Swarm and Evolutionary Computation 39 (2018) 24–35

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Survey Paper

Population topologies for particle swarm optimization and differential evolution

Nandar Lynn [a], Mostafa Z. Ali [b], Ponnuthurai Nagaratnam Suganthan [a,*]

[a] *School of Electrical & Electronics Engineering, Nanyang Technological University, Singapore*
[b] *Jordan University of Science & Technology, Jordan*

ARTICLE INFO

Keywords:
Particle swarm optimization
Differential evolution
Optimization
Population topology
Social network
Cellular
Distributed
Static
Dynamic
Ring
Wheels
Random
von Neumann
Star
Hierarchical
Niching
Multiswarm
Subswarm
Subpopulation
Heterogeneous

ABSTRACT

Over the last few decades, many population-based swarm and evolutionary algorithms were introduced in the literature. It is well known that population topology or sociometry plays an important role in improving the performance of population-based optimization algorithms by enhancing population diversity when solving multiobjective and multimodal problems. Many population structures and population topologies were developed for particle swarm optimization and differential evolutionary algorithms. Therefore, a comprehensive review of population topologies developed for PSO and DE is carried out in this paper. We anticipate that this survey will inspire researchers to integrate the population topologies into other nature inspired algorithms and to develop novel population topologies for improving the performances of population-based optimization algorithms for solving single objective optimization, multiobjective optimization and other classes of optimization problems.

1. Introduction

Evolutionary algorithms (EAs) are population-based metaheuristic optimization algorithms inspired by Darwin's theory of evolution [1–3]. The individuals in the population represent the potential solutions to an optimization problem. The individuals collaborate with and compete against each other to find the optimal solution in the search space. Survival in the population is based on the quality of the solutions which is determined by the fitness function. The better candidate solutions are selected to breed offspring for the next generation by applying stochastic variation operators to them. The mutation operator is applied to one selected candidate solution to generate to a new candidate solution. The recombination operator is applied to two or more selected candidate solutions to produce one or more new candidate solutions. In this way, evolution of the population takes place via the processes of selection, mutation and recombination and the population moves toward better solutions in the search space.

Darwin described the importance of a population structure on evolution in Ref. [4] as well as it is well-known in the literature that the performance of an evolutionary algorithm is significantly influenced by the organization of the individuals in the population [5–9]. The simple and standard population structure used in Evolutionary Algorithms (EAs) is the *panmictic* structure. In the panmictic population structure, mating selection is random and all the individuals can interact with any other individual in the population during the evolutionary process. Therefore, this structure promotes rapid information flow and consequently the population loses its diversity potentially leading to premature convergence. In order to address this problem, researchers suggested that like

* Corresponding author.
E-mail addresses: nandar001@ntu.edu.sg (N. Lynn), mzali.pn@gmail.com (M.Z. Ali), epnsugan@ntu.edu.sg (P.N. Suganthan).

https://doi.org/10.1016/j.swevo.2017.11.002
Received 23 January 2016; Received in revised form 29 October 2017; Accepted 5 November 2017
Available online 6 November 2017

natural biological populations, it is possible to define a population structure with neighborhood or topological properties [5,8]. In the structured populations, mating selection depends on fitness as well as topological relationships and mating is more common between the individuals that are close [10,11]. Thus, spreading of good solutions throughout the population is slowed down thereby discouraging stagnation and premature convergence. The structured population models can be divided into two main groups: cellular/fine-grained model [12,13] and distributed/coarse-grained model [7,8]. The population topologies of cellular and distributed models are presented in Fig. 1.

The cellular/fine-grained model is also known as the *diffusion* model [14]. In cellular models, the population is distributed in a grid and each individual has a unique coordinate in the grid. The individuals are allowed to interact only within their neighborhood. Thus, due to the slow diffusion of information through the grid from neighborhood to neighborhood, the risk of premature convergence is lowered [5].

The distributed/coarse-grained models are also known as *island* model or *multipopulation* model [5,15]. In the distributed model, the panmictic population is divided into several smaller subpopulations (islands). Separate EAs evolve independently on different islands and individuals are allowed to migrate between the islands according to a given migration policy. In this way, the divided subpopulations can explore different regions of the search space while sharing information by means of migration. Hence, the population diversity can be maintained [15].

Also, it is possible to combine the above two models. For instance, the island model can be considered in which the islands themselves have cellular models. Alternatively, it is possible to have the hierarchical island model in which islands are further divided into islands and the migration only occurs between islands at the same level. The hierarchical models are also known as hybrid models [5,8]. How individuals or subpopulations communicate in the structured population model can be represented by the communication graph which specifies the neighborhood for each individual/each subpopulation and represents the interrelationships between the individuals/subpopulations in the cellular/distributed model. Further, population topologies can be either static or dynamic. Topologies defined based on indices of population members can be static [16] or dynamic [17,18] while topologies defined based on Euclidean distance in the decision variable space [19–21] or objective space [22,23] are dynamic.

The fields of evolutionary and swarm algorithms include evolution strategy (ES), differential evolution (DE), genetic algorithm (GA), particle swarm optimization (PSO), evolutionary programming (EP), genetic programming (GP) and several others. As mentioned above, performance of population-based metaheuristic optimization algorithms is highly influenced by the population structures and topologies used. This issue has been actively investigated in the literature [3,5–9,24–29].

Among the population-based algorithms, DE and PSO are inspired by the biological and sociological motivations. In mid 1990s, Eberhart and Kennedy proposed particle swarm optimization (PSO) algorithm inspired by the swarm social behavior of bird flocking and fish schooling [30,31]. PSO is known for having the ability to quickly converge to an optimum [30]. PSO has no evolution operators such as crossover or mutation and uses simple mathematical operations for updating over iterations. Several variants of PSO have been developed to improve the performance of the original PSO in Refs. [32,33].

Around the time PSO was proposed, Storn and Price introduced the differential operator to replace the mutation operator in GA and proposed differential evolution (DE) algorithm [34,35]. However, unlike PSO, DE uses mutation and recombination operators to evolve the population and is similar to other EAs like ES and EP. The feature that distinguishes the DE algorithm from the traditional EAs is its mutation mechanism. In DE, a difference vector is used. Another feature that differs from the traditional EAs is that DE employs the parent-offspring competition to determine whether or not the offspring solution survives to the next generation. Different variants of DE algorithms have been proposed to improve its performance in the literature [36–41] and it has been successfully extended to solve large-scale, dynamic, multi-objective, constrained and multimodal optimization problems [42–45].

It can be clearly seen from above that both PSO and DE algorithms are conceptually simple and use only simple operators. Besides, they are easy to implement in any computer language and require less parameter tuning. Therefore, both optimization techniques have received great research interest and shown great promise in several real-world applications. However, like other population-based optimization algorithms, the population topology has a significant impact on their performance and their performance depends on the population topology used.

In PSO, a population of candidate solutions is randomly initialized and each individual search for optimal regions of the space by learning from its most successful neighbor(s) in the population. Thus, the population topology of PSO determines the breadth of influences on the individual and the performance of each individual depends on the used population topology [16,46–52]. DE and PSO with panmictic structure might suffer from premature convergence and the population may converge to a local optimum, losing its diversity. One of the solutions for this drawback is to use a structured population/population topology instead of panmictic structure, as evidenced in the literature [53–55].

Due to their benefits for population-based algorithms, numerous population topologies have been proposed for PSO and DE algorithms to solve different types of optimization problems. However, there is a lack of comprehensive literature review on the developed PSO and DE population topologies. Therefore, this study attempts to provide a

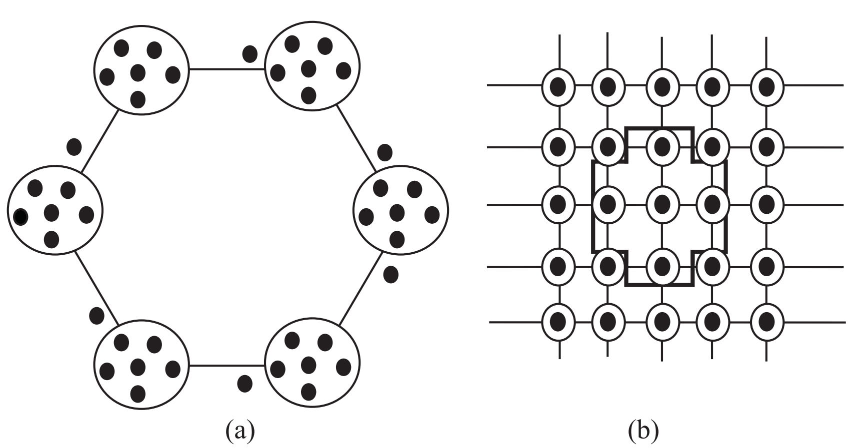

Fig. 1. (a) Distributed/Coarse-grained model, (b) Cellular/Fine-grained model.

comprehensive survey of population topologies proposed for PSO and DE and encourages the researchers to introduce these population topologies to other population-based algorithms to achieve better performance. The population topologies of other EAs are also briefly described in the paper. This survey focus on algorithmic aspects of the topologies while topologies necessitated for parallelization or cloud implementation are excluded.

The rest of this paper is arranged as follows. PSO and DE algorithms are briefly explained in Section 2. The cellular and distributed models of PSO, DE and other EAs are presented in Section 3. Other population topologies of random/irregular, hierarchical/hybrid and coevolution models are presented in IV. Finally, the performance of cellular and distributed population topologies proposed for PSO and DE are discussed in Section 5. The paper is concluded in Section 6.

2. Classical PSO and DE algorithms

2.1. Particle swarm optimization (PSO) algorithm

PSO is a population-based algorithm inspired by swarm social interactions. In PSO, each particle in the swarm population represents a potential solution to the given optimization problem. Initially, the particles are randomly distributed over the search space with random velocity values. Besides its position and velocity, each particle also stores the best position found so far in the search space. Then, each particle's velocity is adjusted towards its own previous best position and the best position found by any particle in its neighborhood. The neighborhood is determined by the used population topology. This phenomenon is formulated as follows [31]:

$$V_i^d = w^* V_i^d + c_1{}^* rand1_i^d{}^* \left(P_i^d - X_i^d\right) + c_2{}^* rand2_i^d{}^* \left(P_g^d - X_i^d\right) \tag{1}$$

$$X_i^d = X_i^d + V_i^d \tag{2}$$

where, i represents each particle in the population ($i = 1, 2, \ldots, N$) and d represents a dimension ($d = 1, 2, \ldots, D$). X_i^d and V_i^d are the dth position and velocity components of the ith particle in the population, respectively. P_i is the best position of the ith particle and P_g is the best position. c_1 and c_2 are the acceleration coefficients and $rand1_i^d$ and $rand2_i^d$ are randomly generated numbers in the range of [0, 1]. Inertia weight w is defined as a linearly decreasing function of the run time. In this way, PSO attempts to locate the optimal solution by simply adjusting the trajectory of each particle towards its own best and towards the best position of the swarm at each generation.

2.2. Differential evolution (DE) algorithm

DE employs evolutionary operators to explore different areas of the search space and gradually move its population towards better regions in the search space. Similar to other population-based EAs, DE algorithm starts the search with a population of candidate solutions which are randomly initialized within the search space using the following equation:

$$X_i^j = X_{min}^j + rand_i^j(0,1)^* \left(X_{max}^j - X_{min}^j\right) \tag{3}$$

where, $rand_i^j(0,1)$ represents a uniformly distributed random number which is in the range of [0, 1] and is instantiated independently for each dimension of each member. X_{max}^j and X_{min}^j are the upper and lower bounds on each decision variable. After initializing the population, DE employs mutation, crossover and selection as follows.

Mutation: Unlike the traditional EAs, DE adds difference of candidate solutions to another solution to generate a mutant vector. At each generation, DE mutation creates a mutant vector $V_{i,g}$ for each target vector $X_{i,g}$ by adding the weighted difference of two individuals to a third candidate solution as follows:

$$V_{i,g} = X_{r1,g} + F^* \left(X_{r2,g} - X_{r3,g}\right) \tag{4}$$

where, r_1, r_2 *and* r_3 are randomly selected from the population and $i \neq r_1 \neq r_2 \neq r_3$. The parameter F is a scaling factor which is a positive control parameter for scaling the difference vector. The DE mutation strategies are represented by the notation "*DE/a/b*", where "*a*" represents the base vector to be perturbed and "*b*" represents the number of differential vectors. The most frequently used mutation strategies are DE/rand-to-best/2 [35], DE/current-to-best/1 [37] and DE/current-to-*pbest* [36].

Crossover: After the mutation operation, the trial vector $U_{i,g}$ is generated by performing a crossover operation between the target vector $X_{i,g}$ and the mutant vector $V_{i,g}$. DE uses either exponential or binomial crossover operation to create a trial vector [35]. In exponential crossover operator, an integer l, randomly selected in $[1, D]$, acts as a starting index where the target vector exchanges the components with the mutant vector. An integer L is selected in the interval [1, D] based on the crossover rate C_r and represents the number of components that the mutant vector contributes to the target vector. With the exponential crossover operator, the trial vector $U_{i,g}$ is generated according to:

$$u_{i,j} = \begin{cases} v_{i,j}, \; for \;\; j = l_D, l_{D+1}, \ldots, l + L - 1_D \\ \quad x_{i,j} \, for \;\; all \;\; other \;\; j \in [1, D] \end{cases} \tag{5}$$

where, the angular brackets $\langle l \rangle_D$ denote a modulo function with modulus D with a starting index l. With the binomial crossover operator, the trial vector $U_{i,g}$ is generated for each dimension according to:

$$u_{i,j} = \begin{cases} v_{i,j} \; if \;\; rand_{i,j}(0,1) \le C_r \;\; or \;\; j = j_{rand} \\ \qquad x_{i,j} \;\; otherwise \end{cases} \tag{6}$$

where, $rand_{i,j}(0,1)$ is a uniformly distributed random number (between [0, 1]) which is generated for each dimension. j_{rand} is a randomly selected integer in $[1, D]$. C_r is the crossover control parameter in the range of [0, 1]. Due to *jrand*, the trial vector differs from its target vector in at least one dimension. If the trial vector $U_{i,g}$ is beyond the boundary of the search space, it is usually reinitialized within the range of $[X_{min}, X_{max}]$. ***Selection:*** DE's *selection operation is conducted by comparing the* target vector $X_{i,g}$ with the trial vector $U_{i,g}$ in terms of their fitness values. If $f(X)$ is an objective function to be minimized, whether the target vector $X_{i,g}$ or the trial vector $U_{i,g}$ is survived to the next generation is decided according to:

$$X_{i,g+1} = \begin{cases} U_{i,g} \;\; if \;\; f\left(U_{i,g}\right) \le f\left(X_{i,g}\right) \\ X_{i,g} \;\; if \;\; f\left(U_{i,g}\right) > f\left(X_{i,g}\right) \end{cases} \tag{7}$$

3. Population topologies for PSO, DE and other EAs

In this survey paper, the population topologies proposed for PSO and DE algorithms are organized as presented in Fig. 2. The two main structured population models, cellular and distributed models are presented in this survey. Moreover, the two models as used with other EAs are also briefly presented in this paper. In addition to these two main population models, other population topologies such as hierarchical,

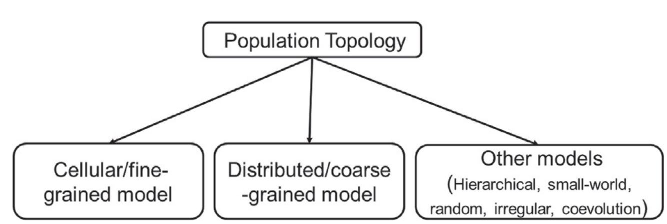

Fig. 2. Population topologies studied for PSO, DE and other population-based algorithms.

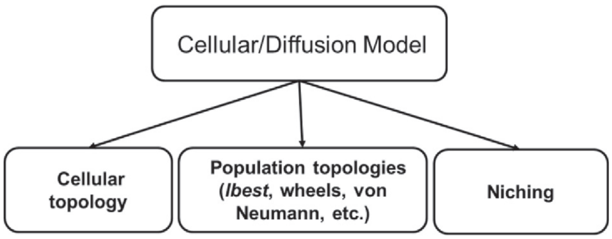

Fig. 3. Cellular/Diffusion models studied for PSO and DE algorithms.

small-world, random, irregular and coevolution models are also briefly reviewed to provide a complete comprehensive survey of population topologies developed for PSO and DE algorithms (see Fig. 3).

3.1. Cellular topology

In cellular topology, a population is divided into many very small subpopulations typically composed of only one individual. The individuals are arranged in the toroidal mesh and the interaction takes place only within the small neighborhood defined around each individual solution or subpopulation. Thus, the neighborhoods are partially overlapped in cellular topology and it promotes slow diffusion of information through the grid and helps to maintain the population diversity [13]. The popular cellular topology proposed for PSO, DE and other population-based algorithms are presented in this section.

In addition to cellular topology, niche PSO and niche DE are also presented as a subsection of cellular topology. The characteristic of the cellular (fine-grained architecture) model, in particular the inbreeding within a neighborhood, causes speciation and inclines the population to form the niches [56,57]. However, over time, strong characteristics developed locally within the neighborhoods gradually spread across the population and there is a tendency for the population to become homogeneous [46]. Thus, the cellular model is also termed as diffusion model. In cellular niches, the convergence rate becomes much slower and the slow homogenization helps the population to reduce the chances of premature convergence to a local optimum.

The idea of using cellular structured population instead of the use of panmictic population in EAs can also be applied in PSO. Therefore, besides *gbest* topology, other population topologies such as *lbest* and wheels are studied to understand their influence on performance of PSO in Ref. [16]. Inspired by this, the population topologies of particle swarms are also applied to DE. These population topologies can be viewed as the generalization of the diffusion model and thus, they are presented under the section of cellular topology.

3.1.1. Cellular PSO

Inspired by the cellular genetic algorithm [12], Waintraub proposed the cellular version of PSO called Cellular-PSO in Ref. [58]. In Cellular-PSO, the particles are arranged in a two-dimensional cellular topology. Unlike PSO algorithm, the best particle in the entire population is no more visible to all the particles. Instead, the particles are guided by the neighborhood local best particle within their respective neighborhood. As a consequence, the information exchange between the non-neighboring particles is delayed and the population diversity can be retained in Cellular-PSO. In Ref. [58], the cellular-PSO was applied to solve a thermal optimization problem and compared with standard PSO algorithm. The results showed that the cellular-PSO was more efficient and robust than the standard PSO algorithm. In Ref. [59], Hashemi also introduced two dimensional cellular automata into PSO algorithm and proposed Cellular PSO to address the diversity loss problem of PSO in dynamic environments. The performance of proposed Cellular PSO was tested on the moving peaks benchmark problem and cellular PSO outperformed compared mQSO, a well-known PSO model for dynamic environment [60]. Other cellular versions of PSO were also proposed for dynamic environments in Refs. [61,62].

In order to address the issue of getting trapped into a local optimum, Shi integrated the lattice cellular model into PSO and proposed cellular PSO algorithm called CPSO in Ref. [63]. In CPSO, three typical lattice cellular structures, namely cubic, trigonal, hexagonal, are used as the neighborhood topologies and the individuals can only exchange information within their neighborhood. The experiment is conducted using a large set of benchmark functions and the experimental results demonstrated that the CPSO performed better on most of the test functions compared to other PSO algorithms [63]. Later, CPSO was combined with constraint handling technique and proposed to solve a nonlinear constraint optimization problem in Ref. [64].

Recently, Fang presented a quantum-inspired PSO with cellular structured population called (cQPSO) in Ref. [65]. In cQPSO, the particles are distributed in a two-dimensional grid and restricted to communicate with their own neighbors locally. In Ref. [65], cQPSO was studied with six different cellular neighborhood topologies on 42 different benchmark problems. The performance of cQPSO was compared with a set of PSOs with different neighborhood topologies as well as other swarm-based algorithms. The experimental and statistical analysis results showed superior performance of the cQPSO with local best compared to other swarm-based algorithms [65].

In *gbest* topology [66], every particle is attracted to the best solution found by any other member of the entire population. All particles can communicate with each other and the topology can be represented by a fully-connected social network graph. Eberhart proposed an *lbest* topology in Ref. [67]. Unlike the *gbest* topology, the particles are attracted to the solution called local best (*lbest*) among the particles within the local neighborhood topologies.

Kennedy introduced small-world social network into PSO and investigated different population topologies such as circles, wheels, stars and randomly-assigned edges in Ref. [16]. The findings showed that the sociometry of a swarm population significantly affected its ability to find the optimum and the effect was dependent on the problem to be optimized. The populations with fewer connections might perform better on solving multimodal problems and highly interconnected population could perform well on solving unimodal problems [16]. Based on these findings, Kennedy and Mendes introduced heterogeneous population structures, with some subpopulations that are tightly connected and others that are relatively isolated, and studied the effect of *gbest*, *lbest*, star, small, pyramid and von Neumann population topologies by varying the number of neighbors or the number of neighbors in common [68]. The authors conceptualized the influence within the swarm population as information flow between the particles and recommended the PSO researchers to try von Neumann sociometry. The population topologies of *gbest*/all, *lbest*/ring lattice, star, wheel, von Neumann and random are shown in Fig. 4.

Mendes proposed the fully informed particle swarm (FIPS) in which information of all the neighbors are used to guide a particle in the swarm, rather than using only the information of the best particle [69]. The performance of FIPS was tested using five different topologies of *gbest*/all, ring, four clusters, pyramid and square from Ref. [68] and the results showed that FIPS with all sociometries were able to find the optimum [69].

Dynamic topology was firstly proposed by Suganthan in 1999 [19]. In this topology, a small number of particles are selected as the neighbors at the early iterations based on Euclidean distance and then the neighborhood size is dynamically increased over iterations until the particles are fully connected at the end of the iterations. Thus, the search is started with Euclidean distance based (instead of population indices based) *lbest* topology and gradually changed to *gbest* topology by the end of the search [19]. Nasir developed dynamic neighborhood learning PSO (DNLPSO) in which the ring topology was used as the neighborhood topology and the neighbors of each particle were changed after certain intervals [70].

The dynamic neighborhood topology was also proposed in Ref. [71] where the neighbors of each particle were dynamically changed over

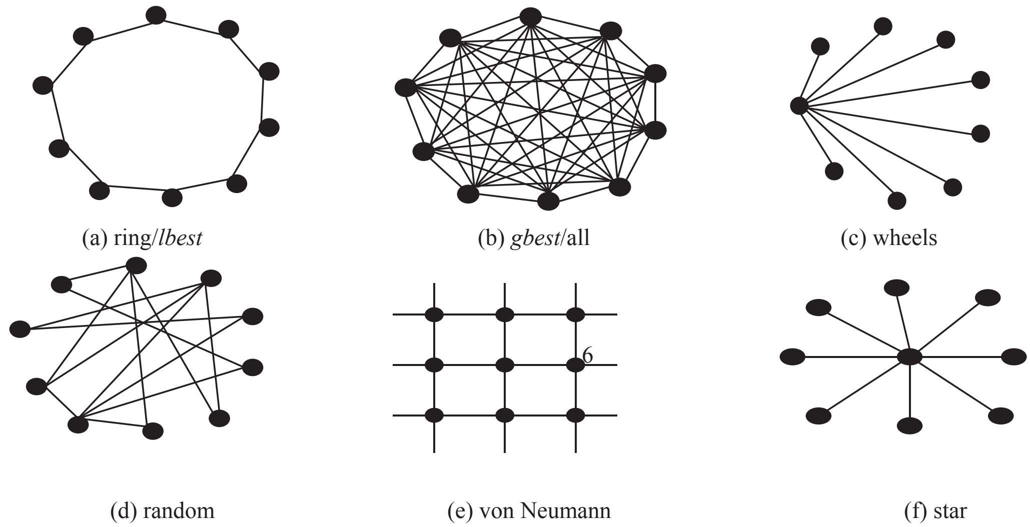

Fig. 4. Population topologies in particle swarms.

time. At each generation, the nearest neighbors in the search space or the function space are selected as neighbors or the neighbors are selected randomly from the population [71]. In Ref. [72], dynamic sociometry was introduced into PSO in which the number of neighbors was randomly increased at any successful iteration to change the swarm search ability from exploration to exploitation. In Ref. [73], for each dimension of the particle, the particles with highest fitness-distance-ratio (FDR) were selected as the neighbors. FDR is the ratio of the difference between the target particle's fitness and the neighbor's fitness to the distance between them in the search space on that dimension [73].

Liu developed a new dynamic topology in which each particle had a small number of connections at the beginning and a number of new connections were gradually added between the particles to improve the exploitation ability [74]. Clerc proposed an autonomous version of PSO called TRIBES which did not require any parameters [75]. In TRIBES, the entire swarm is divided into subswarms and each subswarm has its own order and topology. The population size and topology evolve over time depending on the performance feedback. Good tribes can reduce their population and bad tribes can recruit a new member. The population size is adaptive and the population structure is modified every $L/2$ iteration, where L is the number of links in the population in Ref. [75].

3.1.2. Niche PSO

A niching particle swarm optimizer called NichePSO was proposed by Brits and Engelbrecht in Ref. [76]. In NichePSO, the swarm is divided into subswarms based on the fitness of the individual particles and the subswarms use guaranteed convergence [77] to locate multiple optimal solutions in multimodal optimization problems. The experimental results showed that NichePSO can successfully locate all maxima [76]. Dynamic niching PSO called DNPSO and niching PSO with local search were proposed for multimodal optimization problems in Refs. [20,78], respectively. X. Li suggested that a PSO with ring topology can be used as a niching algorithm without requiring niching parameters in Ref. [79]. Qu et al. proposed a stable niching algorithm called a distance-based locally informed particle swarm (LIPS) optimizer in Ref. [21]. In LIPS, the particles are guided by the local information from the neighborhoods measured in terms of Euclidean distances. In this ways, niches are formed and the local search ability is enhanced without needing to specify any niching parameter [21].

3.1.3. Cellular DE

Noman and Iba proposed a cellular DE (cDE) algorithm and studied the influence of cellular topology on classic DE algorithm [83]. The investigation showed that the cellular topology made DE converge faster and cDE outperformed the classic DE in terms of error value and success ratio [83]. The cellular DE algorithm was applied to solve dynamic optimization problems in Refs. [84,85]. In Ref. [86], the cellular topology was implemented into four different DE algorithms namely canonical DE [34], DEGL [87], JADE [36] and SaDE [88] algorithms to enhance their performance as well as to study the influence of cellular population structure on DE. The new algorithms with cellular population structure were compared with the original ones. The results showed that the cellular versions outperformed the original DE, DEGL and SaDE algorithms, except the original JADE that outperformed the cellular version of the JADE algorithm [86].

Recently Liao proposed cellular direction information based DE algorithm called DE-CDI [89]. In DE-CDI, the cellular topology is defined as a neighborhood topology for each individual in the population and the direction of information flow based on the cellular topology is cooperated into the mutation operation. DE-CDI was employed with several DE variants and evaluated using CEC 2005 benchmark problems. The experimental results demonstrated that cellular directional mutation operation helped to improve the performance of most of the DE algorithms [89].

In the literature, different population topologies were proposed for DE algorithms. Omran et al. investigated the cellular DE models with ring and Von Neumann neighborhood topologies in Refs. [90,91] where the proximity of each individual in the population was defined by the ring and Von Neumann neighborhood topologies. In the ring topology, the individuals are connected in a ring and each individual can only interact with its right and left neighbors. Using the von Neumann topology, the individuals are arranged in a two-dimensional lattice and each individual can interact with its four nearby neighbors located to its right, left, above, and below. DE/*lbest*/1 was used to generate a mutant vector, where *lbest* is the best individual found so far in the neighborhood. The results showed that the performance of DE was improved when using the ring neighborhood topology and decreased when using the von Neumann neighborhood topology [90,91]. Thus, self-adaptive DE with ring neighborhood topology was investigated using SDE/*lbest*/1 mutation mechanism in Ref. [92]. The study showed that self-adaptive DE with ring topology improved the performance of SDE on unimodal benchmark problems.

Based on bare bones PSO, bare bones DE (BBDE) was proposed and investigated with the ring and von Neumann topologies in Ref. [93]. Inspired by PSO's *lbest* and *gbest* topologies [94], another cellular DE model with local and global mutation operators was proposed in the literature [87,95] where neighborhood-best and global best solutions were used to generate a new candidate solution. Ring neighborhood topology was also proposed for local-neighborhood mutation mechanism in adaptive memetic DE algorithm in Ref. [96]. Based on hierarchical PSO, Shi et al. [97] proposed hierarchical DE (HDE) in which the individuals in the population were organized in a tree hierarchy. According to their fitness values, the individuals move up and down in the HDE algorithm. The individuals with better fitness are located at upper/parent level and the parent of the current individual is selected as the base vector in HDE's mutation mechanism. HDE was applied to solve two kinetic model parameter estimation problems and the results showed that HDE was an efficient method for solving problem [97].

In Ref. [98], a proximity-based DE (ProDE) algorithm was proposed where the information of neighboring individuals was used to guide the population towards the global optimum. Learning-enhanced DE (LeDE) algorithm was proposed in Ref. [99] where the clustering-based learning strategies were used to exchange the neighborhood information within the same cluster as well as between different clusters. Neighborhood and direction information based DE (NDi-DE) algorithm was proposed in Ref. [100] where neighborhood guided selection method was used to select the parent individuals for the mutation process and the direction information of best/worst of nearby neighbor was induced into mutation strategy. In Ref. [101], DE/rand-to-neighborhood-best/1 mutation strategy was proposed to improve the performance of DE and different neighborhood structures such as Moore, von Neumann, Cole and Smith were employed to determine the best individual among the neighbors.

3.1.4. Niche DE

The most frequently used niching methods are crowding, speciation, clearing and fitness sharing [102]. Thomsen proposed crowding-based DE (CDE) in Ref. [122] to track and maintain multiple optima during the optimization process. In crowding-based DE, unlike the conventional DE, the offspring replaces the most similar individual among a subset of the population. The author also introduced the fitness sharing concept into DE to locate the multiple optima simultaneously in Ref. [103]. The author evaluated the performance of both crowding-based DE and the sharing DE using several benchmark problems and concluded that crowding-based DE was able to locate as well as maintain the multiple optima and satisfied the criteria for multimodal optimization [103].

Li et al. presented another niching algorithm called species-based DE (SDE) in Ref. [104] where the multiple global optima were located simultaneously through the adaptive formation of multiple species. The performance of SDE was tested using five multimodal functions and the experiment results demonstrated that SDE was computationally more efficient than crowding-based DE algorithm. Epitropakis incorporated the information of the nearest neighbor of the current individual into the mutation operation and proposed a parameter-free niching DE algorithm in Ref. [105]. The author added the external dynamic archive mechanism into the niching DE algorithm and presented a dynamic archive niching DE algorithm in Ref. [106].

Qu et al. [107] proposed a modified fitness sharing DE algorithm called ShDE in which the neighborhood is defined based on Euclidean distance [19] and the mutation is operated within the Euclidean neighborhood. The author incorporated this Euclidean neighborhood concept with different niching methods and presented NCDE, NSDE and NShDE algorithms in Ref. [107]. Hui et al. [108] introduced the ensemble of mutation strategies and parameters into NCDE algorithm and proposed ensemble NCDE algorithm to maintain the population diversity as well as to improve the exploitation ability of the individuals to solve difficult multimodal optimization problems.

Locally informative niching DE (LoINDE) algorithm was introduced in Ref. [109] where distance and estimated gradient information of the adjacent members were used to guide the search process of each individual in the population. Later LoINDE was integrated with crowding, speciation, and fitness sharing methods resulting in LoICDE, LoISDE, LoIShde, respectively. The performances of the three variants were tested using 31 multimodal benchmark problems. The results showed that the three enhanced variants outperformed their original methods namely CDE, SDE and ShDE. However, the overall performance differences between the three variants were not statistically significant. Therefore, only the results of LoISDE algorithm were reported and compared against other algorithms in Ref. [109]. The comparison results showed that overall LoISDE ranked first and outperformed the compared DE and PSO based niching algorithms.

3.1.5. Other cellular EA models

The cellular GA model was developed independently by Tomassini [110] and Whitley [111]. Later, Janson introduced a pyramidal hierarchical topology into cellular GA model and proposed a new hierarchical cellular genetic algorithm (cGA) in Ref. [112]. The cellular topology was introduced into genetic programming as cellular genetic programming (CGP) algorithm [113]. The performance of various cellular EAs were studied and analysed in the literature [7,25,114–116].

3.2. Distributed topology

The distributed topology subdivides the single panmictic population into smaller islands/subpopulations called *demes*. Each island/subpopulation is isolated from each other and evolves independently and simultaneously. The interaction takes place in the form of selected individuals moving or copying between the subpopulations. During migration, one or more individuals are migrated into one or more of the other islands according to a given/predefined migration policy. The islands may exchange the migrants with other islands at a fixed predefined time for all the islands or at independent times. Therefore, in distributed population topology, exploration is promoted while the subpopulations are non-interacting and exploitation is promoted while occasionally migrating between the islands. In this way, independent evolution and migration schemes delay the stagnation of evolution and provide a good balance between the exploration and exploitation of population-based algorithms.

3.2.1. Distributed PSO

A multi-swarm PSO using charged particles (PSO-2S) was developed in Ref. [29] where the search space was partitioned and two kinds of swarms called main and auxiliary were used. In PSO-2S, the auxiliary swarms are initialized in different partitioned areas, using charged particles. After a certain number of generations, the main swarm is formed with the best individual of the auxiliary swarms to continue the search. Liang proposed dynamic multi-swarm PSO (DMS-PSO) [17]. In DMS-PSO, the population is divided into many small-sized sub-swarms and these sub-swarms are regrouped frequently by using various regrouping schedules. The information is exchanged between the sub-swarms and shared within the whole population. DMS-PSO is combined with Quasi-Newton method to improve the local search in Ref. [17]. In Ref. [80], DMS-PSO is combined with a modified multi-trajectory search algorithm and the sub-regional Harmony search resulting in DMS-PSO-SHS to solve large scale global optimization problems. DMS-PSO was also extended to solve multi-objective optimization [81] problems by combining an external archive and a new updating strategy in Ref. [82].

In order to improve the performance of PSO for solving large-scale optimization problems, Fang proposed parallel PSO algorithm with island population model in Ref. [117]. The experimental results showed that parallel PSO algorithm with island population model performed significantly better than the classic PSO algorithm in terms of convergence and solution quality [117]. An agent based parallel PSO algorithm called APPSO was proposed in Ref. [118]. In APPSO, the whole swarm is divided into subswarms and the particles are traded between the subswarms using different trading strategies. The parallel version of multi-objective particle swarm optimization based on decomposition (MOPSO/D) was proposed by Li in Ref. [119]. The proposed multi-objective PSO algorithm adopts the island population model and divides the whole swarm into the several subspecies. Recently, Vlachogiannis proposed a parallel vector evaluated PSO algorithm to determine generator contribution to transmission system in Ref. [120].

Cui and Potok proposed a distributed adaptive PSO algorithm to track the optimal solution in a dynamically changing environment [121]. Jiang presented multi-swarm accelerating PSO (MSA-PSO) algorithm to solve dynamic continuous functions in Ref. [122]. In MSA-PSO, the population is divided into many small subswarms. These subswarms are dynamic and regrouped frequently, as in DMS-PSO [82]. The information is exchanged between the subswarms and accelerating operators is combined to improve its local search ability. The performance of MSA-PSO algorithm is tested on some dynamic continuous functions and the results demonstrated that MSA-PSO recognized the changes in the search space and could adjust to the changes in the dynamic environment. An empirical study of parallel and distributed PSO algorithm was presented in Ref. [123].

3.2.2. Distributed DE

The impact of population topologies on the performance of distributed version of DE algorithm was investigated in Refs. [53–55,124,125]. The distributed DE (dDE) models can be characterized according to the neighborhood topology, the migration policy, the immigrant selection and replacement function. In addition, dDE can be further classified into homogenous and heterogeneous dDE based on the parameter setting of islands. In homogenous dDE, the same parameter setting is used for all the islands whereas different parameter settings are used for different islands in heterogeneous dDE.

Zaharie and Petcu [126] proposed the first version of dDE algorithm in which the population is divided into several islands and adaptive DE algorithm was executed in parallel in each island. The islands are connected in a random topology and the migration process between the islands is carried out by exchanging the individual with a randomly selected individual from a randomly chosen island. The performance of parallel distributed adaptive DE was tested using six benchmark functions and the results demonstrated that the distributed model offered improved convergence performance [126]. The authors presented another distributed DE variant in which the migration process and the DE control parameters were adaptively controlled according to the genotypical diversity criterion [127]. Tasoulis et al. [128] implemented DE in a virtual parallel environment and proposed PARDE algorithm to improve the performance and reduce the computation time. In PARDE, the islands are connected in a unidirectional ring topology and are assigned to different processors of a parallel virtual machine. Kozlov and Samsonov introduced parallel DE with a new migration scheme in which the best member of an island replaced the oldest solution in the target island [129].

Apolloni [130] presented island-based dDE with a new migration scheme in which the individuals to migrate and to be replaced were randomly chosen by the selection function and the replacement function. In the proposed dDE algorithm, a unidirectional ring topology was used to connect the islands and the individuals are exchanged with the nearest neighbor island. Falco et al. [131] proposed a distributed DE with a locally connected topology in which each island has exactly four neighboring subpopulations and the islands were connected in a toroidal mesh. The islands exchange individuals every predefined migration interval. Ishimizu and Tagawa presented another dDE algorithm called a structured DE (StDE) in Ref. [132]. In StDE, the multiple populations are connected with three different network topologies: the ring, the torus and the hypercube. During migration, the best individual from the emitting population is migrated only to an adjacent neighborhood population to replace a randomly selected individual.

Inspired by biological invasion phenomenon, an invasion-based migration model for distributed DE algorithm (IM-dDE) was proposed in Refs. [133,134]. In IM-dDE, the migration process is implemented based on a multistage process and used within the stepping-stone model where interaction takes place only within logically or physically connected islands. The IM-dDE algorithm was evaluated using a set of benchmark functions and the results proved that IM-dDE achieved superior performance compared to dDE algorithm [134]. Recently, the author introduced adaptive control parameter setting to IM-dDE algorithm and proposed AIM-dDE algorithm in Ref. [135]. AIM-dDE algorithm obtained superior performance against over compared adaptive dDE algorithm and IM-dDE algorithms.

Weber et al. [136] proposed a dDE algorithm with a scale factor inheritance scheme where each island had its own unique scale factor value. During migration, not only the best individual but also the scale factor was inherited to the target neighboring population [136]. In Ref. [137], dDE is studied with four different scale factor schemes: random initialization scheme, equally spaced scale factor scheme, adaptive randomized updating scheme and random updating scheme. The study showed that the employment of the scale factor was beneficial and offered significantly improved performance over the original algorithms [137]. Later, the author analysed the interaction of binomial and exponential crossovers with the four different scale factor schemes in Ref. [138]. The analysis showed that the scale factor scheme improved the performance of dDE algorithm with the binomial crossover, but was not much beneficial to the dDE algorithm with the exponential crossover.

Weber et al. presented dDE with exploration and exploitation population families, and was called DDE-EEPF [139]. In DDE-EEPF, the islands are categorized into two families with constant and dynamic subpopulation sizes. The first family with constant population size is responsible for exploration and the second family with dynamic population size is responsible for exploitation. In the second family, the population size is dynamically reduced during the evolution and the best individuals are migrated to the islands from the first family at the later stages of evolution. The DDE-EEPF was evaluated using 500 and 1000 high dimensional test problems and performed better than PARDE, distributed DE and island based DE algorithms [139]. Weber et al. proposed parallel random injection DE (PRIDE) algorithm and shuffle or update parallel DE (SOUPDE) in Refs. [140] and [141], respectively. In PRIDE, the candidate solutions are randomly generated and injected within each subpopulation to prevent premature convergence occurring due to excessively frequent migration [140]. In SOUPDE, two important mechanisms were introduced into parallel DE algorithm: shuffling the individuals of the subpopulations and updating the scale factors values of each subpopulation randomly. SOUPDE activated the two mechanisms periodically to balance the exploration and exploitation [141].

The distributed DE (dDE) is enhanced with multicultural migration and proposed DDEM algorithm for global numerical optimization in Ref. [142]. To enhance and maintain the population diversity in the subpopulations, DDEM was integrated with two migration schemes: target individual-based migration and representative individual-based migration. The DDEM algorithm was tested using 34 scalable test functions and resulted in better or equal performance compared to other dDE algorithms with classical migration policy [142]. The migration policy, migration size, migration interval and migration topologies of distributed DE were studied and analysed in the literature [55,143–145].

Asynchronous island based parallel DE algorithm (asynPDE) was proposed in Ref. [146] to solve complex problems in computational systems biology. In asynPDE, global search is implemented through asynchronous parallel DE and local search is improved using the heuristics of local solver (tabu list and logarithmic search) to balance exploration and exploitation capabilities. A heterogeneous asynchronous island DE was presented in Ref. [147] where five islands with five different mutation strategies were considered and compared with dDE integrated with the same mutation strategy in all islands. The heterogeneous model performed as well as other dDE algorithms, but did not result in outstanding performance [147].

3.2.3. Other island/distributed EA models

Whitley proposed the island model genetic algorithm in the genetic algorithm tutorial in 1994 [111]. Later the author investigated the separability, population size and convergence characteristics of island model GA in Ref. [148]. Lardeux presented dynamic island-based GA for solving the combinatorial optimization problems in Ref. [149]. Parallel and distributed EAs were studied in the literature [7,24,115,150,151] and their island topologies were also investigated [24,152].

4. Hierarchical, random, irregular and coevolution models

4.1. Hierarchical/hybrid model

The hierarchical model combines either the cellular or island models or two (or more) distributed models hierarchically and shares the properties of different EA models. Zaharie introduced the hierarchical approach to distributed EAs for multi-objective optimization in Ref. [153].

Janson and Middendorf proposed a hierarchical version of PSO (H-PSO) in which dynamic hierarchy was used to define a neighborhood structure [154]. In H-PSO, all particles were connected using a hierarchical topology in which each node has exactly one particle. Depending on the quality of their best solutions found so far, the particles either move up or down the hierarchy so that the best particle has the highest influence on the swarm [154]. H-PSO was used to solve the dynamic optimization problems in Ref. [155] and its adaptive variants were presented in Ref. [156]. A new variant of hierarchical PSO called hierarchical dynamic local neighborhood PSO (H-DLPSO) was proposed in Ref. [157]. In H-DLPSO, each level of hierarchy is composed of many subswarms of the particles and the information is shared between the subswarms at the same level of the hierarchy. The subswarms are randomly regrouped in each generation and dynamically varied according to H-PSO version [157].

The hierarchical topology was introduced into parallel DE in Ref. [158] and cellular DE in Ref. [53]. Herrera proposed hierarchical distributed genetic algorithm in Ref. [159]. Folino and Spezzano developed hierarchical distributed genetic programming in which the island model was adopted in the upper level and the cellular model was used in the lower level of the hierarchy [160].

4.2. Random and irregular model

Clerc firstly proposed random topology in Ref. [161] and the random topologies were studied in details and introduced to dynamic PSO in Ref. [162]. Clerc also introduced adaptive topology in Ref. [163]. PSO with dynamic random population topology was proposed in Ref. [164] where various random population topologies were generated and used during the evolution.

The small-world topology was introduced into PSO in Refs. [16,165,166]. Inspired by random and small-world structured population [167], random and small-world topology DE algorithms were proposed in Ref. [53]. Zu proposed PSO with Watts-Strogatz Model in Ref. [168]. The effects of Watts–Strogatz and Albert–Barabási models on cellular EAs were investigated in Ref. [169].

4.3. Coevolution models

Coevolution models can be divided into two classes: cooperative coevolution (dimension-distributed model) and competitive coevolution (population-distributed model) [5]. The modelling and convergence characteristics of distributed co-evolutionary algorithms were studied in Ref. [170]. Cooperative co-evolutionary architecture was introduced into PSO and DE algorithms in the literature.

In Ref. [171], a cooperative approach was applied to PSO (CPSO-Sk) in which the search space dimensionality was split and different swarms were used to search over different dimensions of a solution cooperatively. Knowledge-based cooperative PSO (KCPSO) [172] introduced using a knowledge billboard to record a variety of search information and used a multi-swarm to maintain the population diversity. The local search is carried out simultaneously in different regions of the search space by each sub-swarm and the global search is carried out using a cooperative evolution among the sub-swarms [172]. The cooperative as well as competitive coevolution architectures were integrated into PSO algorithm in Ref. [173].

The cooperative co-evolutionary approach was firstly introduced into DE algorithm in Refs. [174–176]. In this paper, the solution vector of high dimensional problems is split into smaller vectors using the classical cooperative co-evolution approach [177] and they are separately solved by DE algorithms to co-evolve for the resolution of the big problem. The two more co-evolutionary DE algorithms are proposed for solving constrained optimization problems in Refs. [175,176].

5. Discussion

The population topology represents the interrelationship between the individuals in the population influencing the performance of population-based algorithms. The panmictic population is the simplest model and the population has no particular structure at all, i.e., during the evolutionary process, all individuals can interact with any other individual in the population and the mate selection is based on the entire population. This accelerates the flow of information between individuals, consequently may lead to premature convergence. To address this problem, spatially structured populations are proposed in which individuals have their own neighborhood and only those that are in the same neighborhood can interact. In this way, the flow of information is slowed down within the population and the algorithm has higher exploration ability compared to the panmictic population. The two main spatial structured populations are cellular/diffusion (fine-grained) and distributed (coarse-grained) models.

The population topology interacts with the problem to be optimized as well as the algorithm. Hence, some population topologies performed well on some classes of problems while other population structure yields better performance on other classes of optimization problems. For example, panmictic topology is suitable for unimodal problems while multi-modal problems require sub-population based topologies. However, it is not straightforward to deterministically state which topology is the best for a given problem. Therefore, different variants of population topologies have been proposed for the population-based optimization algorithms. As mentioned in Introduction Section, currently the PSO and DE algorithms are the most frequently used algorithms among the population-based optimization algorithms. Therefore, we focus on the cellular and distributed models proposed for PSO and DE algorithms in this paper.

With the panmictic population, it is difficult for population based algorithm to escape from local optima, especially when the problem becomes complex and its dimensionality increases. Moreover, due to its rapid flow of information, the algorithms may also suffer diversity loss when adapting to dynamic environments. In cellular topology, individuals are connected in a unique coordinate grid and mate only within the specified neighborhood. By introducing the cellular topology to PSO and DE algorithms, the cellular population promotes the slow diffusion of information through the grid thereby maintaining the population diversity for more generations or iterations. Therefore, it can be seen in the literature [61,65,84,85,89] that PSO and DE with cellular topology showed improved performance over original PSO and DE when solving complex high dimensional problems as well as dynamic optimization problems.

In cellular (fine-grained) model, the in-breeding within demes tends to cause speciation as clusters of related solutions develop, leading to the niching behavior [56]. Niching methods can increase diversity and locating multiple solutions to optimization problems. In niching algorithms, subpopulations are formed within a population and each subpopulation searches for optimal solutions in different regions of the search space. Thus, niching methods are regarded as strong diversification methods [178]. Niching PSO and DE algorithms showed promising performance on multimodal optimization problems.

For other population topologies (such as *lbest*, wheels, von Neumann, etc.), the population with fewer connection may perform better on multimodal problems while highly interconnected populations may yield better performance on unimodal problems [16].

In distributed topology, a large panmictic population is partitioned into several small subpopulations (islands). Each island/subpopulation is isolated from each other and evolves independently and simultaneously. In the island model approach, information sharing platform (migration method) is very important and the migration policies govern the performance of an algorithm. Compared to the original PSO and DE algorithms with panmictic single population, the use of distributed structured population in PSO and DE improves the performance on large-scale optimization problems as well as real-world applications [55,121,123,139,145].

6. Conclusion

Most population structures and topologies were developed for PSO and DE algorithms. Therefore, this paper presents population topologies in general and provides a comprehensive literature review of the population topologies proposed for PSO and DE. In this paper, the population topologies have been categorized and presented as representative examples of current topologies so that researchers can develop novel topologies and apply population topologies to improve performance of other population-based algorithms.

Future research should apply more extensively the population topologies to other problem classes such as constrained, multi-objective, large scale, bi-level and so on. Current research is primarily restricted to bound constrained single objective optimization. With an exception of DE and PSO, population topologies are not frequently used in other population-based optimizers. Hence, other search methods should also integrate topologies to improve their performances. Another promising research direction is to integrate ensemble strategies [40,107,179,180] with population topologies. Currently, ensemble methods and topologies are mostly investigated separately. The advantage of this integration, in particular with Euclidean distance-based neighborhood topology [19–21] or heterogeneous topology [181], is that explorative ensemble strategies can be used for poor solutions while exploitative strategies can be used for the best solutions of the population [181,182].

References

[1] K.A. De Jong, Evolutionary Computation: a Unified Approach, MIT Press, 2006.
[2] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Springer Berlin Heidelberg, 2015.
[3] T. Back, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford University Press, 1996.
[4] C. Darwin, On the Origin of Species by Means of Natural Selection: or the Preservation of Favoured Races in the Struggle for Life, Watts, 1859.
[5] M. Tomassini, Spatially Structured Evolutionary Algorithms: Artificial Evolution in Space and Time, Springer Berlin Heidelberg, 2010.
[6] J.L. Payne, M. Giacobini, J.H. Moore, Complex and dynamic population structures: synthesis, open questions, and future directions, Soft Comput. 17 (2013) 1109–1120.
[7] E. Alba, M. Tomassini, Parallelism and evolutionary algorithms, IEEE Trans. Evol. Comput. 6 (2002) 443–462.
[8] E. Cantú-Paz, Efficient and Accurate Parallel Genetic Algorithms, Springer US, 2012.
[9] M.Z. Ali, P.N. Suganthan, R.G. Reynolds, A.F. Al-Badarneh, Leveraged neighborhood restructuring in cultural algorithms for solving real-world numerical optimization problems, IEEE Trans. Evol. Comput. 20 (2016) 218–231.
[10] M. Gorges-Schleuter, Explicit parallelism of genetic algorithms through population structures, in: H.-P. Schwefel, R. Männer (Eds.), Parallel Problem Solving from Nature: 1st Workshop, PPSN I Dortmund, FRG, October 1–3, 1990 Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 1991, pp. 150–159.
[11] M.G. Norman, A Genetic Approach to Topology Optimization for Multiprocessor Architectures, University of Edinburgh, 1988. Technical Report.
[12] E. Alba, J.M. Troya, Cellular evolutionary algorithms: evaluating the influence of ratio, in: M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J. Merelo, et al. (Eds.), Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, 2000 Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, pp. 29–38.
[13] E. Alba, B. Dorronsoro, Cellular Genetic Algorithms, Springer US, 2010.
[14] H. Mühlenbein, M. Schomisch, J. Born, The parallel genetic algorithm as function optimizer, Parallel Comput. 17 (1991) 619–632.
[15] W.N. Martin, J. Lienig, J.P. Cohoon, Island (migration) models: evolutionary algorithms based on punctuated equilibria, in: Handbook of Evolutionary Computation, Oxford University Press, 1999.
[16] J. Kennedy, Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance, in: Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress, vol. 3, 1999, pp. 1–1938.
[17] J.J. Liang, P.N. Suganthan, Dynamic multi-swarm particle swarm optimizer, in: Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005, 2005, pp. 124–129.
[18] G. Wu, R. Mallipeddi, P.N. Suganthan, R. Wang, H. Chen, Differential evolution with multi-population based ensemble of mutation strategies, Inf. Sci. 329 (2016) 329–345, 02/01/2016.
[19] P.N. Suganthan, Particle swarm optimiser with neighbourhood operator, in: Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress vol. 3, 1999, pp. 1–1962.
[20] B.Y. Qu, J.J. Liang, P.N. Suganthan, Niching particle swarm optimization with local search for multi-modal optimization, Inf. Sci. 197 (8/15/2012) 131–143.
[21] B.Y. Qu, P.N. Suganthan, S. Das, A distance-based locally informed particle swarm model for multimodal optimization, IEEE Trans. Evol. Comput. 17 (2013) 387–402.
[22] S.-Z. Zhao, P.N. Suganthan, S. Das, Self-adaptive differential evolution with multi-trajectory search for large-scale optimization, Soft Comput. 15 (2011) 2175–2185.
[23] H. Li, Q. Zhang, Multiobjective optimization problems with complicated pareto sets, MOEA/d and NSGA-II, IEEE Trans. Evol. Comput. 13 (2009) 284–302.
[24] Y.-J. Gong, W.-N. Chen, Z.-H. Zhan, J. Zhang, Y. Li, Q. Zhang, et al., Distributed evolutionary algorithms and their models: a survey of the state-of-the-art, Appl. Soft Comput. 34 (9/2015) 286–300.
[25] E. Alba, B. Dorronsoro, M. Giacobini, M. Tomassini, Decentralized cellular evolutionary algorithms, in: Handbook of Bioinspired Algorithms and Applications, vol. 7, 2005, pp. 103–120.
[26] E. Alba, G. Luque, S. Nesmachnow, Parallel metaheuristics: recent advances and new trends, Int. Trans. Oper. Res. 20 (2013) 1–48.
[27] N. Nedjah, E. Alba, L. De Macedo Mourelle, Parallel Evolutionary Computations, Springer, 2006.
[28] S. Mo, J. Zeng, Y. Tan, Particle swarm optimisation based on self-organisation topology driven by different fitness rank, Int. J. Comput. Sci. Eng. 6 (2011) 24–33.
[29] A. El Dor, M. Clerc, P. Siarry, A multi-swarm PSO using charged particles in a partitioned search space for continuous optimization, Comput. Optim. Appl. 53 (2012) 271–295.
[30] J. Kennedy, R. Eberhart, Particle swarm optimization, in: In Neural Networks, 1995. Proceedings., IEEE International Conference, 1995, pp. 1942–1948.
[31] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., the 1998 IEEE International Conference, 1998, pp. 69–73.
[32] R. Cheng, Y. Jin, A competitive swarm optimizer for large scale optimization, IEEE Trans. Cybern. 45 (2015) 191–204.
[33] M. Campos, R.A. Krohling, I. Enriquez, Bare bones particle swarm optimization with scale matrix adaptation, IEEE Trans. Cybern. 44 (2014) 1567–1578.
[34] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (1997) 341–359.
[35] K. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: a Practical Approach to Global Optimization, Springer Berlin Heidelberg, 2014.
[36] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans. Evol. Comput. 13 (2009) 945–958.
[37] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2009) 398–417.
[38] S.M. Islam, S. Das, S. Ghosh, S. Roy, P.N. Suganthan, An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization, IEEE Trans. Syst. Man Cybern. Part B Cybern. 42 (2012) 482–500.
[39] M.Z. Ali, N.H. Awad, P.N. Suganthan, Multi-population differential evolution with balanced ensemble of mutation strategies for large-scale global optimization, Appl. Soft Comput. 33 (8/2015) 304–327.
[40] R. Mallipeddi, P.N. Suganthan, Q.K. Pan, M.F. Tasgetiren, Differential evolution algorithm with ensemble of parameters and mutation strategies, Appl. Soft Comput. 11 (3/2011) 1679–1696.
[41] A. Glotić, A. Zamuda, Short-term combined economic and emission hydrothermal optimization by surrogate differential evolution, Appl. Energy 141 (3/1/2015) 42–56.
[42] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (2011) 4–31.
[43] F. Neri, V. Tirronen, Recent advances in differential evolution: a survey and experimental analysis, Artif. Intell. Rev. 33 (2010) 61–106.
[44] S. Das, S.S. Mullick, P.N. Suganthan, Recent advances in differential evolution – an updated survey, Swarm Evol. Comput. 27 (4/2016) 1–30.
[45] P.N. Suganthan, Differential evolution algorithm: recent advances, in: Theory and Practice of Natural Computing, Springer, 2012, pp. 30–46.
[46] R. Mendes, Population Topologies and Their Influence in Particle Swarm Performance, Citeseer, 2004.
[47] T. Tsujimoto, T. Shindo, T. Kimura, K. Jin'no, A relationship between network topology and search performance of PSO, in: 2012 IEEE Congress on Evolutionary Computation, 2012, pp. 1–6.

[48] A.J.R. Medina, G.T. Pulido, J.G. Ramírez-Torres, A comparative study of neighborhood topologies for particle swarm optimizers, in: International Conference on Evolutionary Computation, 2009, pp. 152–159.
[49] D. Schor, W. Kinsner, J. Anderson, A study of optimal topologies in swarm intelligence, in: Electrical and Computer Engineering (CCECE), 2010 23rd Canadian Conference, 2010, pp. 1–8.
[50] R. Mendes, J. Kennedy, J. Neves, Watch thy neighbor or how the swarm can learn from its environment, in: In Swarm Intelligence Symposium, 2003. SIS\'03. Proceedings of the 2003 IEEE, 2003, pp. 88–94.
[51] R. Mendes, J. Kennedy, Avoiding the pitfalls of local optima: how topologies can save the day, in: In Proceedings of the 12th Conference Intelligent Systems Application to Power Systems (ISAP), 2003.
[52] J. Kennedy, Probability and dynamics in the particle swarm, in: Evolutionary Computation, 2004. CEC2004. Congress, vol. 1, 2004, pp. 340–347.
[53] B. Dorronsoro, P. Bouvry, Improving classical and decentralized differential evolution with new mutation operator and population topologies, IEEE Trans. Evol. Comput. 15 (2011) 67–98.
[54] R. Storn, Differential evolution research–trends and open questions, in: Advances in Differential Evolution, Springer, 2008, pp. 1–31.
[55] I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, E. Tarantino, Impact of the topology on the performance of distributed differential evolution, in: I.A. Esparcia-Alcázar, M.A. Mora (Eds.), Applications of Evolutionary Computation: 17th European Conference, EvoApplications 2014, Granada, Spain, April 23-25, 2014, Revised Selected Papers, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 75–85.
[56] Y. Davidor, A naturally occurring niche and species phenomenon: the model and first results, in: In Proceedings of 4th International Conference on Genetic Algorithms, 1991, pp. 257–263.
[57] K. Deb, D.E. Goldberg, An investigation of niche and species formation in genetic function optimization, in: Proceedings of the 3rd International Conference on Genetic Algorithms, 1989, pp. 42–50.
[58] M. Waintraub, C. Pereira, R. Schirru, The cellular particle swarm optimization algorithm, in: International Nuclear Atlantic Conference, Santos, SP, Brazil, 2007.
[59] A.B. Hashemi, M.R. Meybodi, Cellular PSO: a PSO for dynamic environments, in: Z. Cai, Z. Li, Z. Kang, Y. Liu (Eds.), Advances in Computation and Intelligence: 4th International Symposium, ISICA 2009 Huangshi, China, Ocotober 23-25, 2009 Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 422–433.
[60] T. Blackwell, J. Branke, Multiswarms, exclusion, and anti-convergence in dynamic environments, IEEE Trans. Evol. Comput. 10 (2006) 459–472.
[61] S. Nabizadeh, A. Rezvanian, M.R. Meybodi, A multi-swarm cellular PSO based on clonal selection algorithm in dynamic environments, in: Informatics, Electronics & Vision (ICIEV), 2012 International Conference, 2012, pp. 482–486.
[62] A. Sharifi, V. Noroozi, M. Bashiri, A.B. Hashemi, M.R. Meybodi, Two phased cellular PSO: a new collaborative cellular algorithm for optimization in dynamic environments, in: 2012 IEEE Congress on Evolutionary Computation, 2012, pp. 1–8.
[63] Y. Shi, H. Liu, L. Gao, G. Zhang, Cellular particle swarm optimization, Inf. Sci. 181 (10/15/2011) 4460–4493.
[64] L. Gao, J. Huang, X. Li, An effective cellular particle swarm optimization for parameters optimization of a multi-pass milling process, Appl. Soft Comput. 12 (11/2012) 3490–3499.
[65] W. Fang, J. Sun, H. Chen, X. Wu, A decentralized quantum-inspired particle swarm optimization algorithm with cellular structured population, Inf. Sci. 330 (2/10/2016) 19–48.
[66] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium, 1995, pp. 39–43.
[67] R. Eberhart, P. Simpson, R. Dobbins, Computational Intelligence PC Tools, Academic Press Professional, Inc., Boston, 1996.
[68] J. Kennedy, R. Mendes, Population structure and particle swarm performance, in: Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress, 2002, pp. 1671–1676.
[69] R. Mendes, J. Kennedy, J. Neves, The fully informed particle swarm: simpler, maybe better, IEEE Trans. Evol. Comput. 8 (2004) 204–210.
[70] M. Nasir, S. Das, D. Maity, S. Sengupta, U. Halder, P.N. Suganthan, A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization, Inf. Sci. 209 (2012) 16–36.
[71] S.B. Akat, V. Gazi, Particle swarm optimization with dynamic neighborhood topology: three neighborhood strategies and preliminary results, in: Swarm Intelligence Symposium, 2008. SIS 2008, IEEE, 2008, pp. 1–8.
[72] M. Richards, D.A. Ventura, Dynamic sociometry in particle swarm optimization, in: Presented at the in Joint Conference on Information Sciences, 2003.
[73] T. Peram, K. Veeramachaneni, C.K. Mohan, Fitness-distance-ratio based particle swarm optimization, in: Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE, 2003, pp. 174–181.
[74] H. Liu, E. Howely, J. Duggan, Particle swarm optimisation with gradually increasing directed neighbourhoods, in: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, 2011, pp. 29–36.
[75] M. Clerc, Particle Swarm Optimization, Wiley, 2013.
[76] R. Brits, A.P. Engelbrecht, F. Van den Bergh, A niching particle swarm optimizer, in: Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, 2002, pp. 692–696.
[77] F. v. d. Bergh, A.P. Engelbrecht, A new locally convergent particle swarm optimiser, in: Systems, Man and Cybernetics, 2002 IEEE International Conference, vol. 3, 2002, p. 6.
[78] A. Nickabadi, M.M. Ebadzadeh, R. Safabakhsh, DNPSO: a dynamic niching particle swarm optimizer for multi-modal optimization, in: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), 2008, pp. 26–32.
[79] X. Li, Niching without niching parameters: particle swarm optimization using a ring topology, IEEE Trans. Evol. Comput. 14 (2010) 150–169.
[80] S.Z. Zhao, J.J. Liang, P.N. Suganthan, M.F. Tasgetiren, Dynamic multi-swarm particle swarm optimizer with local search for Large Scale Global Optimization, in: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), 2008, pp. 3845–3852.
[81] J.J. Liang, B.Y. Qu, P.N. Suganthan, B. Niu, Dynamic multi-swarm particle swarm optimization for multi-objective optimization problems, in: 2012 IEEE Congress on Evolutionary Computation, 2012, pp. 1–8.
[82] S.Z. Zhao, P.N. Suganthan, S. Das, Dynamic multi-swarm particle swarm optimizer with sub-regional harmony search, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–8.
[83] N. Noman, H. Iba, Cellular differential evolution algorithm, in: J. Li (Ed.), AI 2010: Advances in Artificial Intelligence: 23rd Australasian Joint Conference, Adelaide, Australia, December 7-10, 2010. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 293–302.
[84] N. Noman, H. Iba, Solving dynamic economic dispatch problems using cellular differential evolution, in: 2011 IEEE Congress of Evolutionary Computation (CEC), 2011, pp. 2633–2640.
[85] V. Noroozi, A.B. Hashemi, M.R. Meybodi, CellularDE: a cellular based differential evolution for dynamic optimization problems, in: A. Dobnikar, U. Lotrič, B. Šter (Eds.), Adaptive and Natural Computing Algorithms: 10th International Conference, ICANNGA 2011, Ljubljana, Slovenia, April 14-16, 2011, Proceedings, Part I, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 340–349.
[86] B. Dorronsoro, P. Bouvry, Differential evolution algorithms with cellular populations, in: R. Schaefer, C. Cotta, J. Kołodziej, G. Rudolph (Eds.), Parallel Problem Solving from Nature, PPSN XI 11th International Conference, Kraków, Poland, September 11-15, 2010, Proceedings, Part II, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 320–330.
[87] S. Das, A. Abraham, U.K. Chakraborty, A. Konar, "Differential evolution using a neighborhood-based mutation operator, IEEE Trans. Evol. Comput. 13 (2009) 526–553.
[88] A.K. Qin, P.N. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, in: 2005 IEEE Congress on Evolutionary Computation, vol. 2, 2005, pp. 1785–1791.
[89] J. Liao, Y. Cai, T. Wang, H. Tian, Y. Chen, Cellular direction information based differential evolution for numerical optimization: an empirical study, Soft Comput. (2015) 1–27.
[90] M. Omran, A. Engelbrecht, A. Salman, Using neighborhood topologies with differential evolution, in: Proc. Int. Workshop Conf. Comput. Intell. Security, 2005.
[91] M. Omran, A. Engelbrecht, A. Salman, Empirical analysis of using neighborhood topologies with differential evolution, Comput. Optim. Appl. 1 (3) (2005) 189–222.
[92] M.G.H. Omran, A.P. Engelbrecht, A. Salman, Using the ring neighborhood topology with self-adaptive differential evolution, in: L. Jiao, L. Wang, X.-b. Gao, J. Liu, F. Wu (Eds.), Advances in Natural Computation: Second International Conference, ICNC 2006, Xi'an, China, September 24-28, 2006. Proceedings, Part I, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 976–979.
[93] M.G. Omran, A.P. Engelbrecht, A. Salman, Bare bones differential evolution, Eur. J. Oper. Res. 196 (2009) 128–139.
[94] K.E. Parsopoulos, M.N. Vrahatis, Unified particle swarm optimization in dynamic environments, in: F. Rothlauf, J. Branke, S. Cagnoni, D.W. Corne, R. Drechsler, Y. Jin, et al. (Eds.), Applications of Evolutionary Computing: EvoWorkkshops 2005: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, and EvoSTOC Lausanne, Switzerland, March 30 - April 1, 2005 Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 590–599.
[95] U.K. Chakraborty, S. Das, A. Konar, Differential evolution with local neighborhood, in: 2006 IEEE International Conference on Evolutionary Computation, 2006, pp. 2042–2049.
[96] A.P. Piotrowski, Adaptive memetic differential evolution with global and local neighborhood-based mutation operators, Inf. Sci. 241 (2013) 164–194.
[97] Y. Shi, X. Zhong, Hierarchical differential evolution for parameter estimation in chemical kinetics, in: PRICAI 2008: Trends in Artificial Intelligence, Springer, 2008, pp. 870–879.
[98] M.G. Epitropakis, D.K. Tasoulis, N.G. Pavlidis, V.P. Plagianakos, M.N. Vrahatis, "Enhancing differential evolution utilizing proximity-based mutation operators, Evol. Comput. IEEE Trans. 15 (2011) 99–119.
[99] Y. Cai, J. Wang, J. Yin, Learning-enhanced differential evolution for numerical optimization, Soft Comput. 16 (2012) 303–330.
[100] Y. Cai, J. Wang, Differential evolution with neighborhood and direction information for numerical optimization, IEEE Trans. Cybern. 43 (2013) 2202–2215.
[101] J.K. Kordestani, A. Ahmadi, M.R. Meybodi, An improved Differential Evolution algorithm using learning automata and population topologies, Appl. Intell. 41 (2014) 1150–1169.
[102] O.M. Shir, Niching in evolutionary algorithms, in: G. Rozenberg, T. Bäck, J.N. Kok (Eds.), Handbook of Natural Computing, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 1035–1069.
[103] R. Thomsen, Multimodal optimization using crowding-based differential evolution, in: Evolutionary Computation, 2004. CEC2004. Congress vol. 2, 2004, pp. 1382–1389.

[104] X. Li, Efficient differential evolution using speciation for multimodal function optimization, in: Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, 2005, pp. 873–880.
[105] M.G. Epitropakis, V.P. Plagianakos, M.N. Vrahatis, Finding multiple global optima exploiting differential evolution's niching capability, in: Differential Evolution (SDE), 2011 IEEE Symposium, 2011, pp. 1–8.
[106] M.G. Epitropakis, X. Li, E.K. Burke, A dynamic archive niching differential evolution algorithm for multimodal optimization, in: 2013 IEEE Congress on Evolutionary Computation,, 2013, pp. 79–86.
[107] B.Y. Qu, P.N. Suganthan, J.J. Liang, Differential evolution with neighborhood mutation for multimodal optimization, IEEE Trans. Evol. Comput. 16 (2012) 601–614.
[108] S. Hui, P.N. Suganthan, Ensemble crowding differential evolution with neighborhood mutation for multimodal optimization, in: Differential Evolution (SDE), 2013 IEEE Symposium, 2013, pp. 135–142.
[109] S. Biswas, S. Kundu, S. Das, Inducing niching behavior in differential evolution through local information sharing, IEEE Trans. Evol. Comput. 19 (2015) 246–263.
[110] M. Tomassini, The parallel genetic cellular automata: application to global function optimization, in: R.F. Albrecht, C.R. Reeves, N.C. Steele (Eds.), Artificial Neural Nets and Genetic Algorithms: Proceedings of the International Conference in Innsbruck, Austria, 1993, Springer Vienna, Vienna, 1993, pp. 385–391.
[111] D. Whitley, A genetic algorithm tutorial, Stat. Comput. 4 (1994) 65–85.
[112] S. Janson, E. Alba, B. Dorronsoro, M. Middendorf, Hierarchical cellular genetic algorithm, in: J. Gottlieb, G.R. Raidl (Eds.), Evolutionary Computation in Combinatorial Optimization: 6th European Conference, EvoCOP 2006, Budapest, Hungary, April 10-12, 2006. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 111–122.
[113] G. Folino, C. Pizzuti, G. Spezzano, A cellular genetic programming approach to classification, in: GECCO, 1999, pp. 1015–1020.
[114] J. Stender, Parallel Genetic Algorithms: Theory and Applications, IOS Press, 1993.
[115] E. Alba, J.M. Troya, A survey of parallel distributed genetic algorithms, Complexity 4 (1999) 31–52.
[116] E. Alba, F. Luna, A.J. Nebro, Advances in parallel heterogeneous genetic algorithms for continuous optimization, Int. J. Appl. Math. Comput. Sci. 14 (2004) 317–334.
[117] F. Huang, X. Fang, Parallel particle swarm optimization algorithm with island population model, Control Decis. 21 (2006) 175.
[118] Y. Lorion, T. Bogon, I.J. Timm, O. Drobnik, An agent based parallel particle swarm optimization - APPSO, in: Swarm Intelligence Symposium, 2009. SIS '09, IEEE, 2009, pp. 52–59.
[119] J.Z. Li, W.N. Chen, J. Zhang, Z.H. Zhan, A parallel implementation of multiobjective particle swarm optimization algorithm based on decomposition, in: Computational Intelligence, 2015 IEEE Symposium Series, 2015, pp. 1310–1317.
[120] J.G. Vlachogiannis, K.Y. Lee, Determining generator contributions to transmission system using parallel vector evaluated particle swarm optimization, IEEE Trans. Power Syst. 20 (2005) 1765–1774.
[121] X. Cui, T.E. Potok, Distributed adaptive particle swarm optimizer in dynamic environment, in: 2007 IEEE International Parallel and Distributed Processing Symposium, 2007, pp. 1–7.
[122] Y. Jiang, W. Huang, L. Chen, Applying multi-swarm accelerating particle swarm optimization to dynamic continuous functions, in: Knowledge Discovery and Data Mining, 2009. WKDD 2009. Second International Workshop, 2009, pp. 710–713.
[123] L. Vanneschi, D. Codecasa, G. Mauri, An empirical study of parallel and distributed particle swarm optimization, in: F. Fernández de Vega, I.J. Hidalgo Pérez, J. Lanchares (Eds.), Parallel Architectures and Bioinspired Algorithms, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 125–150.
[124] J. Apolloni, J. García-Nieto, E. Alba, G. Leguizamón, Empirical evaluation of distributed Differential Evolution on standard benchmarks, Appl. Math. Comput. 236 (2014) 351–366.
[125] E. Alba, G. Luque, Theoretical models of selection pressure for dEAs: topology influence, in: 2005 IEEE Congress on Evolutionary Computation, vol. 1, 2005, pp. 214–221.
[126] D. Zaharie, D. Petcu, Parallel implementation of multi-population differential evolution, Concurr. Inf. Process. Comput. 195 (2005) 223.
[127] D. Zaharie, Control of population diversity and adaptation in differential evolution algorithms, in: Proc. Of MENDEL, 2003, pp. 41–46.
[128] D.K. Tasoulis, N.G. Pavlidis, V.P. Plagianakos, M.N. Vrahatis, Parallel differential evolution, in: Evolutionary Computation, 2004. CEC2004. Congress, vol. 2, 2004, pp. 2023–2029.
[129] K. Kozlov, A. Samsonov, New migration scheme for parallel differential evolution, in: Proceedings of the International Conference on Bioinformatics of Genome Regulation and Structure, 2006, pp. 141–144.
[130] J. Apolloni, E. Alba, Island based distributed differential evolution: an experimental study on hybrid testbeds, in: Eighth International Conference on Hybrid Intelligent Systems, 2008, pp. 696–701.
[131] I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, E. Tarantino, Satellite image registration by distributed differential evolution, in: Applications of Evolutionary Computing, Springer, 2007, pp. 251–260.
[132] T. Ishimizu, K. Tagawa, A structured differential evolution for various network topologies, Int. J. Comput. Commun. 4 (2010) 2–8.
[133] I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, E. Tarantino, A model based on biological invasions for island evolutionary algorithms, Artif. Evol. (2011) 157–168.
[134] I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, E. Tarantino, "Biological invasion–inspired migration in distributed evolutionary algorithms, Inf. Sci. 207 (2012) 50–65.
[135] I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, E. Tarantino, "An adaptive invasion-based model for distributed differential evolution, Inf. Sci. 278 (2014) 653–672.
[136] M. Weber, V. Tirronen, F. Neri, Scale factor inheritance mechanism in distributed differential evolution, Soft Comput. 14 (2010) 1187–1207.
[137] M. Weber, F. Neri, V. Tirronen, A study on scale factor in distributed differential evolution, Inf. Sci. 181 (2011) 2488–2511.
[138] M. Weber, F. Neri, V. Tirronen, A study on scale factor/crossover interaction in distributed differential evolution, Artif. Intell. Rev. 39 (2013) 195–224.
[139] M. Weber, F. Neri, V. Tirronen, Distributed differential evolution with explorative–exploitative population families, Genet. Program. Evolvable Mach. 10 (2009) 343–371.
[140] M. Weber, F. Neri, V. Tirronen, Parallel random injection differential evolution, in: Applications of Evolutionary Computation, Springer, 2010, pp. 471–480.
[141] M. Weber, F. Neri, V. Tirronen, Shuffle or update parallel differential evolution for large-scale optimization, Soft Comput. 15 (2011) 2089–2107.
[142] J. Cheng, G. Zhang, F. Neri, Enhancing distributed differential evolution with multicultural migration for global numerical optimization, Inf. Sci. 247 (2013) 72–93.
[143] J. Lässig, D. Sudholt, Design and analysis of migration in parallel evolutionary algorithms, Soft Comput. 17 (2013) 1121–1144.
[144] Z. Skolicki, K. De Jong, The influence of migration sizes and intervals on island models, in: Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, 2005, pp. 1295–1302.
[145] G. Jeyakumar, C.S. Velayutham, Empirical study on migration topologies and migration policies for island based distributed differential evolution variants, in: Swarm, Evolutionary, and Memetic Computing, Springer, 2010, pp. 29–37.
[146] D. Penas, J. Banga, P. González, R. Doallo, Enhanced parallel Differential Evolution algorithm for problems in computational systems biology, Appl. Soft Comput. 33 (2015) 86–99.
[147] D. Izzo, M. Ruciński, C. Ampatzis, Parallel global optimisation meta-heuristics using an asynchronous island-model, in: Evolutionary Computation, 2009. CEC'09. IEEE Congress, 2009, pp. 2301–2308.
[148] D. Whitley, S. Rana, R.B. Heckendorn, The island model genetic algorithm: on separability, population size and convergence, J. Comput. Inf. Technol. 7 (1999) 33–48.
[149] F. Lardeux, A. Goëffon, A dynamic island-based genetic algorithms framework, in: K. Deb, A. Bhattacharya, N. Chakraborti, P. Chakroborty, S. Das, J. Dutta, et al. (Eds.), Simulated Evolution and Learning: 8th International Conference, SEAL 2010, Kanpur, India, December 1-4, 2010. Proceedings, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 156–165.
[150] M. Tomassini, L. Vanneschi, Introduction: special issue on parallel and distributed evolutionary algorithms, part I, Genet. Program. Evolvable Mach. 10 (2009) 339–341.
[151] M. Tomassini, L. Vanneschi, Guest editorial: special issue on parallel and distributed evolutionary algorithms, part two, Genet. Program. Evolvable Mach. 11 (2010) 129–130.
[152] I. Arnaldo, I. Contreras, D. Millán-Ruiz, J.I. Hidalgo, N. Krasnogor, Matching island topologies to problem structure in parallel evolutionary algorithms, Soft Comput. 17 (2013) 1209–1225.
[153] D. Zaharie, D. Petcu, S. Panica, A hierarchical approach in distributed evolutionary algorithms for multiobjective optimization, in: I. Lirkov, S. Margenov, J. Waśniewski (Eds.), Large-scale Scientific Computing: 6th International Conference, LSSC 2007, Sozopol, Bulgaria, June 5-9, 2007. Revised Papers, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 516–523.
[154] S. Janson, M. Middendorf, A hierarchical particle swarm optimizer, in: Evolutionary Computation, 2003. CEC '03. The 2003 Congress, vol. 2, 2003, pp. 770–776.
[155] S. Janson, M. Middendorf, A hierarchical particle swarm optimizer for dynamic optimization problems, in: Applications of Evolutionary Computing, Springer, 2004, pp. 513–524.
[156] S. Janson, M. Middendorf, A hierarchical particle swarm optimizer and its adaptive variant, IEEE Trans. Syst. Man Cybern. Part B Cybern. 35 (2005) 1272–1282.
[157] P. Ghosh, H. Zafar, S. Das, A. Abraham, Hierarchical dynamic neighborhood based Particle Swarm Optimization for global optimization, in: 2011 IEEE Congress of Evolutionary Computation (CEC), 2011, pp. 757–764.
[158] P. Bujok, Hierarchical topology in parallel differential evolution, in: I. Dimov, S. Fidanova, I. Lirkov (Eds.), Numerical Methods and Applications: 8th International Conference, NMA 2014, Borovets, Bulgaria, August 20-24, 2014, Revised Selected Papers, Springer International Publishing, Cham, 2015, pp. 62–69.
[159] F. Herrera, M. Lozano, C. Moraga, Hierarchical distributed genetic algorithms, Int. J. Intell. Syst. 14 (1999) 1099–1121.
[160] G. Folino, G. Spezzano, P-CAGE: an environment for evolutionary computation in peer-to-peer systems, in: P. Collet, M. Tomassini, M. Ebner, S. Gustafson, A. Ekárt (Eds.), Genetic Programming: 9th European Conference, EuroGP 2006, Budapest, Hungary, April 10-12, 2006. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 341–350.
[161] M. Clerc, Back to Random Topology, 2007.
[162] Q. Ni, J. Deng, A new logistic dynamic particle swarm optimization algorithm based on random topology, Sci. World J. 2013 (2013).
[163] M. Clerc, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, in: Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress, vol. 3, 1999, pp. 1–1957.

[164] Q. Ni, C. Cao, X. Yin, A new dynamic probabilistic Particle Swarm Optimization with dynamic random population topology, in: 2014 IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 1321–1327.

[165] Y. Liu, D. Luo, Q. Zhao, C. Sui, Particle swarm optimizer based on small-world topology and comprehensive learning, in: D.-S. Huang, Z. Zhao, V. Bevilacqua, J.C. Figueroa (Eds.), Advanced Intelligent Computing Theories and Applications: 6th International Conference on Intelligent Computing, ICIC 2010, Changsha, China, August 18-21, 2010. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 595–601.

[166] M. Vora, T.T. Mirnalinee, Small world particle swarm optimizer for global optimization problems, in: P. Maji, A. Ghosh, M.N. Murty, K. Ghosh, S.K. Pal (Eds.), Pattern Recognition and Machine Intelligence: 5th International Conference, PReMI 2013, Kolkata, India, December 10-14, 2013. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 575–580.

[167] M. Giacobini, M. Tomassini, A. Tettamanzi, Takeover time curves in random and small-world structured populations, in: Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, 2005, pp. 1333–1340.

[168] Z. Zhu, Particle swarm optimization with Watts-Strogatz model, in: B.K. Panigrahi, S. Das, P.N. Suganthan, S.S. Dash (Eds.), Swarm, Evolutionary, and Memetic Computing: First International Conference on Swarm, Evolutionary, and Memetic Computing, SEMCCO 2010, Chennai, India, December 16-18, 2010. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 506–513.

[169] M. Giacobini, M. Preuss, M. Tomassini, Effects of scale-free and small-world topologies on binary coded self-adaptive CEA, in: Evolutionary Computation in Combinatorial Optimization, Springer, 2006, pp. 86–98.

[170] R. Subbu, A.C. Sanderson, Modeling and convergence analysis of distributed coevolutionary algorithms, IEEE Trans. Syst. Man Cybern. Part B Cybern. 34 (2004) 806–822.

[171] F. v. d. Bergh, A.P. Engelbrecht, A Cooperative approach to particle swarm optimization, IEEE Trans. Evol. Comput. 8 (2004) 225–239.

[172] J. Jie, J. Zeng, C. Han, Q. Wang, Knowledge-based cooperative particle swarm optimization, Appl. Math. Comput. 205 (11/15/2008) 861–873.

[173] Yuhua Li, Zhi-Hui Zhan, Shujin Lin, Jun Zhang, Xiaonan Luo, Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems, Inf. Sci. 293 (2015) 370–382.

[174] Y. Shi, H. Teng, Z. Li, Cooperative Co-evolutionary differential evolution for function optimization, in: L. Wang, K. Chen, Y.S. Ong (Eds.), Advances in Natural Computation. ICNC 2005. Lecture Notes in Computer Science, vol. 3611, Springer, Berlin, Heidelberg, 2005.

[175] Fu-zhuo Huang, Ling Wang, Qie He, An effective co-evolutionary differential evolution for constrained optimization, Appl. Math. Comput. 186 (1) (2007) 340–356.

[176] B. Liu, H. Ma, X. Zhang, Y. Zhou, A memetic co-evolutionary differential evolution algorithm for constrained optimization, in: Evolutionary Computation, 2007. CEC 2007. IEEE Congress, 2007, pp. 2996–3002.

[177] M.A. Potter, K.A. De Jong, A cooperative coevolutionary approach to function optimization, in: Y. Davidor, H.-P. Schwefel, R. Männer (Eds.), Parallel Problem Solving from Nature — PPSN III: International Conference on Evolutionary Computation the Third Conference on Parallel Problem Solving from Nature Jerusalem, Israel, October 9–14, 1994 Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 1994, pp. 249–257.

[178] T. Friedrich, P.S. Oliveto, D. Sudholt, C. Witt, Theoretical analysis of diversity mechanisms for global exploration, in: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, 2008, pp. 945–952.

[179] E. Yu, P.N. Suganthan, Ensemble of niching algorithms, Inf. Sci. 180 (2010) 2815–2833.

[180] A. Zamuda, J. Brest, Self-adaptive control parameters' randomization frequency and propagations in differential evolution, Swarm Evol. Comput. 25 (12/2015) 72–99.

[181] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. 24 (10/2015) 11–24.

[182] S. Hui, P.N. Suganthan, Ensemble and arithmetic recombination-based speciation differential evolution for multimodal optimization, IEEE Trans. Cybern. 46 (2016) 64–74.