.380	144.784	153.221	1.368
F31	Mean	0.0001254882834238	0.0000000000000000	0.0077905311094958	0.0020185116261490	0.0002674563703837	0.0000000000000000	0.0000000111676630	0.0071082040000000	0.0000000000000000
	S.D.	0.0001503556280087	0.0000000000000000	0.0062425841086448	0.0077448684015362	0.0003044909265796	0.0000000000000000	0.0000000184322163	0.0000000000000000	0.0000000000000000
	Best	0.0000000156460198	0.0000000000000000	0.0003958766023752	0.0000000000000000	0.0000023064754605	0.0000000000000000	0.0000000000000000	0.0071082039505830	0.0000000000000000
	R.T.	250.248	12.062	34.665	48.692	227.817	220.886	149.882	43.098	22.492
F32	Mean	0.0003548345513179	0.0701619169853449	0.0250163252527030	0.0013010316180679	0.0019635752485802	0.0016730768406953	0.0019955316015528	0.0002254250000000	0.0000989055208389
	S.D.	0.0001410817500914	0.0288760292572957	0.0077209314806873	0.0009952078711752	0.0043423828633839	0.0007330246909835	0.0009698942217908	0.0005270410000000	0.0000521772789680
	Best	0.0001014332605364	0.0299180701536354	0.0094647580732654	0.0001787238105452	0.0004206447422138	0.0005630852254632	0.0006084880639553	0.0000023800831017	0.0000104209894311
	R.T.	290.669	2.154	34.982	82.124	103.283	171.637	48.237	218.722	50.737

(*continued on next page*)

Table 4 (*continued*)

$f(x)$	Statistic	PSO2011	CMAES	ABC	JDE	CLPSO	SADE	BSA	IA	SELO
F33	Mean	25.6367602258676000	95.9799861204982000	0.0000000000000000	1.1276202647057400	0.6301407361590880	0.8622978494808570	0.0000000000000000	0.0000000000000000	0.0000000000000000
	S.D.	8.2943512684216700	56.6919245985100000	0.0000000000000000	1.0688393637536800	0.8046401822326410	0.9323785263847000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	12.9344677422129000	29.8487565993415000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	R.T.	76.083	2.740	4.090	7.635	18.429	23.594	5.401	2.266	5.941
F34	Mean	2.6757043114269700	0.3986623855035210	0.2856833465904130	1.0630996944802500	5.7631786582751800	1.2137377447007000	0.3986623854300930	0.0000154715000000	0.0000000000000000
	S.D.	12.3490058210004000	1.2164328621946200	0.6247370987465170	1.7930895051734300	13.9484817304201000	1.8518519388285700	1.2164328622195200	0.0000022373400000	0.0000000000000000
	Best	0.0042535368984501	0.0000000000000000	0.0004266049929880	0.0000000000000000	0.0268003205820685	0.0001448955835246	0.0000000000000000	0.0000118803557196	0.0000000000000000
	R.T.	559.966	9.462	35.865	23.278	187.894	268.449	34.681	7.250	5.855
F35	Mean	0.0000000000000000	0.4651202457398910	0.0000000000000000	0.0038863639514140	0.0019431819755029	0.0006477273251676	0.0000000000000000	0.0000000000000000	0.0000000000000000
	S.D.	0.0000000000000000	0.0933685176073728	0.0000000000000000	0.0048411743884718	0.0039528023354469	0.0024650053428137	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0097159098775144	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	R.T.	18.163	24.021	7.861	4.216	8.304	5.902	1.779	33.155	1.835
F36	Mean	−7684.6104757783800000	−6835.1836730901400000	−12569.4866181730000	−12304.9743375341000	−12210.8815698372000	−12549.746895737300000	−12569.486618173000000	−12569.3622100000000000	−0.3402784042291390
	S.D.	745.3954005014180000	750.7338055436110000	0.0000000000022659	221.4322514436480000	205.9313376284770000	44.8939348779747000	0.0000000000024122	0.000000273871000	3.2212919091274600
	Best	−8912.8855854978200000	−8340.0386911070600000	−12569.4866181730000	−12569.4866181730000	−12569.4866181730000	−12569.486618173000000	−12569.486618173000000	−12569.3622054081000000	−0.0325083488969540
	R.T.	307.427	3.174	19.225	10.315	31.499	34.383	11.069	2.306	15.084
F37	Mean	0.0000000000000000	0.0000000000000000	14.5668734126948000	0.0000000000000000	6.4655746330439100	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000009
	S.D.	0.0000000000000000	0.0000000000000000	8.7128443012950300	0.0000000000000000	8.2188901353055800	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000001
	Best	0.0000000000000000	0.0000000000000000	4.0427699323673400	0.0000000000000000	0.1816624029553790	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000007
	R.T.	543.180	3.370	111.841	19.307	179.083	109.551	57.294	100.947	9.060
F38	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000005	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.000000000000000
	S.D.	0.0000000000000000	0.0000000000000000	0.0000000000000001	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000003	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.000000000000000
	R.T.	163.188	2.558	20.588	1.494	12.563	5.627	3.208	47.009	3.520
F39	Mean	−10.1061873621653000	−5.2607563471326400	−10.5364098166920000	−10.3130437162426000	−10.3130437162026000	−10.5364098166921000	−10.5364098166921000	−10.5063235800000000	−10.536281667618100
	S.D.	1.6679113661236400	3.6145751818694000	0.0000000000000023	1.2234265179812200	1.2234265179736500	0.0000000000000016	0.0000000000000018	0.000000025211900	0.0000481237097736
	Best	−10.5364098166921000	−10.5364098166921000	−10.5364098166920000	−10.5364098166921000	−10.5364098166920000	−10.5364098166921000	−10.5364098166920000	−10.5063235792920000	−10.5363928369535000
	R.T.	31.018	11.024	16.015	8.345	37.275	28.031	7.045	55.666	12.547
F40	Mean	−9.5373938082045500	−5.7308569926624600	−10.1531996790582000	−9.5656135761215700	−10.1531996790582000	−9.9847854277673500	−10.1531996790582000	−10.1529842600000000	−10.1531669871808000
	S.D.	1.9062127067994200	3.5141202468383400	0.0000000000000055	1.8315977756329900	0.0000000000000076	0.9224428443735560	0.0000000000000072	0.0000000000542921	0.0000172333322304
	Best	−10.1531996790582000	−10.1531996790582000	−10.1531996790582000	−10.153199679058200	−10.1531996790582000	−10.1531996790582000	−10.1531996790582000	−10.1529842649756000	−10.1531973132210000
	R.T.	25.237	11.177	11.958	7.947	30.885	25.569	6.864	51.507	10.116
F41	Mean	−10.4029405668187000	−6.8674070870953700	−10.4029405668187000	−9.1615813354737300	−10.4029405668187000	−10.4029405668187000	−10.4029405668187000	−10.3988303400000000	−10.4028748144797000
	S.D.	0.0000000000000018	3.6437803702691000	0.0000000000000006	2.8277336448396200	0.0000000000000010	0.0000000000000018	0.0000000000000017	0.0000000001978980	0.0000478046191696
	Best	−10.4029405668187000	−10.4029405668187000	−10.4029405668187000	−10.4029405668187000	−10.4029405668187000	−10.4029405668187000	−10.4029405668187000	−10.3988303385534000	−10.4029869270437000
	R.T.	21.237	11.482	14.911	8.547	31.207	27.064	8.208	53.190	12.219
F42	Mean	−186.7309073569880000	−81.5609772893002000	−186.730908831024000	−186.730908831024000	−186.730908831024000	−186.7309088310240000	−186.7309088310240000	−186.2926481000000000	−186.7153981691330000
	S.D.	0.0000046401472660	66.4508342743478000	0.0000000000000236	0.0000000000000388	0.0000000000000279	0.0000000000000377	0.0000000000000224	0.0000000000000578	0.0190762312882078
	Best	−186.7309088310240000	−186.7309088310240000	−186.730908831024000	−186.730908831024000	−186.730908831024000	−186.7309088310240000	−186.7309088310240000	−186.2926480689880000	−186.7363874875390000
	R.T.	19.770	25.225	13.342	8.213	20.344	27.109	9.002	31.766	23.870
F43	Mean	−1.0316284534898800	−1.0044229658530100	−1.0316284534898800	−1.0316284534898800	−1.0316284534898800	−1.0316284534898800	−1.0316284534898800	−1.0304357800000000	−1.0303924506027700
	S.D.	0.0000000000000005	0.1490105926664260	0.0000000000000005	0.0000000000000005	0.0000000000000005	0.0000000000000005	0.0000000000000005	0.0014911900000000	0.0025133845110030
	Best	−1.0316284534898800	−1.0316284534898800	−1.0316284534898800	−1.0316284534898800	−1.0316284534898800	−1.0316284534898800	−1.0316284534898800	−1.0314500753985900	−1.0314918740874000
	R.T.	16.754	24.798	11.309	7.147	18.564	27.650	5.691	39.897	2.245
F44	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000004	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.000000000000000
	S.D.	0.0000000000000000	0.0000000000000000	0.0000000000000001	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000003	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.000000000000000
	R.T.	159.904	2.321	21.924	1.424	14.389	5.920	3.302	174.577	1.200
F45	Mean	2.3000000000000000	0.0666666666666667	0.0000000000000000	0.9000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000538870000000	0.0000000000000010
	S.D.	1.8597367258983700	0.2537081317024630	0.0000000000000000	3.0211895350832500	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000005399890	0.0000000000000001
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000538860819891	0.0000000000000006
	R.T.	57.276	1.477	1.782	2.919	3.042	4.307	0.883	2.215	5.274
F46	Mean	0.1333333333333330	0.2666666666666670	0.0000000000000000	0.0000000000000000	0.2000000000000000	0.0000000000000000	0.0000000000000000	−0.0153463301609662	0.0000000000000000
	S.D.	0.3457459036417600	0.9444331755018490	0.0000000000000000	0.0000000000000000	0.4068381021724860	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	−0.0153463301609662	0.0000000000000000
	R.T.	20.381	2.442	1.700	1.074	6.142	4.319	0.764	31.068	0.230
F47	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000005	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	S.D.	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000003	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	R.T.	564.178	2.565	24.172	1.870	15.948	6.383	4.309	31.296	1.009
F48	Mean	−50.0000000000002000	−50.0000000000002000	−49.9999999999997000	−50.0000000000002000	−49.4789234062579000	−50.0000000000002000	−50.0000000000002000	−44.7416748700000000	−46.672022811734100
	S.D.	0.0000000000000361	0.0000000000000268	0.0000000000001408	0.0000000000000354	1.3150773145311700	0.0000000000000268	0.0000000000000361	0.0000000000000217	1.1721159591339900
	Best	−50.0000000000002000	−50.0000000000002000	−50.0000000000001000	−50.0000000000002000	−49.9999994167392000	−50.0000000000002000	−50.0000000000002000	−44.7416748706606000	−48.933141750726300
	R.T.	24.627	8.337	22.480	8.623	142.106	36.804	7.747	52.486	30.015

(continued on next page)

Table 4 (*continued*)

$f(x)$	Statistic	PSO2011	CMAES	ABC	JDE	CLPSO	SADE	BSA	IA	SELO
F49	Mean	−210.000000000001000	−210.000000000003000	−209.999999999947000	−210.000000000003000	−199.592588547503000	−210.000000000003000	−210.000000000003000	−150.5540859185450000	−162.571266865506000
	S.D.	0.0000000000009434	0.0000000000003702	0.0000000000138503	0.0000000000008251	9.6415263953591700	0.0000000000004625	0.0000000000003950	0.0000000000000000	0.2649613601835890
	Best	−210.000000000003000	−210.000000000003000	−209.999999999969000	−210.000000000004000	−209.985867409029000	−210.000000000004000	−210.000000000004000	−150.5540859185450000	−162.922114827822000
	R.T.	48.580	5.988	36.639	11.319	187.787	54.421	11.158	70.887	39.873
F50	Mean	0.0000000000000000	0.0000000000000000	0.0000000402380424	0.0000000000000000	0.0000000001597805	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	S.D.	0.0000000000000000	0.0000000000000000	0.0000002203520334	0.0000000000000000	0.0000000006266641	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	Best	0.0000000000000000	0.0000000000000000	0.0000000000000210	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
	R.T.	86.369	1.868	86.449	1.412	157.838	4.930	5.702	33.573	3.261

Table 5
Performance comparison and rankings of algorithms PSO, CMAES, ABC, JDE, CLPSO, SADE, BSA, IA and proposed SELO on 50 benchmark functions.

F(*x*)	Optimal	Statistics	PSO2011	CMAES	ABC	JDE	CLPSO	SADE	BSA	IA	SELO
F1	0.998003	Mean	1.3316029264876300 (+)	10.0748846367972000 (+)	0.9980038377944500	1.0641405484285200 (+)	1.8209961275956800 (+)	0.9980038377944500	0.9980038377944500	0.9980038690000000	0.9980038538690870
		Rank	3	5	1	2	4	1	1	1	1
F2	3	Mean	2.9999999999999200	21.8999999999995000 (+)	3.0000000465423000	2.9999999999999200	3.0000000000000700	2.9999999999999200	2.9999999999999200	3.0240147900000000 (+)	3.0013971187248700
		Rank	1	3	1	1	1	1	1	2	1
F3	0	Mean	0.1278728062391630	0.0241892995662904	0.0000000000000004	0.0034556340083499	0.0000000000000000	0.0034556340083499	0.0000000000000000	0.3536752140000000	0.2899597890213580
		Rank	5	4	2	3	1	3	1	7	6
F4	0	Mean	0.0043949463343535	0.0003662455278628	0.0000000000000004	0.0007324910557256	0.0000000000000000	0.0440448539086004	0.0000000000000000	0.0179485820000000	2.3720510573781100
		Rank	5	3	2	4	1	7	1	6	8
F5	0	Mean	1.5214322973725000 (+)	11.7040011684582000 (+)	0.0000000000000340 (+)	0.0811017056422860 (+)	0.1863456353861950 (+)	0.7915368220335460 (+)	0.0000000000000105 (+)	0.0000000000000009 (+)	0.0000000000000000
		Rank	8	9	4	5	6	7	3	2	1
F6	0	Mean	0.0000000041922968	0.2540232169641050	0.0000000000000028	0.0000000000000000	0.0000444354499943	0.0000000000000000	0.0000000000000000	0.0082236060000000	0.0000997928359263
		Rank	3	7	2	1	4	1	1	6	5
F7	0	Mean	0.0000000000000000	0.0622354533647150 (+)	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
		Rank	1	2	1	1	1	1	1	1	1
F8	0	Mean	0.0000000000000000	0.0072771062590204 (+)	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
		Rank	1	2	1	1	1	1	1	1	1
F9	0	Mean	0.0000000000000000	0.0001048363065820 (+)	0.0000000000000006 (+)	0.0000000000000000	0.0000193464326398 (+)	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
		Rank	1	4	2	1	3	1	1	1	1
F10	0	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0006005122443674 (+)	0.0000000000000000	0.0000000000000000	0.8346587090000000 (+)	0.0000000000000000
		Rank	1	1	1	1	2	1	1	3	1
F11	0.39788	Mean	0.3978873577297380	0.6372170283279430 (+)	0.3978873577297380	0.3978873577297380	0.3978873577297390	0.3978873577297380	0.3978873577297380	0.4156431270000000 (+)	0.3978943993817670
		Rank	1	3	1	1	1	1	1	2	1
F12	0	Mean	0.0000000000000000	0.0000000000000000	0.0715675060725970	0.0000000000000000	0.1593872502094070	0.0000000000000000	0.0000000000000000	0.0014898620000000	3.6688019971758100
		Rank	1	1	3	1	4	1	1	2	5
F13	0	Mean	0.6666666666666750	0.6666666666666670	0.0000000000000038	0.6666666666666670	0.0023282133668190	0.6666666666666670	0.6444444444444440	0.2528116640000000	0.9737369841168760
		Rank	6	5	1	5	2	5	4	3	7
F14	−1	Mean	−1.0000000000000000	−0.1000000000000000	−1.000000000000000	−1.000000000000000	−1.000000000000000	−1.000000000000000	−1.000000000000000	−0.999798962000000	0.0000000000000000
		Rank	1	3	1	1	1	1	1	2	4
F15	0	Mean	0.0000000000000000	1028.3930784026900000 (+)	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
		Rank	1	2	1	1	1	1	1	1	1
F16	0	Mean	48.7465164446927000 (+)	1680.3460230073400000 (+)	0.0218688498331872 (+)	0.9443728655432830 (+)	81.7751618148164000 (+)	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
		Rank	4	6	2	3	5	1	1	1	1
F17	0	Mean	918.9518492782850000 (+)	12340.2283326398000000 (+)	11.0681496253548000 (+)	713.7226974626920000 (+)	0.8530843976878610 (+)	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
		Rank	5	6	3	4	2	1	1	1	1
F18	0	Mean	0.0068943694819713 (+)	0.0011498935321349 (+)	0.0000000000000000	0.0048193578543185 (+)	0.0000000000000000	0.0226359326967139 (+)	0.0004930693556077 (+)	0.0000000000000000	0.0000000000000000
		Rank	5	3	1	4	1	6	2	1	1
F19	−3.86278	Mean	−3.8627821478207500	−3.7243887744664700	−3.8627821478207500	−3.8627821478207500	−3.8627821478207500	−3.8627821478207500	−3.8627821478207500	−3.8596352620000000	−2.2922815000937700
		Rank	1	3	1	1	1	1	1	2	4
F20	−3.32237	Mean	−3.3180320675402500	−3.2942534432762600	−3.3219951715842400	−3.2982165473202600	−3.3219951715842400	−3.3140689634962500	−3.3219951715842400	−2.5710247593206100	−1.1719158908829300
		Rank	2	5	1	4	1	3	1	6	7
F21	0.0003	Mean	0.0003074859878056	0.0064830287538208 (+)	0.0004414866359626 (+)	0.0003685318137604 (+)	0.0003100479704151	0.0003074859878056	0.0003074859878056	0.0016993410000000 (+)	0.0003493601571991
		Rank	1	4	2	2	1	1	1	3	1
F22	−1.0809	Mean	−1.0809384421344400	−0.7323679641701760 (+)	−1.0809384421344400	−1.0764280762657400 (+)	−1.0202940450426400 (+)	−1.0809384421344400	−1.0809384421344400	−1.4315374190000000 (+)	−1.0835400071766800
		Rank	1	5	1	3	4	1	1	6	1

(*continued on next page*)

Table 5 (*continued*)

$F(x)$	Optimal	Statistics	PSO2011	CMAES	ABC	JDE	CLPSO	SADE	BSA	IA	SELO
F23	−1.5	Mean	−1.3891992200744600 (+)	−0.5235864386288060 (+)	−1.4999990070800800	−1.3431399432579700 (+)	−1.4765972735526500 (+)	−1.4999992233525000	−1.4821658762555300 (+)	−1.5000000000000000	−1.4999998390866700
		Rank	4	6	1	5	3	1	2	1	1
F24	−1.5	Mean	−0.9166206788680230 (+)	−0.3105071678265780 (+)	−0.8406348096500680 (+)	−0.8827152798835760 (+)	−0.9431432797743700 (+)	−1.2765515661973800 (+)	−1.3127183561646500 (+)	−1.5000000000000000	−1.4999991427332700
		Rank	5	8	7	6	4	3	2	1	1
F25	0	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000004 (+)	0.0000000000000000	0.0000041787372626 (+)	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
		Rank	1	1	2	1	3	1	1	1	1
F26	−1.8013	Mean	−1.8210436836776800 (+)	−1.7829268228561700 (+)	−1.8210436836776800 (+)	−1.8210436836776800 (+)	−1.8210436836776800 (+)	−1.8210436836776800 (+)	−1.8210436836776800 (+)	−1.8203821100000000 (+)	−1.8166465888521900
		Rank	4	2	4	4	4	4	4	3	1
F27	−4.6876	Mean	−4.6565646397053900	−4.1008953007033700	−4.6934684519571100	−4.6893456932617100	−4.6920941990586400	−4.6884965299983800	−4.6934684519571100	−3.2820108350000000	−3.3591408962129900
		Rank	5	6	4	2	3	1	4	8	7
F28	−9.66015	Mean	−8.9717330307549300	−7.6193507368464700	−9.6601517156413500	−9.6397230986132500	−9.6400278592589600	−9.6572038232921700	−9.6601517156413500	−6.2086254390000000	−3.9793838974626000
		Rank	5	6	1	4	3	2	1	7	8
F29	0	Mean	0.0119687224560441	0.0788734736114700	0.0838440014038032	0.0154105130055856	0.0198686590210374	0.0140272066690658	0.0007283694780796	1.3116221610000000	2.0169277899221400
		Rank	2	6	7	4	5	3	1	8	9
F30	0	Mean	0.0000130718912008 (+)	0.0000000000000000	0.0002604330013462 (+)	0.0000000000000001 (+)	0.0458769685199585 (+)	0.0000002733806735 (+)	0.0000000028443186 (+)	0.0000000000000000	0.0000000000000000
		Rank	5	1	6	2	7	4	3	1	1
F31	0	Mean	0.0001254882834238 (+)	0.0000000000000000	0.0077905311094958 (+)	0.0020185116261490 (+)	0.0002674563703837 (+)	0.0000000000000000	0.0000000111676630 (+)	0.0071082040000000 (+)	0.0000000000000000
		Rank	3	1	7	5	4	1	2	6	1
F32	0	Mean	0.0003548345513179 (+)	0.0701619169853449 (+)	0.0250163252527030 (+)	0.0013010316180679 (+)	0.0019635752485802 (+)	0.0016730768406953 (+)	0.0019955316015528 (+)	0.0002254250000000 (+)	0.0000989055208389
		Rank	3	9	8	4	6	5	7	2	1
F33	0	Mean	25.6367602258676000 (+)	95.9799861204982000 (+)	0.0000000000000000	1.1276202647057400 (+)	0.6301407361590880 (+)	0.8622978494808570 (+)	0.0000000000000000	0.0000000000000000	0.0000000000000000
		Rank	5	6	1	4	2	3	1	1	1
F34	0	Mean	2.6757043114269700 (+)	0.3986623855035210 (+)	0.2856833465904130 (+)	1.0630996944802500 (+)	5.7631786582751800 (+)	1.2137377447007000 (+)	0.3986623854300930 (+)	0.0000154715000000 (+)	0.0000000000000000
		Rank	8	5	3	6	9	7	4	2	1
F35	0	Mean	0.0000000000000000	0.4651202457398910 (+)	0.0000000000000000	0.0038863639514140 (+)	0.0019431819755029 (+)	0.0006477273251676 (+)	0.0000000000000000	0.0000000000000000	0.0000000000000000
		Rank	1	5	1	4	3	2	1	1	1
F36	0	Mean	−7684.6104757783800000 (+)	−6835.1836730901400000 (+)	−12569.4866181730000000 (+)	−12304.9743375341000000 (+)	−12210.8815698372000000 (+)	−12549.7468957373000000 (+)	−12569.4866181730000000 (+)	−12569.3622100000000000 (+)	−0.3402784042291390
		Rank	3	2	8	5	4	6	8	7	1
F37	0	Mean	0.0000000000000000	0.0000000000000000	14.5668734126948000 (+)	0.0000000000000000	6.4655746330439100 (+)	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
		Rank	1	1	4	1	3	1	1	1	1
F38	0	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000005 (+)	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
		Rank	1	1	2	1	1	1	1	1	1
F39	−10.5364	Mean	−10.1061873621653000 (+)	−5.2607563471326400 (+)	−10.5364098166920000	−10.3130437162426000 (+)	−10.3130437162026000 (+)	−10.5364098166921000	−10.5364098166921000	−10.5063235800000000 (+)	−10.5362816676181000
		Rank	4	5	1	3	3	1	1	2	1
F40	−10.1532	Mean	−9.5373938082045500 (+)	−5.7308569926624600 (+)	−10.1531996790582000	−9.5656135761215700 (+)	−10.1531996790582000	−9.9847854277673500 (+)	−10.1531996790582000	−10.1529842600000000	−10.1531669871808000
		Rank	4	5	1	3	1	2	1	1	1
F41	−10.4029	Mean	−10.4029405668187000	−6.8674070870953700 (+)	−10.4029405668187000	−9.1615813354737300 (+)	−10.4029405668187000	−10.4029405668187000	−10.4029405668187000	−10.3988303400000000 (+)	−10.4028748144797000
		Rank	1	4	1	3	1	1	1	2	1
F42	−186.7309	Mean	−186.7309073569880000	−81.5609772893002000	−186.7309088310240000	−186.7309088310240000	−186.7309088310240000	−186.7309088310240000	−186.7309088310240000	−186.2926481000000000	−186.7153981691330000
		Rank	1	4	1	1	1	1	1	3	2
F43	−1.03016	Mean	−1.031628453489880 (+)	−1.0044229658530100 (+)	−1.0316284534898800 (+)	−1.0316284534898800 (+)	−1.0316284534898800 (+)	−1.0316284534898800 (+)	−1.0316284534898800 (+)	−1.0304357800000000	−1.0303924506027700
		Rank	2	3	2	2	2	2	2	1	1
F44	0	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000004 (+)	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
		Rank	1	1	2	1	1	1	1	1	1

(*continued on next page*)

Table 5 (*continued*)

$F(x)$	Optimal	Statistics	PSO2011	CMAES	ABC	JDE	CLPSO	SADE	BSA	IA	SELO
F45	0	Mean	2.3000000000000000	0.0666666666666667	0.0000000000000000	0.9000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000538870000000	0.0000000000000010
		Rank	6	4	1	5	1	1	1	3	2
F46	0	Mean	0.1333333333333330 (+)	0.2666666666666670 (+)	0.0000000000000000	0.0000000000000000	0.2000000000000000 (+)	0.0000000000000000	0.0000000000000000	−0.015346330160966 (+)	0.0000000000000000
		Rank	3	5	1	1	4	1	1	2	1
F47	0	Mean	0.0000000000000000	0.0000000000000000	0.0000000000000005 (+)	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
		Rank	1	1	2	1	1	1	1	1	1
F48	−50	Mean	−50.0000000000002000	−50.0000000000002000	−49.9999999999997000	−50.0000000000002000	−49.4789234062579000	−50.0000000000002000	−50.0000000000002000	−44.7416748700000000	−46.6720228117341000
		Rank	1	1	1	1	2	1	1	4	3
F49	−200	Mean	−210.0000000000010000	−210.0000000000030000	−209.9999999999470000	−210.0000000000030000	−199.5925885475030000	−210.0000000000030000	−210.0000000000030000	−150.5540859185450000	−162.5712668655060000
		Rank	3	4	2	4	1	4	4	6	5
F50	0	Mean	0.0000000000000000	0.0000000000000000	0.0000000402380424 (+)	0.0000000000000000	0.0000000001597805 (+)	0.0000000000000000	0.0000000000000000	0.0000000000000000	0.0000000000000000
		Rank	1	1	3	1	2	1	1	1	1
Rank (Point)			5(**22**)	8(**11**)	4(**24**)	7(**20**)	7(**20**)	3(**31**)	1(**36**)	6(**21**)	2(**35**)

The '+' sign indicates that while comparing the performance of SELO with other algorithm, value of t_0 with the corresponding ν is significant at $\alpha = 0.5$ by the two-sample t-test.

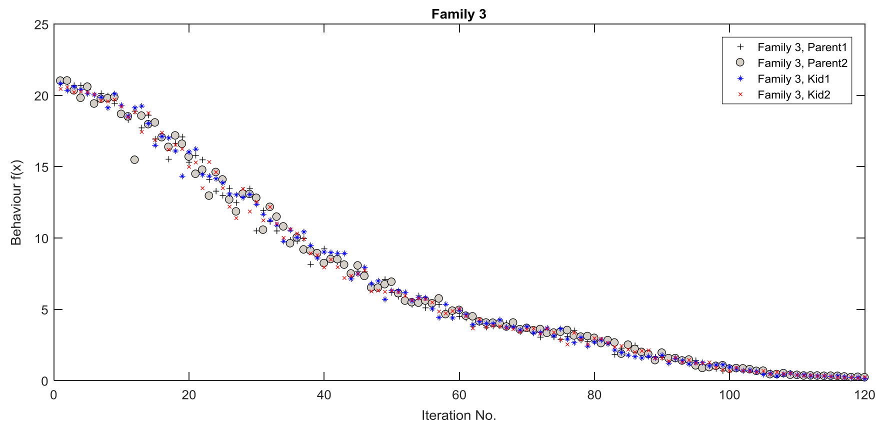

Fig. 7. Plot showing progress of behaviour values for Ackley function for Family-3.

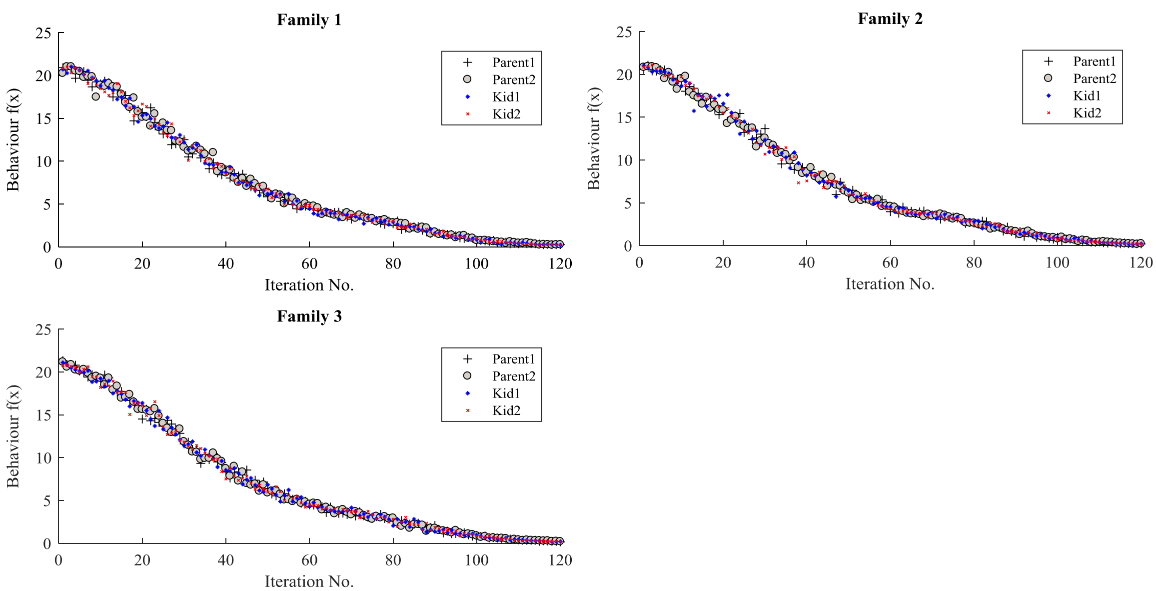

Fig. 8. Plot showing convergence progress pattern for all three families.

many test functions. Taking into thought these rankings (overall rankings as well as the places where the rank is '1') SELO emerges as a competitive and promising metaheuristic. Considering that SELO is a recent and a young algorithm and is still maturing, this performance is very significant. Fig. 7 illustrates the convergence plot of SELO during a run of solving the Ackley function. It exhibits the ability of the algorithm to jump out of the local minima and reach the global minima or the optimal solution. In the figure, we show the plot of how the behaviours ($f(x)$) progress and steadily move towards convergence for family number 3 (comprising of total four members). The emulation and learning behaviour of all the families is illustrated using the graphical convergence plot in Fig. 8.

From the results of Table 5 it can be observed that the average performance of SELO is still acceptable especially as compared to an advanced algorithm such as BSA (which has shown superior performance). The average problem solving ability of SELO is better than BSA and lead on 10 benchmark problems ($f5, f18, f23, f24, f26, f30, f31, f32, f34, f36$). SELO underperforms in 12 functions in comparison to BSA (where it scores the highest rank of 1). On functions $f3, f4, f6, f12, f14, f19, f20, f27, f28, f29, f42, f48$ SELO ranks lower than BSA.

As can be seen from the results of Table 4, other than SELO none of the comparator algorithms are able to attain the optimal global minima for Ackley and Rosenbrock. The Ackley and the Rosenbrock are non-separable functions where interdependency is seen among the variables making them relatively difficult to solve. Both these problems are considered as challenging for optimization problems and often used as benchmarks for testing. Ackley becomes even more difficult since it is also multimodal. If the exploration process of an algorithm is poorly designed, then it cannot search the function landscape effectively. The function of Ackley poses a risk for optimization algorithms of getting stuck in one of its many local minima and the algorithm must be able to steer the search, as far as, towards the global minima. For Rosenbrock, the global minimum lies in a narrow valley with the shape of a parabola. However even though this valley is easy to locate; due to its non-linearity converging to the minimum is difficult. In order to obtain consistent and good results for these functions, the search strategy must coalesce the exploration and exploitation strategies effectively.

In the current study, none of the metaheuristics attain the global optimum for Schwefel, Michalewicz2 and the noisy Quartic function; only SELO gives the closest approximations to the global solution. The Schewefel with several local minima is particularly hard because the global minimum is at the bounds of the search space and not in the origin as is the case with most other test problems [75]. Michalewicz function a multimodal test function with $n!$ local minima and its complexity increases with a larger parameter leading to a more difficult search. For the Michalewicz5 and the Michalewicz10, the proposed SELO methodology underperforms where again all the other methodologies under comparison have been unable to attain the expected global. The closest approximations of the global best have been recorded for BSA and ABC with very small S.D. from the mean. The test problems for which other algorithms perform on average better than SELO are the Penalized and Penalized2 function. SELO fairs lower in the rankings for these optimization functions where again, most of the methodologies have been unable to attain the expected global. The Penalized functions are challenging to solve since it composed of a combination of different sine based functions. Only BSA and CLPSO and ABC reach the global optimum value of zero. No algorithm could meet the success criteria in optimizing the Unimodal Dixon–Price problem. It may also be noted that SELO generally shows comparable (or even better) success rate in solving problems with higher dimensionality. With increase in the number of parameters or dimension, there is exponential increase in the search space. On the Powersum($f31$) test function, Powell($f30$) and Langermann10($f24$), very few algorithms find the global optimum very consistently, including the SELO. The flatness of the Powersum function makes it especially tricky to optimize since the flat surface does not guide the search of the algorithm towards the optimum minima. Only SELO, CMAES and SADE locate the global optimum (refer Table 4). The non-separable function of Powell has very diminutive minima as compared to the entire search-space making it difficult for the algorithms to locate the global minima and SELO, CMAES and IA attain the minima on all the trials runs conducted. SELO and IA show very consistent results on the multimodal, non-separable function Langermann ($f24$) which is characterized with many randomly distributed local minima, just like the Schwefel function. The function's complexity lies in the fact that there is no implicit symmetry advantage that may simplify the optimization for the algorithms. This indicates the strength of SELO in diversifying the search to avoid getting trapped in a local optimum and reaching the global optimal in a speedy manner. The strong exploration capabilities of the proposed methodology lies in the fact that even though the families in the society are all aiming for a single goal, they are all diverse in their behaviour and attributes. A set of different families with individual members may search the largely unknown region successively increasing the chances of arriving at the global best many folds. This diversity in families in a social setup adds to the exploration ability of the SELO. The algorithm spans the search space very effectively due to the diverse behaviour of different family individuals. This exploration capability is also evident from the smaller run time of the algorithm for most of the test problems. An analysis of the Table 5 also directs that if the local searching (exploitation) capabilities of SELO are further tuned and enhanced, the success of the proposed methodology may be improved significantly. This is evident in the functions where SELO ranks a second or third based on the quality of the solution and its closeness to the global convergence value. Currently, the average performance of SELO in certain problems may be attributed to its limitations to intensify the search in the neighbourhood of local region where the global optimum may lie. This upcoming methodology as of now does not make use of local gradients or derivatives or the history of the search process intensively which may give it the strength of exploiting the local information. Thus there is a research direction that if SELO is equipped with even more local exploitation capability, the method would see more success.

The authors also compare the mean outcomes for each of the algorithms under study by using an approximate two-sample t-test which was used for pair wise comparisons, with the statistical significance value $\alpha = 0.5$. A statistical analysis as adopted by Kashan [42] and Zhang et al. [76] is used to test if the proposed algorithm performed statistically better than a comparator algorithm at solving any and every benchmark test problem considered in the study. Table 5 presents the results of the two-sample t-test to establish the significance of difference between the proposed SELO and the other algorithms under comparison. The following statistics are used:

$$t_0 = \frac{\overline{y_1} - \overline{y_2}}{\sqrt{\left(\frac{S_1^2}{n_1}\right) + \left(\frac{S_2^2}{n_2}\right)}}. \tag{3}$$

Where $\overline{y_1}$ and $\overline{y_2}$ are mean values. S_1 and S_2 are standard deviations of the results obtained by two algorithmic approaches under comparison. The independent runs are denoted by n_1 and n_2. Then the degree of freedom ν is computed as under:

$$\nu = \left[\frac{1}{\left(\left(\left(\frac{S_1^2/n_1}{S_1^2/n_1 + S_2^2/n_2} \right)^2 \middle/ n_1 \right) + \left(\left(\frac{S_2^2/n_2}{S_1^2/n_1 + S_2^2/n_2} \right)^2 \middle/ n_2 \right) \right)} \right] \tag{4}$$

In the Table 5, each of the columns (which corresponds to one comparator algorithm) an '+' sign appears in a certain cell. This '+' sign indicates that while comparing the performance of SELO versus the corresponding algorithm, the value of t_0 with the corresponding ν is significant at $\alpha = 0.5$ by our approximate two-sample t-test. The t-tests have not been conducted for the problems where both the algorithms have obtained the optimal solutions across all trials of a particular function. On the cases where the t-test is conducted, the results in Table 5 indicate significant differences between the sample means of SELO and other competitor algorithms. The pair-wise difference between the results achieved by SELO versus BSA is significant on 11 problems. For SELO versus SADE this record is 12, for SELO versus IA it is 14 problems, for SELO versus PSO2011 it is17, for SELO versus ABC it is 19. SELO is a winner over JDE on 21 problems; similarly this record is 23 for CLPSO and 26 for CMAES. These results of the two-sample t-test are in agreement to the rank wise results as presented earlier. The rank wise system was used to represent an ordering amongst the competing algorithms in the current study. Thus in conclusion, it may be stated that SELO algorithm is very much comparable in performance to other optimizer algorithms examined in this study.

4. Conclusions and future direction

In the paper, a new socio-inspired methodology referred to as Socio Evolution and Learning Optimization (SELO) is proposed which mimics the natural social tendency of humans organized as family groups. It is motivated by the evolution of social behaviour of every individual in a family. The parents and children of a family evolve (become better) by observing and learning from one another as well as from other families. A group of families co-existing together may be called as a society. In this societal setup, the individuals learn and adapt to the behaviours of other individuals in the same as well as the other family individuals. Such socio-behavioural models inspired from social tendencies in humans and their social interactions are fairly recent developments. SELO captures the essence that social interactions enable individuals to adapt and evolve faster through social evolution than biological evolution based on genetic inheritance alone. Two-way evolution is a key feature of SELO, as it allows for a two way system of learning and adaptation to take place. Evolution and learning takes place at both: social level and population level (for every individual). When a kid is created it is genetically similar to its parent and exhibits similar fitness; however at behaviour level may later evolve differently inspired by his peers' social behaviours. This fundamental feature of SELO sets it apart from other existing EA's. In this work, the preliminary and the basic version of SELO is tested and compared with few other accepted optimization algorithms on several benchmark problems. The results ascertain the potential and logical correctness of the proposed work. The empirical study shows that the novel SELO algorithm shows promising performance as measured against the other comparator algorithms. The convergence rate is acceptable and the algorithm guides the search towards more promising areas of the solution space and the optimal minimum in a reasonable amount of time. Since the population consists of diverse families, this diversity lends a faster and rapid exploration capability to SELO. This can be of advantage to large scale problems with larger sample spaces. Hence the authors anticipate that large scale problems can benefit from SELO where the solving time can be considered to be a significant factor. Also, the intrinsic parameters to be adjusted for SELO are comparatively less and thus SELO can be tailored to solve diverse optimization problems with significantly less modifications and parameter tuning.

However SELO is still in its very initial stages and further efforts are needed to exploit the full effectiveness of the methodology to attain even better problem-solving success. The results section discusses that if the local search capability is intensified; this novel algorithm will achieve more success and provides a clear direction for further research. The authors need to further establish the effectiveness of this metaheuristic by solving purposeful and real world problems using this. A key task would be to identify real world problems where advantage of adopting SELO is obvious, considering its strengths and limitations. This could also prove to be a major direction for future research. The current version of SELO uses parameters such as parent_follow_probability r_p, peer_follow_probability r_k, follow_prob_factor_ownparent, follow_prob_factor_otherkids and sampling interval reduction factor r (refer Table 3) which were fine tuned to attain desired performance of the algorithm. Several preliminary trials were carried out to choose an acceptable combination of these parameters and a strong supporting algorithm/mechanism is needed to optimize these parameters. SELO could be designed as a self-tuning algorithm; with an adaptive feature which enables the search algorithm to optimize and self tune the parameters at run time. Thus as the algorithm approaches convergence, the control parameters would be iteratively reduced. This may help the algorithm exhibit expanded local search i.e. improved exploitation. Another research direction can be to modify SELO to integrate it with the generalized constrained handling techniques [77] so that it can be used to solve constrained problems too. Popular constraint-handling techniques like penalty function and multi-objective approaches may be considered. Most nature inspired evolutionary algorithms are adept at solving unconstrained problems; but their performance may be impacted when attempting constrained optimization problems. Thus designing and adaptation of evolutionary algorithms to constrained optimization is a very clear research direction. The authors anticipate that the paper will also motivate other researchers to mature the further theory and applications of SELO.

References

[1] S. Luke, Essentials of Metaheuristics, Lulu, second ed., 2013 Available for free at http://cs.gmu.edu/~sean/book/metaheuristics/.

[2] E.G. Talbi, Metaheuristics: From Design to Implementation, Vol. 74, John Wiley & Sons, 2009.

[3] X.S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver press, 2010.

[4] J. Brownlee, Clever Algorithms: Nature-Inspired Programming Recipes, Jason Brownlee, Australia, 2011.

[5] I. Fister Jr., X.S. Yang, I. Fister, J. Brest, D. Fister, A Brief Review of Nature-Inspired Algorithms for Optimization, 2013. ArXiv preprint arXiv:1307.4186.

[6] S.J. Nanda, G. Panda, A survey on nature inspired metaheuristic algorithms for partitional clustering, Swarm Evol. Comput. 16 (2014) 1–18.

[7] H. Zang, S. Zhang, K. Hapeshi, A review of nature-inspired algorithms, J. Bionic Eng. 7 (Suppl.) (2010) S232–S237.

[8] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, No. 1, Oxford University Press Santa Fe, USA, 1999.

[9] A.E. Hassanien, E. Emary, Swarm Intelligence: Principles, Advances, and Applications, CRC Press, 2016.

[10] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (4) (2006) 28–39.

[11] C. Filho, F. de Lima Neto, A. Lins, A. Nascimento, M. Lima, Fish school search, in: Nature-Inspired Algorithms for Optimisation, Vol. 193, Springer, Berlin, Heidelberg, 2009, pp. 261–277.

[12] A. Biswas, K.K. Mishra, S. Tiwari, A.K. Misra, Physics-inspired optimization algorithms: a survey, J. Optim. (2013) Article ID 438152, 16 pages. http://dx.doi.org/10.1155/2013/438152.

[13] S.P. Brooks, B.J. Morgan, Optimization using simulated annealing, Statistician 44 (2) (1995) 241–257. http://dx.doi.org/10.2307/2348448.

[14] W.L. Goffe, G.D. Ferrier, J. Rogers, Global optimization of statistical functions with simulated annealing, J. Econometrics 60 (1–2) (1994) 65–99.

[15] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.

[16] Z.W. Geem, State-of-the-art in the structure of harmony search algorithm, in: Recent Advances in Harmony Search Algorithm, Studies in Computational Intelligence, Springer, Berlin, Heidelberg, 2010.

[17] X.S. Yang, Harmony search as a metaheuristic algorithm, in: Music-Inspired Harmony Search Algorithm, in: Studies in Computational Intelligence, vol. 191, Springer, Berlin, Heidelberg, 2009, pp. 1–14.

[18] A.Y.S. Lam, V.O.K. Li, Chemical-reaction-inspired metaheuristic for optimization, IEEE Trans. Evol. Comput. 14 (3) (2010) 381–399.

[19] M. Gendreau, J.Y. Potvin, HandBook of Metaheuristics, Vol. 2, Springer, New York, 2010.

[20] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948.

[21] N. Moosavian, B.K. Roodsari, Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks, Swarm Evol. Comput. 17 (2014) 14–24.

[22] F. Marini, B. Walczak, Particle swarm optimization (PSO), in: A Tutorial, Chemometrics and Intelligent Laboratory Systems, Vol. 149, 2015, pp. 153–165.

[23] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295.

[24] Clerc, Maurice, Standard particle swarm optimization, 2012, 15 pages, <hal-00764996>.

[25] M.G.H. Omran, M. Clerc, 2011. https://www.particleswarm.info/Programs.html. (Accessed 15 December 2016).

[26] N. Hansen, The CMA evolution strategy: a comparing review, in: Towards a New Evolutionary Computation, in: Studies in Fuzziness and Soft Computing, vol. 192, Springer, 2006, pp. 75–102.

[27] C. Igel, N. Hansen, S. Roth, Covariance matrix adaptation for multi-objective optimization, Evol. Comput. 15 (1) (2007) 1–28.

[28] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (2007) 459–471.

[29] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Appl. Soft Comput. 8 (1) (2008) 687–697.

[30] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.

[31] F. Neri, V. Tirronen, Recent advances in differential evolution: a survey and experimental analysis, Artif. Intell. Rev. 33 (1–2) (2010) 61–106.

[32] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (5) (2009) 945–958.

[33] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (6) (2006) 646–657.

[34] A. Kai Qin, N. Suganthan Ponnuthurai, Self-adaptive differential evolution algorithm for numerical optimization, in: Evolutionary Computation (CEC), 2005 IEEE Congress, Vol. 2, IEEE, Edinburgh, UK, 2005, pp. 1785–1791.

[35] P. Civicioglu, Backtracking search optimization algorithm for numerical optimization problems, Appl. Math. Comput. 219 (15) (2013) 8121–8144.

[36] R.G. Reynolds, W. Sverdlik, Problem solving using cultural algorithms, in: Evolutionary Computation, IEEE World Congress on Computational Intelligence, Proceedings of the First IEEE Conference, Orlando, FL, USA, 1994, pp. 645–650.
[37] T. Ray, K.M. Liew, Society and civilization: An optimization algorithm based on the simulation of social behavior, IEEE Trans. Evol. Comput. 7 (4) (2003) 386–396.
[38] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: Evolutionary Computation, CEC, 2007 IEEE Congress, IEEE, Singapore, 2007, pp. 4661–4667.
[39] S. Hosseini, A. Al Khaled, A survey on the imperialist competitive algorithm metaheuristic: Implementation in engineering domain and directions for future research, Appl. Soft Comput. 24 (2014) 1078–1094.
[40] A.H. Kashan, League championship algorithm: a new algorithm for numerical function optimization, in: International Conference on Soft Computing and Pattern Recognition, SOCPAR09, IEEE, Singapore, 2009, pp. 43–48.
[41] A.H. Kashan, An efficient algorithm for constrained global optimization and application to mechanical engineering design: League championship algorithm (LCA), Comput. Aided Des. 43 (12) (2011) 1769–1792.
[42] A.H. Kashan, League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships, Appl. Soft Comput. 16 (2014) 171–200.
[43] Y. Xu, Z. Cui, J. Zeng, Social emotional optimization algorithm for nonlinear constrained optimization problems, in: Swarm, Evolutionary, and Memetic Computing, SEMCCO 2010, in: Lecture Notes in Computer Science, vol. 6466, Springer, Berlin, Heidelberg, 2010, pp. 583–590.
[44] W. Lv, C. He, D. Li, S. Cheng, S. Luo, X. Zhang, Election campaign optimization algorithm, Procedia Comput. Sci. 1 (1) (2010) 1377–1386.
[45] Q. Xie, W. Lv, Z. Liu, X. Zhang, S. Luo, S. Cheng, Constrained optimization with election campaign algorithm, in: Industrial Mechatronics and Automation (ICIMA), 2nd International Conference, Vol. 1, IEEE, Wuhan, China, 2010, pp. 370–373.
[46] A. Ahmadi-Javid, Anarchic society optimization: A human-inspired method, in: Evolutionary Computation, CEC, 2011 IEEE Congress, IEEE, New Orleans, USA, 2011, pp. 2586–2592.
[47] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems, Inform. Sci. 183 (1) (2012) 1–15.
[48] A.J. Kulkarni, G. Krishnasamy, A. Abraham, Cohort Intelligence: A Socio-Inspired Optimization Method, Vol. 114, Intelligent systems reference library, Springer, 2016, pp. 1–134.
[49] A.J. Kulkarni, H. Shabir, Solving 0–1 knapsack problem using cohort intelligence algorithm, Int. J. Mach. Learn. Cybern. 7 (3) (2016) 427–441.
[50] N. Moosavian, Soccer league competition algorithm for solving knapsack problems, Swarm Evol. Comput. 20 (2015) 14–22.
[51] H. Emami, F. Derakhshan, Election algorithm: a new socio-politically inspired strategy, AI Commun. 28 (3) (2015) 591–603.
[52] Z.Z. Liu, D.H. Chu, C. Song, X. Xue, B.Y. Lu, Social learning optimization (SLO) algorithm paradigm and its application in QoS-aware cloud service composition, Inform. Sci. 326 (2016) 315–333.
[53] S. Satapathy, A. Naik, Social group optimization (SGO): a new population evolutionary optimization technique, Complex Intel. Syst. 2 (3) (2016) 173–203.
[54] T.T. Huan, A.J. Kulkarni, J. Kanesan, et al., Ideology algorithm: a socio-inspired optimization methodology, Neural Comput. Appl. (2016) 1–32. http://dx.doi.org/10.1007/s00521-016-2379-4.
[55] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[56] M. Koppen, D.H. Wolpert, W.G. Macready, Remarks on a recent paper on the "no free lunch" theorems, IEEE Trans. Evol. Comput. 5 (3) (2001) 295–296.
[57] Y.C. Ho, D.L. Pepyne, Simple explanation of the no-free-lunch theorem and its implications, J. Optim. Theory Appl. 115 (3) (2002) 549–570.
[58] Goldsmith Edward, The Stable Society: Its Structure and Control: Towards a Social Cybernetics, Wadebridge Press, Wadebridge, Eng, 1978.
[59] A.J. Kulkarni, K. Tai, Probability collectives: a multi-agent approach for solving combinatorial optimization problems, Appl. Soft Comput. 10 (3) (2010) 759–771.
[60] M. Hechter, C. Horne, Theories of Social Order: A Reader, second ed., Stanford University Press, Stanford, CA, 2009.
[61] P. Civicioglu, E. Besdok, A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms, Artif. Intell. Rev. 39 (315) (2013) 1–32. http://dx.doi.org/10.1007/s10462-011-9276-0.
[62] M. Eisenberg, The peer assumption: A review of the nurture assumption, J. Learn. Sci. 17 (4) (2008) 588–594. http://dx.doi.org/10.1080/10508400802394906.
[63] A. Bandura, Social learning through imitation, in: M.R. Jones (Ed.), Nebraska Symposium on Motivation, University of Nebraska Press, Lincoln, 1962.
[64] A. Bandura, R.H. Walters, Social Learning Theory, General Learning Press, New York, 1977.
[65] E.E. Maccoby, The role of parents in the socialization of children: An historical overview, Dev. Psychol. 28 (6) (1992) 1006–1017. http://dx.doi.org/10.1037/0012-1649.28.6.1006.
[66] D.E. Goldberg, K. Deb, A comparative analysis of selection schemes used in genetic algorithms, in: Foundations of Genetic Algorithms, Vol. 1, Morgan Kaufmann Publishers Inc., 1991, pp. 69–93.
[67] T. Pencheva, K. Atanassov, A. Shannon, Modelling of a roulette wheel selection operator in genetic algorithms using generalized nets, Int. J. Bioautomation 13 (4) (2009) 257–264.
[68] R. Kumar, Blending roulette wheel selection & rank selection in genetic algorithms, Int. J. Mach. Learn. Comput. 2 (4) (2012) 365.
[69] H.C. Kuo, C.H. Lin, Cultural evolution algorithm for global optimizations and its applications, J. Appl. Res. Technol. 11 (4) (2013) 510–522.
[70] A.J. Kulkarni, I.P. Durugkar, M. Kumar, Cohort intelligence: a self supervised learning behavior, in: Systems, Man, and Cybernetics, SMC, IEEE International Conference, IEEE, Manchester, UK, 2013, pp. 1396–1400.
[71] X.S. Yang, Appendix A: test problems in optimization, Eng. Optim. (2010) 261–266.
[72] M. Jamil, X.S. Yang, A literature survey of benchmark functions for global optimisation problems, Int. J. Math. Model. Numer. Optim. 4 (2) (2013) 150–194.
[73] T. Krink, B. Filipic, G.B. Fogel, Noisy optimization problems-a particular challenge for differential evolution?in: Congress on Evolutionary Computation, CEC2004, Vol. 1, IEEE, 2004, pp. 332–339.
[74] S. Surjanovic, D. Bingham, British Columbia, 2015. https://www.sfu.ca/~ssurjano/optimization.html. (Accessed 15 January 2017).
[75] M. Molga, C. Smutnicki, Test functions for optimization needs, 2005, pp. 101.
[76] M. Zhang, W. Luo, X. Wang, Differential evolution with dynamic stochastic selection for constrained optimization, Inform. Sci. 178 (15) (2008) 3043–3074.
[77] A.M. Deshpande, G.M. Phatnani, A.J. Kulkarni, Constraint handling in firefly algorithm, in: Cybernetics, CYBCONF, 2013 IEEE International Conference, IEEE, United States, 2013, pp. 186–190.

Anand J. Kulkarni holds a Ph.D. in Distributed Optimization from Nanyang Technological University, Singapore, M.S. in Artificial Intelligence from University of Regina, Canada, Bachelor of Engineering from Shivaji University, India and Diploma from the Board of Technical Education, Mumbai. He worked as a research fellow on a cross-border supply-chain disruption project at Odette School of Business, University of Windsor, Canada. Currently, he is working as Head and Associate Professor at the Symbiosis Institute of Technology, Symbiosis International University, Pune, India. His research interests include optimization algorithms, multi-objective optimization, continuous, discrete and combinatorial optimization, multiagent systems, complex systems, cohort intelligence, probability collectives, swarm optimization, game theory, self-organizing systems and fault-tolerant systems. He is the founder and chairman of the Optimization and Agent Technology (OAT) Research Lab. Anand has published over 30 research papers in peer-reviewed journals and conferences. He also published two research books.