Journal of Network Intelligence ©2022 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **7**, Number **3**, August **2022**

Overview of Parallel Computing for Meta-Heuristic Algorithms

Ying Sun

College of Computer Science and Engineering
Shandong University of Science and Technology
No.579 Qianwan'gang Road, Qingdao, Shandong, 266590, China
603345365@qq.com

Shu-Chuan Chu*

College of Computer Science and Engineering
Shandong University of Science and Technology
No.579 Qianwan'gang Road, Qingdao, Shandong, 266590, China
scchu0803@gmail.com

Pei Hu

College of Computer Science and Engineering
Shandong University of Science and Technology
No.579 Qianwan'gang Road, Qingdao, Shandong, 266590, China
huxiaopei163@163.com

Junzo Watada

Graduate School of Information, Production and Systems
Waseda University
Kitakyushu 808-0135, Japan
junzo.watada@gmail.com

Mingchao Si

College of Computer Science and Engineering
Shandong University of Science and Technology
No.579 Qianwan'gang Road, Qingdao, Shandong, 266590, China
SMC_95@163.com

Jeng-Shyang Pan

College of Computer Science and Engineering
Shandong University of Science and Technology, China
Chaoyang University of Technology, Taiwan
Fujian University of Technology, China
jengshyangpan@gmail.com

*Corresponding author: Shu-Chuan Chu

Received May 19, 2022, revised June 13, 2022, accepted July 20, 2022.

Abstract. *The meta-heuristic algorithm is used in the research of various complex problems. Due to the limitations of the original meta-heuristic algorithm, many improved meta-heuristic algorithms have been proposed, such as compact, adaptive, multi-objective and parallel schemes. Among them, the parallel strategy may get a significant improvement for related applications. This paper mainly studies the application of parallel computing in meta-heuristic algorithms. There are two main types of parallelism: one is absolute parallelism, using multiple processors which can solve optimization problems with high computational costs and improve execution efficiency. The other is virtual parallelism (multi-grouping), which decomposes the population into multiple sub-populations, and each sub-population communicates between species to generate better solutions. In addition, the combination of parallel computing and meta-heuristic algorithms can solve a wide variety of application problems: path planning, engineering design, large-scale optimization, image segmentation, neural networks and prediction problems, etc. This paper presents a comprehensive study and systematic survey of parallel meta-heuristics.*
Keywords: Meta-heuristic, Parallel, Multi-grouping, Optimization algorithm

1. **Introduction.** In real life, there are many engineering problems to be solved, such as deep learning, data mining, intelligent computing, etc [1, 2]. As a powerful tool, intelligent computing can solve many problems that could not be solved in real life. Intelligent computing mainly optimizes the problem of a specific target step by step until the optimal solution is found. It can be seen that intelligent computing is compelling and widely used in various fields. Some of the proposed meta-heuristic algorithms have been shown to be effective: Genetic Algorithm (GA) [3, 4], which reflects the process of natural selection. Grenade Explosion Method (GEM) [5] is a new algorithm based on the principle of the grenade explosion. The algorithm takes into account the damage caused to surrounding objects when throwing a grenade. The greater the loss, the better the throwing position. The Sine Cosine Algorithm (SCA) [6] uses the sine and cosine functions to move the population in the direction of the best solution. Jaya optimization algorithm [7] was proposed in 2016. It turns out that it is a powerful algorithm, and it is always going to win. Ant Colony Optimization (ACO) [8, 9] is a very interesting and effective discrete optimization algorithm. Ants move randomly or along the most traveled path (containing the most pheromone). The Grasshopper Optimization Algorithm (GOA) [10] is an algorithm inspired by nature. Grasshoppers have a small range of motion during their juvenile years, which can be seen as a developmental stage. During adulthood, the range of motion is relatively large and can be seen as the exploratory stage. Also, due to external disturbances, the grasshopper will jump suddenly and randomly. QUasi-Affine TRansformation Evolutionary (QUATRE) algorithm [11] is a swarm-based optimization algorithm that uses affine transformation as an evolutionary method to optimize by using cooperative cooperation among particles. Phasmatodea Population Evolution (PPE) algorithm [12] is inspired by nature, simulates the four growth modes of stick insects, and divides them into four evolutionary stages: convergent evolution, path dependence, population growth, and population competition. PPE passes beneficial genes to the next generation, allowing them to survive better.

There are certain shortcomings because the meta-heuristic algorithm cannot balance the relationship between exploration and development. So many improved meta-heuristics have been proposed. Among them, the parallel strategy is a common improvement method of the meta-heuristic algorithm. There are two types of parallelism: one is parallelism on hardware [13, 14]; the other is grouping [15], which communicates current information every certain number of iterations. At present, many researchers have used parallel strategies to improve the meta-heuristic algorithm [16, 17], and the improved meta-heuristic

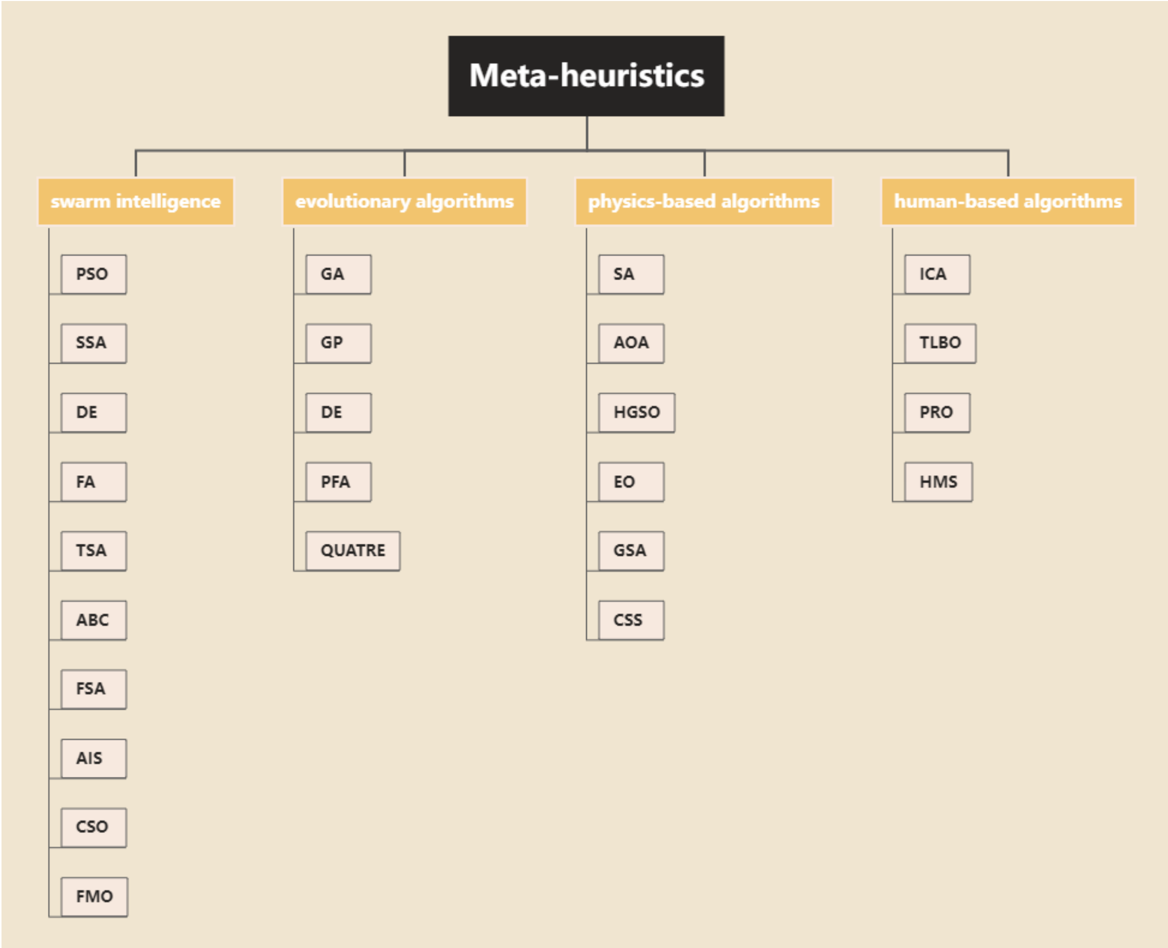


Figure 1. Classification of meta-heuristics

algorithm has been widely used in various fields. For example, the Parallel Particle Swarm Optimization (PPSO) algorithm with three communication strategies [15], parallel Multi-Swarm Particle Swarm Optimization strategy for solving Multi-objective optimization problems (MOPSO) [18], Enhanced Equalization Optimizer (EEO) [19].

The remainder of this paper is organized as follows. The second part describes four common categories of meta-heuristics and some improved strategies. The third part introduces absolute parallelism, mainly including the platforms and models needed to realize parallelism. In addition, the parallel algorithm is described. The fourth part summarizes the multi-grouping algorithms and divides them into six categories. The fifth part sorts out the application problems involved in parallel and explains them in detail. The last section summarizes the work of this paper and makes some substantive suggestions for future research.

2. Meta-heuristics.

This section presents a taxonomy of meta-heuristics and seven common improvement strategies.

2.1. Classification of meta-heuristics.

Generally, metaheuristic algorithms can be divided into four categories according to the source of inspiration: swarm intelligence, evolutionary algorithms, physics-based algorithms, and human-based algorithms [20]. Figure 1 shows the classification of metaheuristics.

Swarm intelligence is a meta-heuristic algorithm generated by imitating the behavior of animals or insects. For example, the most popular swarm intelligence algorithm, the

PSO algorithm, is inspired by birds' foraging behavior [21]. The Salp Swarm Algorithm (SSA) was proposed by Mirjallili etc. [22], by simulating the swarming behavior of salps when they move and prey in the ocean. SSA divides the population into two populations, one is used for exploration, which it is responsible for guiding at the front of the team. Another population is in charge of exploitation, following the guide searching for food. The Dolphin Echolocation (DE) algorithm [23] was inspired by the unique echolocation of dolphins. The dolphin can judge the distance to the object by the sound it makes, and it can judge the direction of the object by the strength of the signal received by the head. The Firefly Algorithm (FA) [24] is a traditional meta-heuristic algorithm based on the flickering behavior of fireflies. Fireflies communicate through light, and the stronger the light, the better the fitness value. All firefly individuals will follow the brightest individual. This category also includes some other algorithms: Tunicate Swarm Algorithm (TSA) [25], Artificial Bee Colony (ABC) [26], Fish Swarm Algorithm (FSA) [27], Artificial Immune System (AIS) [28], Shuffled Frog Leaping Algorithm (SFLA) [29], Cat Swarm Optimization (CSO) [30, 31], Ant Colony Optimization (ACO) [32], Fish Migration Optimization (FMO) [33], Cuckoo Optimization (CS) [34], Whale Optimization Algorithm (WOA) [35] et al.

Inspired by biological evolution, evolutionary algorithms use the crossover and mutation of groups to iterate. The most popular evolutionary algorithm is the Genetic Algorithm (GA) [3], which imitates Darwin's theory of evolution. Genetic Programming (GP) [36] is different from GA in encoding method and encoding length. It uses tree form for encoding and the encoding length is not fixed. Differential Evolution (DE) algorithm [37] is also more common in evolutionary algorithms, which was proposed by Storn and Pirce. Evolutionary algorithms also include: Paddy Field Algorithm (PFA) [38], QUasi-Affine TRansformation Evolutionary (QUATRE) [11], etc.

Physics-based algorithms are inspired by physical phenomena in life. Such algorithms are generated by simulating how physical phenomena occur. The most common is the Simulated Annealing (SA) [39], which uses the heating phenomenon in thermodynamics to simulate the changing process of cooling of an object. The Archimedes Optimization Algorithm (AOA) [40] imitates the Archimedes principle in physics. AOA introduces the formula for when objects sink and sink in water into the algorithm for iterative update. Algorithms based on physical phenomena include: Henry Gas Solubility Optimization (HGSO) [41], Equilibrium Optimization (EO) [42], Gravity Search Algorithm (GSA) [43], Charged System Search (CSS) [44], etc.

The last category of algorithms is inspired by human behavior in social activities. The Imperialist Competition Algorithm (ICA) [45] was proposed by Atshpaz-Gargari and Lucas in 2007. A country consists of two kinds of individuals, colonial and imperial, who compete with each other and eventually converge into a country, abstracting this process into an ICA algorithm. Teaching–Learning-Based Optimization (TLBO) [46] is divided into two processes: teacher teaching and student learning. This type of algorithm also includes: Wealth and Poor Optimization (PRO) [47], Human Mental Search (HMS) [48], Search and Rescue optimization algorithm (SAR) [49], etc.

2.2. **Meta-heuristic improvements.** The meta-heuristic algorithm cannot balance the relationship between exploration and exploitation. To overcome this shortcoming, some improved versions of meta-heuristics have been proposed. The main strategies for improvement are binary, self-adaptive, compact, multi-objective, hybrid algorithm, discretization, and opposition-based Learning.

(1) The binary optimization algorithm is mainly used to solve the optimization problem of binary space (feature selection, knapsack problem). Hu et al. [50] proposed the binary

Gray Wolf Optimization (BGWO), and used four transfer functions to Continuous values are mapped to discrete binary values. The functions are applied to feature selection problems to improve understanding schemes. Du et al. [51] proposed the Binary Symbiotic Organism Search (BSOS) applied to the feature selection problem.

(2) Self-adaptive strategy is the most basic improvement method of evolutionary algorithm. It can adjust parameters adaptively according to experience so that it can quickly find the optimal value. Deb and Beyer proposed an adaptive genetic algorithm [52] in 2001. Xue et al. [53] proposed an adaptive artificial bee colony optimization algorithm in 2018. Zhao et al. proposed a hybrid algorithm based on an adaptive gravitational search algorithm and differential evolution [54]. Meng et al. [55] proposed an improved differential evolution algorithm (PaDE) and used an adaptive scheme to control the parameters to optimize the real-parameter single-objective problem.

(3) The compact strategy [56] is different from the traditional optimization strategy in that it uses less memory and uses a probabilistic model to represent the distribution of individuals. The Compact Genetic Algorithm (CGA) [57] was the first algorithm to implement the compact strategy, and later Compact Differential Evolution (CDE) [58], Compact Particle Swarm Optimization (CPSO) [59], Compact Cuckoo Search Algorithm (CCSA) [60], Compact Bat Algorithm (CBA) [61], etc. Overall, the most used probability model is the Gaussian distribution. Firefly Algorithm (FA) [54] introduces 12 chaotic maps, and it is finally proved that the Gaussian map is used to optimize the coefficients.

(4) Multi-objective optimization involves multiple objective functions. Therefore, to find the optimal solution, it is necessary to compare multiple objective functions to obtain the global optimal value [62, 63, 64]. Common multi-objective meta-heuristic algorithms are Multi-objective Particle Swarm Optimization (MOPSO) [65], Multi-objective Whale Optimization Algorithm (MOWOA) [66], Multi-objective Ant Colony Optimization (MOACO) [67], Multi-objective Ion Motion Optimization (MOIMO) [68].

(5) Hybrid algorithm combines two or more meta-heuristic algorithms and integrates the advantages of multiple algorithms to solve complex application problems. Combining TLBO with DE, an efficient hybrid algorithm is proposed [69]. Hu et al. [70] proposed a novel hybrid algorithm, which integrates the advantages of SFLA and GWO, and effectively solves the problem of daily power load forecasting .

(6) In order to solve discrete problems in the real world, the paper [71] proposed a discrete fish migration optimization algorithm and introduced Hamming Distance (HD) to solve the Traveling Salesman Problem (TSP). The paper [72] proposed a Discrete Firefly Algorithm (DFA) to solve the formation problem of manufacturing cells. The paper [73] proposed a Discrete Artificial Bee Colony (DABC) optimization algorithm, which efficiently solves the discrete flow shop scheduling problem.

(7) Opposition-based Learning (OBL) [74] was first introduced in 2005, inspired by real-life opposites. The paper [75] proposed an Opposition-based Learning Differential Evolution (OBLDE), which accelerates the convergence of the original algorithm. The paper [76] proposed an improved sine cosine algorithm, which increases the performance of the meta-heuristic algorithm by finding the opposite position of its solution in the search space. The paper [77] proposed an improved grasshopper optimization algorithm using an adversarial learning strategy.

3. **True parallelism.** When combining parallel computing with meta-heuristic algorithms, we need to consider many factors, such as which computing platform is implemented, what model is used, etc., which will affect the algorithm's execution time. This section will introduce two parallel computing platforms, four parallelization models, communication methods, and some common parallel metaheuristics.

3.1. **Implementation platform.** When the parallel meta-heuristic algorithm is executed, the implementation platform will significantly affect the execution efficiency of the algorithm. The paper [78], according to the instruction and data flow, proposed four traditional classification methods: Single Instruction Single Data (SISD), Single Instruction Multiple Data (SIMD), Multiple Instruction Multiple Data (MISD), Multiple Instruction Single Data (MIMD). Currently, some typical implementation platforms are Graphics Processing Unit (GPU), distributed computing platform, and Field Programmable Gate Array (FPGA). This section mainly introduces two parallelization platforms.

3.1.1. *GPU-based parallelization.* Among many hardware platforms, GPU is one of the most significant accelerators. It utilizes graphics processors to significantly improve the computational efficiency of meta-heuristic algorithms. In [79], a new parallel method is proposed, which increases the number of threads of CUDA. Finally, the parallel implementation of the algorithm on GPU is faster than the sequential execution of the CPU. Since GPUs have more cores to run a large number of threads, it is not feasible to increase the algorithm's speed simply by increasing the number of individuals. A new optimization algorithm is proposed in [80], which allows each subgroup to search in different neighborhoods (multi-distance search strategy) to achieve the purpose of increasing the diversity of the population.

The paper [81] applied a meta-heuristic search algorithm to discrete black-box problems using a Graphics Processing Unit (GPU) platform. A GPU consists of multiple processors, each with multiple cores. It is experimentally verified that the implementation in this paper is ten times faster than an optimized multi-core CPU. The paper [82] used GPU parallelism to improve the performance of local search algorithms. This paper proposed a new distributed local search process specially designed for multi-core and multi-GPU systems to achieve better performance. Due to the excellent performance of ACO after parallelization, the paper [83] presented three different algorithms, all of which are GPU-based ACO. The paper [84] researched the implementation, application, and development of the parallel genetic algorithm, parallel particle swarm algorithm, parallel differential evolution algorithm, and parallel simulated annealing algorithm on GPU. In order to make full use of the resources provided by the parallel computing GPU platform, the paper [85] processed a large number of particles through a fast GPU core and divided them into several particle swarms. The paper [86] developed a parallel bee algorithm running on GPU, and modified the local search process of the original algorithm, and avoiding the waste of GPU computing power. At the same time, parity sorting and two-stage communication are also used to achieve a better convergence effect. The paper [87] parallelized the Improved Invasive Weed Optimization (IIWO) algorithm and applies it to the GPU platform. At the same time, compared with the serial IIWO algorithm, the parallelized algorithm runs fewer iterations and obtains better results. The paper [88] used multi-core technology to parallelize and evaluate two algorithms, TLBO and Jaya. Experiments show that the iteration time of TLBO is higher than that of Jaya, but the former converges faster than the latter in a less iteration range. The paper [89] utilized high-throughput accelerator GPUs to improve the performance of ACO.

3.1.2. *FPGA-based parallelization.* FPGA is a new chip technology between application-specific integrated circuit chips and general-purpose chips. Its parallelism can improve the real-time processing of data. Using the FPGA platform in [90] to deal with complex algorithms has solved the risk of delay. The paper [91] used PSO to solve real-time optimization problems and applied the algorithm to FPGA, which improves the speed of real-time processing of the algorithm. The paper [92] proposed a method that combines three parallel PSOs and a communication operator on a GPGA chip, which improves the

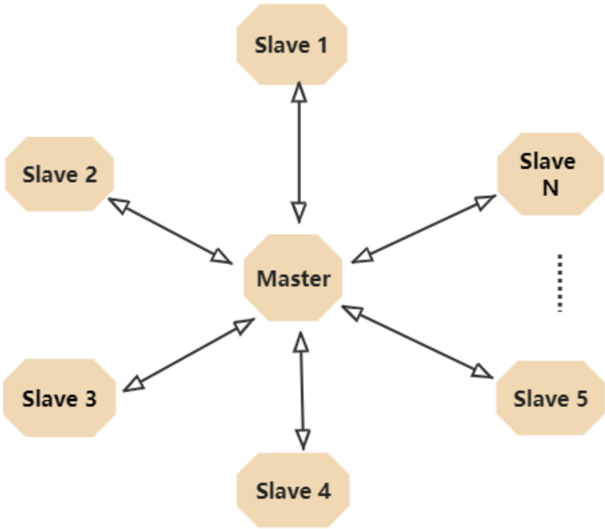


FIGURE 2. Master-slave mode

diversity and convergence of the algorithm. In order to improve the parallelism of intelligent optimization algorithms, the paper [93] studied a parallel design method based on FPGA. In order to verify the validity of the proposed method, two common optimization algorithms (PSO and GA) are used for verification, and the results have good real-time performance. Finally, the parallel simulated annealing algorithm (SA) is used to solve the Job Shop Scheduling Problem (JSSP), which further verifies the effectiveness of the proposed algorithm. Homomorphic encryption technology is also very challenging in recent years, but there are specific problems. It runs for a long time on a general-purpose computer and cannot be processed in real-time. Therefore, the paper [94] spent a year designing a heterogeneous platform. The platform combines an FPGA processor with an Arm processor and adds parallel processing to achieve desirable results. In order to solve the communication and hardware problems when BP neural network processes big data, the paper [95] proposed a PSO algorithm based on MapReduce, which makes the classification accuracy rate reach 92%.

3.2. **Parallelization model.** Parallelization models are mainly divided into the following four types: master-slave model, island model (coarse-grained), cell meta-model (fine-grained) and mixed model [96, 97, 98, 99, 100, 101].

(1) Master-slave mode: The master-slave mode has one master processor and multiple slave processors. All operations work in parallel on the slave processors and are controlled by the master processor. As shown in Figure 2, the master-slave mode structure is similar to the star [102]. The paper [92] used the master-slave mode PPSO algorithm to solve the path planning problem. In order to shorten the running time, three PSO slave processors and one master process are used for concurrent execution. Every fixed number of generations, information is exchanged between the slave and master processors, and a better solution has been obtained. The paper [103] proposed a parallel Comprehensive Learning Particle Swarm Optimizer (PLCPSO) algorithm and used the master-slave model to run each sub-swarm on different machines to achieve the global optimum. Moreover, the various subpopulations regularly cooperate in exchanging optimal solutions.

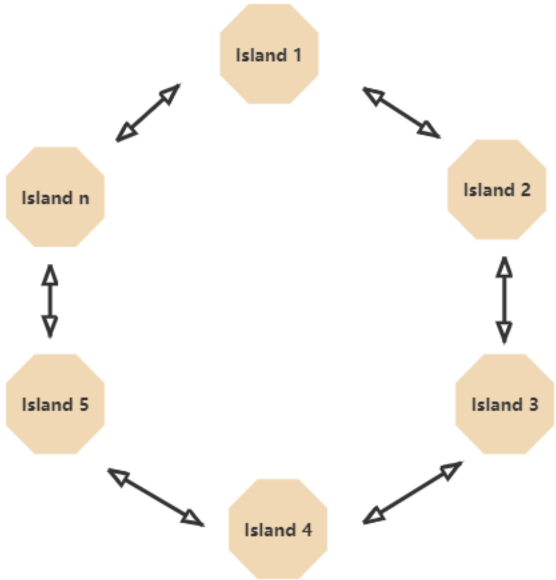


Figure 3. Coarse-grained model

(2) Coarse-grained model: The coarse-grained model, also known as the island model, divides a population into multiple sub-populations, each of which represents a processor unit. As shown in Figure 3, each subpopulation communicates through an exchange strategy, thereby exchanging some individuals. The paper [104] uses the island model and immigration to solve the one-dimensional bin packing problem. This model allows the islands to cooperate with each other and use the migration operation for the initialization phase. The paper [105] proposes a parallelized evolutionary algorithm and uses a multi-swarm island model. Within each population, fewer individuals exchange.
(3) Fine-grained model: The fine-grained model, also known as the cell meta-model, divides a population into multiple small sub-populations and maps the sub-populations to a two-dimensional grid. Typically, each subpopulation contains four domain individuals. However, when exchanging information with domain individuals, it is faster, and there are delays when communicating with other individuals. As shown in Figure 4, each subpopulation exchanges information with the surrounding four subpopulations. In [106], a strategy based on asynchronous parallel meta-cellular genetic algorithm is proposed, which divides the population into multiple populations, and each population runs on a different processor. Individuals within the population are grid-like and connected to each other.
(4) Hybrid model: A parallelized model composed of two or more models above is called a hybrid model, as shown in Figure 5.

3.3. **Communication method.** In a parallel meta-heuristic algorithm, the population is divided into multiple sub-populations, which are executed on multiple processors respectively by parallelizing the model. The subpopulations should communicate with each other to avoid falling into local optimum. Therefore, this section organizes four common communication methods.

(1) Star: The star structure [107] is based on the master-slave model. The master group in the middle transmits information with the surrounding subgroups, and there is no

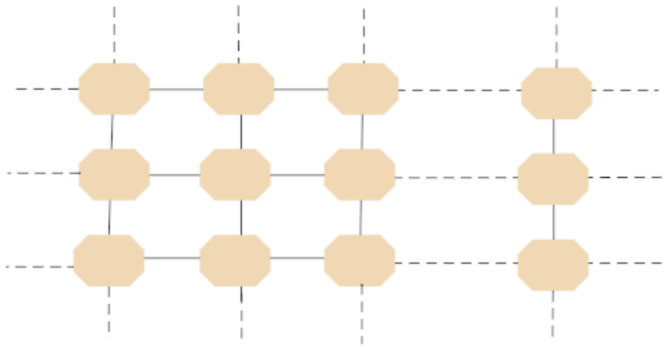


Figure 4. Fine-grained model

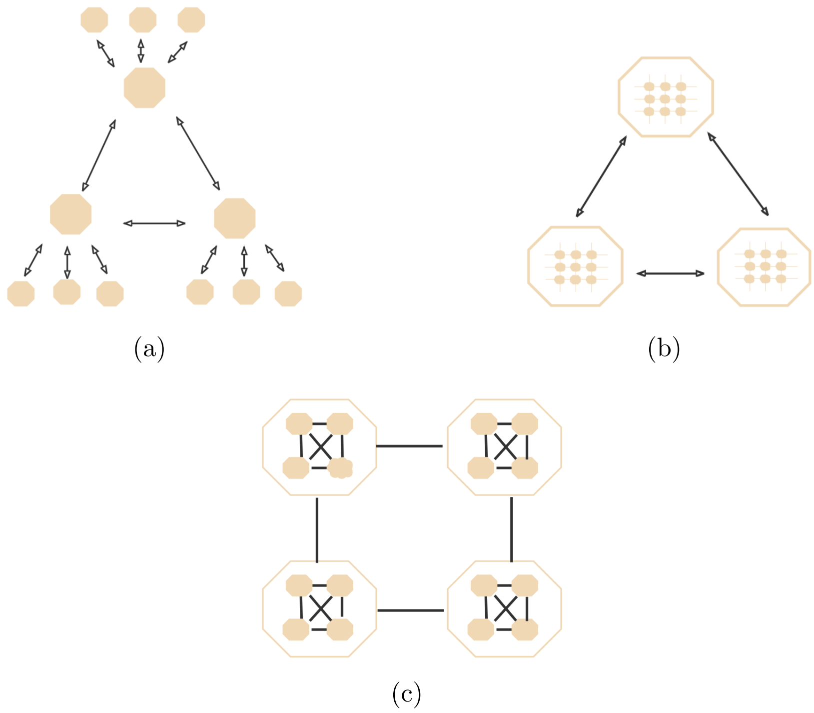


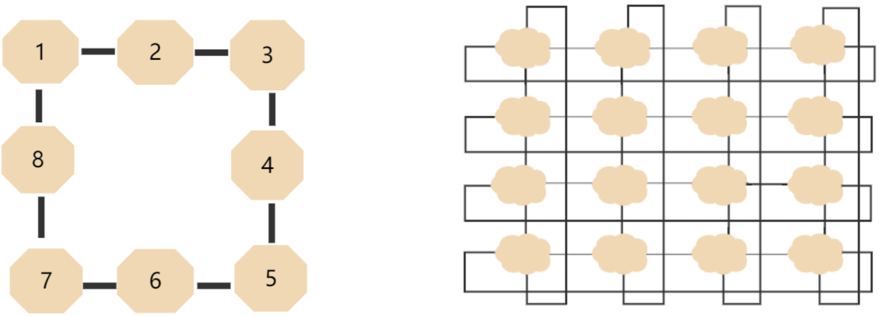
Figure 5. Hybrid model

communication between the subgroups. The subgroup is responsible for updating and iterating, and the final individual optimal value is handed over to the main group.

(2) Mobile: The moving mechanism is similar to the ring structure [108, 109, 110], and each sub-population can only communicate with the adjacent population. In order to collaborate between groups, the paper [111] combined migration and cooperation strategies to increase the diversity of understanding.

(3) Broadcast: Different from the movement method, each sub-population of the broadcast strategy [112] can communicate with each other, forming a fully connected graph.

(4) Diffusion: The diffusion strategy [113, 114] is similar to the mobile strategy, except that the number of communication sub-populations is different. In the move strategy, the main population communicates with each subpopulation. However, in the diffusion strategy, the main population intelligence exchanges information with the four sub-populations up, down, left and right.



(a) *bidirectional ring topology* (b) *torus topology*

FIGURE 6. Hybrid communication method

(5) Other: The random combination of the above four methods constitutes a hybrid communication method [115]. A new parallel elite-biased framework is proposed [116], which uses a different way to exchange information than before. The specific implementation is shown in the following Figure 6: bidirectional ring topology and torus topology (each process has four neighbors)

3.4. **Parallel meta-heuristics.** More and more meta-heuristic algorithms are parallelized to solve real-life problems. The following is a brief description of some parallel meta-heuristic algorithms commonly used in recent years.

Parallel Genetic Algorithm (PGA):

Genetic algorithm is a meta-heuristic algorithm that simulates the theory of species evolution. The combinatorial optimization problem is difficult and complex, and it is difficult to find a suitable solution. A novel scalable parallel grouping genetic algorithm [117] is proposed for this problem, which uses an island model. In [118], the most time-consuming part is handed over to the parallel module for execution. These parallel modules are not simple selection, crossover, and mutation. It is the calculation of the fitness value of the initial population and the calculation of the fitness value of the offspring.

Parallel Simulated Annealing (PSA):

The SA algorithm is a physics-based algorithm that simulates the principle of solid-state annealing. The vehicle routing problem has been widely concerned in order to solve this problem better. A parallel SA algorithm is proposed, which introduces the Markov process, and master-slave structure [119]. In [120], Markov chains are also introduced to parallelize the original SA algorithm. Nevertheless, the difference from the former is that the Monte Carlo algorithm is added to accelerate the convergence efficiency of the algorithm.

Parallel Chaos Optimization Algorithm (PCOA):

The paper [121] used the master-slave model to improve COA and parallelize it to solve the problem of parameter identification. Two carrier chaos searches are used in the master process, and a migration and crossover structure is used in the slave process. The paper [111] also improved the original algorithm in parallel, using two operations of migration and merging, so that each sub-population will not be executed independently, which improves the diversity of the algorithm. At the same time, the global and local search of the algorithm is better balanced.

Parallel Particle Swarm Optimization (PPSO):

PSO is a popular algorithm developed by simulating the foraging behavior of birds. There are more and more improved versions of PSO, and a new multi-swarm variant particle swarm algorithm has been proposed, which has a greater advantage in execution time than traditional algorithms [122]. At the same time, the paper [123] attempted to parallelize the algorithm at the generation level for the first time. Some particles execute the current operation formula, and other particles can execute new operations in the next generation, which greatly shortens the running time. The paper [124] proposed a new constrained optimization algorithm called Parallel Boundary Search Particle Swarm Optimization (PBSPSO). In a global search, the algorithm uses a penalty function to search. In local search, Sequential Quadratic Programming (SQP) is performed. Then compare the results of this time with the results obtained from the previous global search, and choose the best one.

Parallel Artificial Bee Colony (PABC):

The ABC optimization algorithm is generated to simulate the foraging behavior of bees. The bee colony consists of three parts, the employed bees, bystanders, and scouts. To improve the performance of the original ABC optimization algorithm, it is parallelized [125]. At the same time, the colony is divided into multiple subcolonies, and parallel operations are performed in each sub colony. A dynamic migration operation is adopted between each bee colony to improve the efficiency of the algorithm. The paper [126] also made parallel improvements to the ABC algorithm. First, find the parts that can be executed in parallel and design them. Then, the parallel ABC algorithm is implemented based on the ternary light computer. Finally, the algorithm is applied to the function optimization problem to verify the performance of the algorithm. The paper [127] used three models (master-slave model, island model with migration, and hybrid model) to improve the ABC algorithm and analyze the performance between the three algorithms. ASADZADEH implements Parallel Artificial Bee Colony (pABC) [125] using three models: master-slave model, fine-grained model and coarse-grained model. In the coarse-grained model, pABC sets up a colony on each processor, a total of n, the domain topology used by each colony is a cube, and the information communication between every two colonies is dynamic.

Parallel Ant Colony Optimization (PACO):

In order to study the dual-objective workshop scheduling problem, an improved ACO algorithm is proposed. In order to reduce the search space, the algorithm provides a variety of dynamic heuristic information to guide to finding the optimal solution [128]. Due to the ever-increasing complexity of computing, a new computing method is proposed, which can distribute a large number of centralized operations to different PCs. Each PC is executed in parallel, reducing the computational complexity [129]. The paper uses 3-Opt and master-slave mode to improve the ACO algorithm. Finally, through experimental comparison, the algorithm can avoid local minima and is more reliable [130].

Of course, there are some other parallel meta-heuristics (see Table 1). For example, parallel random tree algorithm [131] [132], parallel multi-objective random search algorithm [133], Parallel Discrete Lion Swarm Optimization Algorithm (PDLSO) [134], Parallel Transition Evolution (PTE) [135], Parallel Hybrid Intelligent Algorithm [136], Parallel Imperialist Competition Algorithm (PICA) [137], Parallel Low Memory Quasi-Newton Optimization Algorithm (PP-LBFGS) [138], Parallel Tabu Search (PTS) [139], Parallel Social Spider Optimization (PSSO) [140], Parallel Chaos Local Search Enhanced Harmony Search (MHS-PCLS) [141], Adaptive Chaos Parallel Clonal Selection Algorithm (ACPCSA) [142], etc.

4. **Virtual parallelism (multi-grouping).** This section presents multi-group variants of some common algorithms in the following summaries. We introduce multi-group Cat Swarm Optimization (CSO) in Section 4.1, Multi-group GA in Section 4.2, Multi-group PSO in Section 4.3, Multi-group QUATRE in Section 4.4, and Multi-group in Section

Table 1. Parallel meta-heuristics

Algorithm	Reference	Year	Application
PEBGLS	[116]	2018	traveling salesman problem (TSP)
PCLPSO	[103]	2015	
pABC	[125]	2016	job shop scheduling problem (JSSP)
Parallel Bees Algorithm	[86]	2014	
PBSPSO	[124]	2018	engineering design problem
PACO-3Opt	[130]	2018	traveling salesman problem (TSP)
PIIWO	[87]	2016	large scale global optimization (LSGO)
PSSO	[140]	2018	data clustering problem
PT	[120]	2014	probabilistic sampling
PTC	[139]	2018	maximum vertex weight clique problem (MVWCP)
ACPCSA	[142]	2016	weapon target assignment (WTA)
parallel cGA	[106]	2009	MAXSAT, MMDP, and the p-median problem
PACO	[128]	2017	the bi-criteria problem
IPGGA	[117]	2018	1D Bin-Packing (1DBPP)
LS	[82]	2018	minimum latency problem (MLP)
PPSO	[143]	2014	global path planning
P3SO	[123]	2020	
GCSO	[80]	2021	
parallel ACO	[89]	2018	traveling salesman problem (TSP)
PHAFB	[136]	2016	
Jaya	[79]	2019	
SS	[81]	2017	black-box problem
MSM-PCOA	[121]	2016	the parameter identifification
MsPMmPSO	[122]	2015	the association rule extraction
multi-swarm PSO	[85]	2015	quadratic assignment problem (QAP)
GA/DE/SA/PSO	[84]	2014	
GGAs	[104]	2014	bin packing problem (BPP)
parallel random forest algorithm	[131]	2020	
PACO	[129]	2006	
MH-ES-ABC	[127]	2011	
MMO-PCOA	[111]	2015	parameter extraction and identification
MHS–PCLS	[141]	2019	engineering design problem
parallel IOA	[93]	2018	job shop scheduling problem (JSSP)
PICA	[137]	2018	
PP-LBFGS	[138]	2013	
ParMOSS	[133]	2015	bi-objective competitive facility location problem
PEA	[135]	2019	
PEAs	[105]	2002	
FRPSO	[91]	2018	the optimal message-chain structure
TOC-PABC	[126]	2019	
hybrid ABC–TLBO algorithms	[102]	2020	the multi-dimensional numerical problems
parallel-SA	[119]	2016	vehicle routing problem (VRP)
pACS	[83]	2016	traveling salesman problem (TSP)
PGA	[118]	2019	parameter estimation
parallel random tree	[132]	2019	evaluation of athletes' competitive ability
PDLSO	[134]	2020	traveling salesman problem (TSP)
PGA	[144]	1991	the school timetabling problem
PGAs	[145]	2001	
Parallel metaheuristics	[100]	2013	
MOABC	[146]	2020	software development
parallel GA	[147]	2020	transportation planning and logistics management
pABC	[125]	2016	job shop scheduling problem (JSSP)
PHGA	[148]	2004	vehicle routing problem with time windows (VRPTW)
PGSO	[149]	2020	training an Artificial Neural Network
multiple-population parallel GA	[96]	1998	
parallel GAs	[97]	2000	
PACS	[17]	2003	traveling salesman problem
roposed ACS	[150]	2004	traveling salesman problem
MOPSO	[18]	2019	many-0bjective 0ptimization problems(MaOPs)
PBM	[98]	2009	
P-ABC-TLBO	[151]	2020	multi-dimensional numerical optimization
parallel BSO	[152]	2020	field programmable gate arrays (FPGAs)
PE-EFS	[153]	2021	
PPSO	[101]	2019	
GA parallel and distributed	[154]	2020	cloud computing model
PWOA	[155]	2018	Continuous Stirred Tank Reactor (CSTR)
QQIGSA	[156]	2021	wireless sensor network(WSN)
PGA	[157]	1996	VQ codevector index assignment for noisy channels
parallel ACO	[99]	2011	
PHOA	[158]	2018	economic emission load dispatch (EELD)
P-SSO	[159]	2017	
PPSO	[160]	2009	reactor core design (CD) and fuel reload (FR)
parallel ACO	[161]	2018	traveling salesman problem (TSP)

4.5. The BA, in Section 4.6 introduces the multi-grouping DE. Some common multi-group metaheuristics are organized in Table 2.

Table 2. Multi-group metaheuristics

Algorithm	Reference	Year	Application
PPSO	[15]	2005	
MPGA	[162]	2008	traveling salesman problem (TSP)
PCSO	[163]	2008	
IMPGA-GPS	[164]	2009	
DMSDE	[165]	2010	reactive power optimization of power system
mc-ACO	[166]	2011	Parallel Assembly Line Balancing Problem (PALBP)
HGDMCPSO/DPSO	[167]	2012	clinical pathway (CP)
PSO-2S	[168]	2012	
EPCSO	[169]	2012	the aircraft schedule recovery problem
MCBA	[170]	2014	
PBA	[171]	2016	economic load dispatch problem
IMPSO	[172]	2017	cloud computing scheduling strategy
MGPSO	[173]	2017	transmission expansion planning (TEP)
IMGFA	[174]	2017	numerical optimization
FDA	[175]	2017	
PBA	[176]	2018	job shop scheduling problem(JSSP)
SGA	[177]	2019	Electromagnetic tomography technology (EMT)
P-QUATRE	[178]	2019	
AMG-QUATRE	[179]	2019	wireless sensor network (WSN)
PaDE	[180]	2019	
pcFPA	[181]	2019	wireless sensor network (WSN)
pcBA	[182]	2019	wireless sensor network (WSN)
PGWO	[183]	2019	the Prediction of Wind Power
CCMACO	[184]	2019	traveling salesmen problem (TSP)
MM-QUATRE	[185]	2020	wireless sensor network (WSN)
PSOEL	[186]	2020	motion planning of redundant robotic manipulators
pcCS	[187]	2020	three-dimensional path planning
pcDE	[188]	2020	image segmentation
MGRR-PSO	[189]	2020	black-box adversarial attacks
PMVO	[190]	2020	multilevel image segmentation
MMSCA	[191]	2020	capacitated vehicle routing problem (CVRP)
MG-GWO	[192]	2021	photovoltaic (PV) solar cell model
PSCA	[193]	2021	wireless sensor network (WSN)
PCCSP	[194]	2021	wireless sensor network (WSN)
LDTACO	[195]	2021	traveling salesmen problem (TSP)
AMSSA	[196]	2021	wind power prediction
MOGOW	[197]	2021	
APAOA	[198]	2021	robot path planning

4.1. **Multi-grouping Cat Swarm Optimization.** The original CSO algorithm was proposed by Chu et al. [30], which was inspired by the behavior of cats. The CSO regards the state of the cat at rest as the search mode of the algorithm. Once the cat finds the target, the algorithm will enter another stage, the tracking mode. At the same time, the execution ratio of the two modes is controlled by a certain probability. It is worth noting that: in the search mode, a certain probability is used to select a position from the memory pool to move.

The paper [163] proposed the parallelization of CSO, called Parallel Cat Swarm Optimization (PCSO). PCSO executes a parallel strategy in the tracking process, and when the communication conditions are met, it performs inter-group communication for the grouped individuals. The specific parallel strategies in this paper are as follows:

First, a group of individuals is randomly selected, and the group of individuals is sorted according to the size of the fitness value. The individual L with the smallest fitness value is the individual L;

Second, randomly select a group of individuals in the remaining group, and record the local optimal solution P;

Finally, use the local optimal solution P to replace the individual L with the worst fitness value.

The paper [169], an improved version of the PCSO algorithm, proposed an improved parallel cat swarm optimization algorithm (EPCSO). The intergroup communication strategy is the same as that in PCSO, however, the Taguchi orthogonal method is used to improve PSCO. The paper [194] combined the PCSO algorithm with three strategies and a compact scheme, and applied it to the Wireless Sensor Network (WSN) positioning problem.

Strategy one: Average replacement, where the local optimal solution of each group is averaged to replace the worst solution in each group.

Strategy two: Best replacement, randomly select the best value of a group to replace the worst solution of each group.

Strategy three: Weight replacement, replace the random solution of each group by multiplying the optimal value of each group with the corresponding multiplication sum of different weights.

4.2. **Multi-grouping Genetic Algorithm.** In [162], multiple populations are used instead of a single population. The parameters used by each population are inconsistent. Every certain number of generations, the optimal individual is transferred to each subpopulation one-to-one using the movement and crossover strategies. The paper [164] proposed an improved adaptive and parallel GA. In the initial stage, each population performs GA steps independently and adaptive operations on the crossover and mutation stages to get a better individual for the next update. In [167], a two-layer multi-group cooperative hybrid PSO algorithm and a discrete PSO algorithm (HGDMCPSO/DPSO) are proposed. The algorithm divides the population into a top layer and a bottom layer. The top layer is divided into five groups, one of which is the main population, and the other four are sub-populations. The five populations move in different directions and update their respective speeds. At the same time, every certain number of generations, replace the worst individual in each group with the best individual in any of the other three groups. The bottom layer contains three groups, and each group has a specific function. Among them, the subgroup G1 is responsible for a large-scale search for the current best position and the global best position for exploration. Subgroup G2 is responsible for local search and improves the convergence ability of the algorithm. The subgroup G3 is an equilibrium state. In [177], the population is divided into multiple groups, and the parameters of each population are adaptive and independently evolved within a specific range, and individuals with lower fitness values are given a higher probability of crossover and mutation. Each group is linked through migration operations, and every few generations, the best individuals in each group are moved to other populations to achieve inter-group communication.

4.3. **Multi-grouping Particle Swarm Optimization.** The original PSO algorithm was proposed in 1995, which imitated the foraging behavior of birds by assigning each bird the two attributes of position and speed. Each time, the individual optimal value and the global optimal value are used to guide the flock to move to the optimal position. Parallel Particle Swarm Optimization (PPSO) was proposed [15] in 2005. The algorithm divides the flock into groups, each group uses the same update algebra, but after every fixed number of generations, individuals are updated using three different communication strategies. The three communication strategies are as follows:

(1) Every R1 generation, mutate and update the particles. First, calculate the global optimal value Gt when the number of iterations is t, and mutate it; then, replace the worst particle in each group.
(2) Every R2 generation, the best particles in each group are $G_i t$ moved to the adjacent group, and some poor particles in the adjacent group are replaced.
(3) The third strategy is a combination of the first two strategies, using strategy one every R1 generation and strategy two every R2 generation.

The paper [168] used a main group and multiple auxiliary groups to realize the parallelization of the PSO algorithm. Among them, the main group saves the best individual among all auxiliary groups. In [172], a multi-group PSO (MGPSO) is proposed based on a discrete PSO framework. The algorithm is initialized with a unique Sobol sequence and achieves good results. To increase the diversity of the population, this algorithm divides the population into multiple groups and changes the updated formula of particle velocity. As shown in Eq.(1) and Eq.(2).

$$v_{ik}^{g+1} = \lambda \left[\alpha v_{ik} + \beta^1 \left(x_{kP} - x_i\right) + \beta^2 \left(x_{iG} - x_{ik}\right) + \beta^3 \left(x_G - x_{ik}\right)\right] \tag{1}$$

$$\lambda = 2/(2 - \phi - \sqrt{\phi^2 - 4\phi}), \quad \phi = \beta^1 + \beta^2 + \beta^3 \tag{2}$$

Where g is the number of iterations, v_{ik} represents the i-th individual in the k-th group. and x_{kP} and x_{iG} represent the k-th particle's individual optimal and the k-th group's optimal particle, respectively. x_G is the global optimal particle, β belongs to a constant between [0,1], λ is the influence coefficient. The paper [189] added random distribution on the basis of MGPSO. Two opposing acceleration coefficients are used simultaneously, one for exploration and one for development. The paper [186] also adopts a multi-group form, proposing PSO's improved version, using elite groups and children's groups, using better particles in children to update poor particles in elite populations. The paper [173], to solve power system load problems, Multi-group Particle Swarm Optimization (MGPSO) algorithm is proposed under the framework of discrete PSO. At the same time, the Sobol sequence initialization and multi-set coordination policies and mutation strategies are introduced.

4.4. **Multi-grouping QUasi-Affine TRansformation Evolutionary.** Since the update method of the QUATRE algorithm is guided by the global optimal particle, it will occur in a local optimal situation. In order to avoid this, it introduces parallelization [178]. Each of the algenesses, the adjacent populations exchange part of the particles (using a preferred particle to replace a poor portion). A good way is made in a good way to confirm which two sub-populations communicate with each other. However, to improve the diversity of algorithms, a Multi-group QUasi-Affine TRansformation Evolutionary (AMG-QUATRE) is proposed [179]. The algorithm divides the population into three groups, each group adopts different mutation strategies and applies the results to the localization problem of wireless sensor networks. At the same time, during the search process, an adaptive strategy is used to control the range of parameters to better balance the performance of exploited and exploration. The paper [185] proposed three update strategies to update this group of poor particles. Each time, choose one of three update strategies to replace. As shown in Eq.(3) to (5).

$$Strategy \quad one: Xbad_{i,j} = \omega Xbad_{i,j} + r_1(Xbest_m - Xbad_{i,j}) \tag{3}$$

$$\begin{aligned} Strategy \quad two: Xbad_{i,j} =& \omega Xbad_{i,j} + r_2((\mathrm{Xbest}m + X\mathrm{best}n)/ \\ & 2 - Xbad_{i,j}) \end{aligned} \tag{4}$$

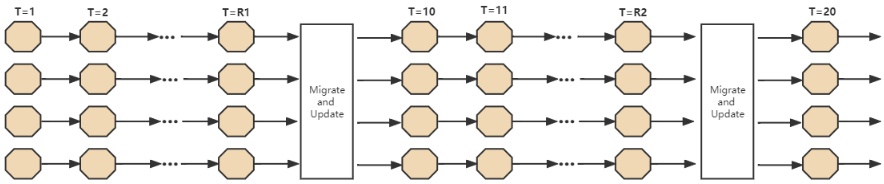


FIGURE 7. PBA communication strategy

$$Strategy\quad three : Xbad_{i,j} = \omega Xbad_{i,j} + r_3((\text{ Xbest }_m + X\text{best}_n + \text{Xbest }_t)/3 - Xbad_{i,j}) \tag{5}$$

Among them, $Xbad_{i,j}$ represents the individual of the j-th teaching poor in the ith group, and $Xbest_m$ represents the global optimal individual in the m-th group. r_1, r_2 and r_3 are continuous values with intervals 1 to 0.3, 0.3 to 0.7 and 0.7 to 1, respectively.

4.5. **Multi-grouping Bat Algorithm.** Obviously, BA [199] is inspired by the unique echolocation features of bats. It was proposed in 2010. In BA, three methods are mainly used to update the position of the bat. The first is the echolocation behavior of bats; the second is that bats emit corresponding frequencies through which distances can be judged; the third is the process of bats looking for prey. In [171], in order to solve the scheduling burden of sharing the economic load, the original bat algorithm is grouped. It enables independent individuals to run on the processor, which speeds up the convergence and accuracy of the algorithm. At the same time, the paper [176] also grouped BA. The difference from the previous one is that a random key encoding strategy and a unique communication strategy are introduced, which are applied to the shop floor scheduling problem. The communication strategy is to replace the poorly taught bats in the group with better bats in other groups. The specific implementation process is shown in Figure 7.

The paper [182] proposed an improved bat algorithm based on grouping and compaction. Among them, the grouping strategy is to exchange their information between groups, and the ways of exchanging information include moving, copying and replacing. This paper adopts three methods: ① Randomly select the best of a group to replace the worst of the current group. ② The best of all groups replaces the worst of the current group. ③ The two groups are exchanged with each other: replace the poor one in the current group with the good one in the other group.

4.6. **Multi-grouping Differential Evolution.** The original DE algorithm [37] was proposed by Storn and Price in 1997. The DE algorithm uses fewer parameters and achieves the best effect, which is very suitable for parallel computing. The DE algorithm has the following main processes: mutation, crossover and selection. In [180], an improved DE algorithm is proposed, which improves the original DE algorithm in three aspects:

(1) Adopt a grouping strategy and use an adaptive strategy for the parameters of each group. For example, the control parameter F of each individual obeys the Cauchy distribution, the crossover probability CR obeys the normal distribution, and the selection probability $P(j) = 1/j$.
(2) A parabolic population scheme is proposed, and the number of populations in each group is dynamically reduced. The detailed reduction formula is as follows:

$$ps^{t+1} = \text{round}[(ps_{\min} - ps_{ini})/(nfe_{\max} - ps_{ini})^2 \times (nfe - ps_{ini})^2 + ps_{ini}] \quad (6)$$

In Eq.(6) ps_{min} represents the minimum value of the population, and the ps_{ini} represents the initial value of the population size. The nfe represents the current individual's adaptivity value, the nfe_{max} represents the maximum value of the function, nfe_{max} represents the maximum value of the function, and the final result is rounded to obtain the number of individuals in the $t+1$ generation ps^{t+1}.

(3) In the main three processes of DE algorithm, mutation is the key process. This paper proposed a mutation scheme based on timestamp.

The paper [165], dynamically divided the population into three groups, and individuals in each group are sorted by the adaptivity value. Then, the top three individuals in each group, randomly replace the three solutions in other groups. In the next iteration, reconnect into a large group, then divided into three subproducts. The papers [188] proposed a Parallel Compact Differential Evolution (PCDE) with two communication strategies. The first is an elite policy that replaces the best solution in all groups into global optimal solutions. The second is a mean elite policy that replaces the global optimal solution for the optimal demand average in all groups.

5. **Application of parallel optimization algorithms.** In this section, we review the application of parallel computing to metaheuristic algorithms.

Power systems: How to keep the total power generation cost less affected by the outside world is a major research problem. To this end, an Adaptive Parallel Seeker Optimization Algorithm (APSOA) [200] is proposed to solve the problem of energy utilization. The paper [201] parallelized the Multi-agent Coordination Optimization (MCO) so that it can solve large-scale problems (load balancing, multi-body formation control, fragile power system). In [165], the multi-group adaptive DE algorithm is applied to the reactive power optimization of the power system. Reactive power optimization means that the active power losses in the network are minimized.

Design issues: The main discussion here is engineering design and network design. Engineering design problems mainly include the following: welded beam design, rolling element bearing design, pressure vessel design, compression spring design, three-bar truss design, reducer design, Belleville spring design problems, etc. In [202], in order to verify the effectiveness of the proposed algorithm, the improved algorithm (OPSOS) is applied to four engineering design problems. The results demonstrate the accuracy and low complexity of OPSOS. The importance of web design in our daily lives is unmistakable. The goal of network design is to survive network failures. In [203], the chaos-based Jaya algorithm effectively solves seven engineering optimization problems. Chaotic Jaya algorithm uses multi-level parallelism (coarse-grained and fine-grained parallelism). The paper [204] proposed a general parallelized hybrid algorithm to solve network design problems. The paper [141] not only optimized several engineering problems but also designed a side-impact vehicle with a negative impact. In [124], the parallelized PSO algorithm is used to optimize five engineering problems.

Structural optimization: In [205], constraint processing mainly includes penalty functions, special representations, and operators, repair algorithms, separation of goals and constraints, Lagrange multipliers, etc. In [206], the frame structure is optimized while considering the cross-sectional area of the structure.

Prediction problems: There are many kinds of forecasting problems, including wind energy forecasting, ship forecasting, etc. Among them, the predictive control of ships in the ocean is crucial. Because there are too many factors to consider, such as the disturbance of wind and waves. In [207], the most common parallel programming technology-OpenMp is used, which can effectively solve the problem of ship predictive control. Wind energy is a medical renewable resource, and the effective use of wind energy can generate considerable economic benefits. In [183], four propagation strategies are proposed to predict wind power effectively. The paper [196] used three communication strategies to improve the multi-group salps algorithm. Strategy one: Update within groups. Take the average of the k best individuals in the group to replace the k poor individuals in the current group. Strategy two: Update between groups. Take the average of the best individuals in each group to replace the k poor individuals in other groups. Strategy three: Combine strategies one and two, and divide the number of groups equally, one part uses strategy one, and the other uses strategy two.

Large scale optimization problems: Although intelligent computing can effectively solve most problems of large-scale problems [208, 209], there are still some shortcomings. The paper [210] proposed a divide-and-conquer method of intelligent computing, which improves the algorithm's solving ability. The paper [211], based on three strategies (adversarial learning, smoothing techniques, parallelization) improved the performance of the harris hawks optimization algorithm. The parallelization here is different from the previous one. It first sorts the entire population and takes the part with better results as the initialization population. The parallelization here is different from the previous one. It first sorts the entire population and takes the part with better results as the initialization population. The populations are then grouped, one using DE to update and the other HHO to update individuals. Finally, experiments are carried out on the re-entry orbit problem of reusable launch vehicles, and the results show that the practicability of the algorithm is strong enough. In [143], a parallel PSO algorithm is proposed based on FPGA. The proposed algorithm solves the problem of robot path planning very well, which is a global problem (relatively large scale).

Neural networks: In recent years, meta-heuristics have been widely used in the optimization of machine learning models. Among them, the most commonly used neural network model is the multi-layer perceptual model. It consists of input layer, output layer, and hidden layer. The paper [212] used the parallel cuckoo optimization algorithm to train neural networks. At the same time, the algorithm is compared with the PSO algorithm, and good results are obtained. The paper [149] taken full advantage of a new algorithm (parallel galaxy group optimization algorithm GSO) to train neural networks. The algorithm is also used to solve churn prediction and predict whether a player can play for several years in a row, showing the algorithm's efficiency through various proofs.

Atomic potential fit: Since there are various potential energies between atoms, calculating the reduction of the potential energy between atoms is a topic of research. The paper [213] combined PSO with the global optimization algorithm and introduced a multi-dimensional search, which can effectively solve the problem of reducing atomic potential energy.

Image segmentation: Image segmentation is the most basic operation in the field of computer vision, and it has been widely used in various fields, such as medicine, transportation, and surveillance. For example, the paper [188] regarded the image as a two-dimensional grayscale image and used a parallel compact differential evolution algorithm to solve the problem of image threshold segmentation. The paper [190] used the minimum crossing threshold method to segment complex images. The threshold is the region boundary that divides the image into multiple valid parts.

Path planning: Path planning is one of the research hotspots in the field of automotive engineering. In path planning, various obstacles may be encountered. In path planning, various obstacles may be encountered. Then, how to find a suitable solution becomes the topic of research. For example, the paper [198] used an improved Archimedes optimization algorithm to plan robot paths. They are considering the situation with obstacles, finding the optimal movement path without encountering obstacles. The paper [191] planned the path that restricts the vehicle and also ensures the carrying capacity of the vehicle. The paper [147] used a fine-grained model to parallelize GA, which is then applied to route planning in transportation and logistics.

6. **Conclusions.** In recent years, there has been increasing research on meta-heuristic algorithms. This paper begins with an introduction to meta-heuristic algorithms and divides them into four categories based on their origins: human-based, physics-based, swarm intelligence, and evolutionary algorithms. Secondly, the improved methods of meta-heuristic algorithms are introduced, mainly including parallel, binary, self-adaptive, compact, hybrid algorithm, multi-objective, discretization, and opposition-based learning. Among them, the parallel improvement strategy has obvious advantages. This paper divides parallelism into two types: true parallelism and multi-grouping. Regarding the real parallel metaheuristic algorithm, the paper mainly introduces two parallel platforms (GPU and FPGA), four parallelization models (master-slave model, coarse-grained model, fine-grained model, and hybrid model), and four kinds of communication modes (star, mobile, broadcast and diffuse). At the same time, the parallel meta-heuristic algorithm is classified according to the algorithm, and the application of parallel computing in the meta-heuristic algorithm is introduced in detail. Since parallel meta-heuristics are widely used in various fields, the paper concludes with an overview of the applications in the surveyed literature. In future work, various parallelization models can be combined with meta-heuristic algorithms to apply to multi-objective problems, effectively improving efficiency and performance.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (61872085), the Natural Science Foundation of Fujian Province, China (2018J01638) and the Fujian Provincial Department of Science and Technology, China (2018Y3001).

REFERENCES

[1] D. Oliva, E. H. Houssein, and S. Hinojosa, *Metaheuristics in Machine Learning: Theory and Applications*. Springer, 2021.

[2] J. M.-T. Wu, L. Sun, G. Srivastava, and J. C.-W. Lin, "A novel synergetic lstm-ga stock trading suggestion system in internet of things," *Mobile Information Systems*, vol. 2021, 6706345, 2021.

[3] J. H. Holland, "Genetic algorithms and the optimal allocation of trials," *SIAM Journal on Computing*, vol. 2, no. 2, pp. 88–105, 1973.

[4] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, "Application of quantum genetic optimization of lvq neural network in smart city traffic network prediction," *IEEE Access*, vol. 8, pp. 104 555–104 564, 2020.

[5] A. Ahrari and A. A. Atai, "Grenade explosion method—a novel tool for optimization of multimodal functions," *Applied Soft Computing*, vol. 10, no. 4, pp. 1132–1140, 2010.

[6] S. Mirjalili, "Sca: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.

[7] R. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 7, no. 1, pp. 19–34, 2016.

[8] A. Colorni, M. Dorigo, V. Maniezzo, D. d. Elettronica, and P. d. Milano, "Distributed optimization by ant colonies," in *Proceedings of the First European Conference on Artificial Life*, vol. 142. Paris, France, 1991, pp. 134–142.

[9] J. M.-T. Wu, J. Zhan, and J. C.-W. Lin, "An aco-based approach to mine high-utility itemsets," *Knowledge-Based Systems*, vol. 116, pp. 102–113, 2017.
[10] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: theory and application," *Advances in Engineering Software*, vol. 105, pp. 30–47, 2017.
[11] Z. Meng, J.-S. Pan, and H. Xu, "Quasi-affine transformation evolutionary (quatre) algorithm: A cooperative swarm based algorithm for global optimization," *Knowledge-Based Systems*, vol. 109, pp. 104–121, 2016.
[12] P.-C. Song, S.-C. Chu, J.-S. Pan, and H. Yang, "Simplified phasmatodea population evolution algorithm for optimization," *Complex & Intelligent Systems*, pp. 1–19, 2021.
[13] G. Venter and J. Sobieszczanski-Sobieski, "Parallel particle swarm optimization algorithm accelerated by asynchronous evaluations," *Journal of Aerospace Computing, Information, and Communication*, vol. 3, no. 3, pp. 123–137, 2006.
[14] B. Cao, J. Zhao, Z. Lv, X. Liu, S. Yang, X. Kang, and K. Kang, "Distributed parallel particle swarm optimization for multi-objective and many-objective large-scale optimization," *IEEE Access*, vol. 5, pp. 8214–8221, 2017.
[15] S.-C. Chu, J. F. Roddick, and J.-S. Pan, "A parallel particle swarm optimization algorithm with communication strategies," *Journal of Information Science and Engineering*, vol. 21, no. 4, p. 9, 2005.
[16] S.-C. Chu and J.-S. Pan, "Intelligent parallel particle swarm optimization algorithms," in *Parallel Evolutionary Computations*. Springer, 2006, pp. 159–175.
[17] S.-C. Chu, J. F. Roddick, J.-S. Pan, and C.-J. Su, "Parallel ant colony systems," in *International Symposium on Methodologies for Intelligent Systems*. Springer, 2003, pp. 279–284.
[18] A. De Campos Jr, A. T. Pozo, and E. P. Duarte Jr, "Parallel multi-swarm pso strategies for solving many objective optimization problems," *Journal of Parallel and Distributed Computing*, vol. 126, pp. 13–33, 2019.
[19] Y. Sun, J.-S. Pan, P. Hu, and S.-C. Chu, "Enhanced equilibrium optimizer algorithm applied in job shop scheduling problem," *Journal of Intelligent Manufacturing*, pp. 1–27, 2022.
[20] S. Binitha and S. S. Sathya, "A survey of bio inspired optimization algorithms," *International Journal of Soft Computing and Engineering*, vol. 2, no. 2, pp. 137–151, 2012.
[21] L. Kang, R.-S. Chen, N. Xiong, Y.-C. Chen, Y.-X. Hu, and C.-M. Chen, "Selecting hyper-parameters of gaussian process regression based on non-inertial particle swarm optimization in internet of things," *IEEE Access*, vol. 7, pp. 59 504–59 513, 2019.
[22] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.
[23] A. Kaveh and N. Farhoudi, "A new optimization method: Dolphin echolocation," *Advances in Engineering Software*, vol. 59, pp. 53–70, 2013.
[24] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *International Symposium on Stochastic Algorithms*. Springer, 2009, pp. 169–178.
[25] S. Kaur, L. K. Awasthi, A. Sangal, and G. Dhiman, "Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization," *Engineering Applications of Artificial Intelligence*, vol. 90, p. 103541, 2020.
[26] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
[27] X.-l. Li, "An optimizing method based on autonomous animats: fish-swarm algorithm," *Systems Engineering-Theory & Practice*, vol. 22, no. 11, pp. 32–38, 2002.
[28] D. Dasgupta, *Artificial immune systems and their applications*. Springer Science & Business Media, 2012.
[29] M. M. Eusuff and K. E. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," *Journal of Water Resources Planning and Management*, vol. 129, no. 3, pp. 210–225, 2003.
[30] S.-C. Chu, P.-W. Tsai, and J.-S. Pan, "Cat swarm optimization," in *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2006, pp. 854–858.
[31] L. Kong, C.-M. Chen, H.-C. Shih, C.-W. Lin, B.-Z. He, and J.-S. Pan, "An energy-aware routing protocol using cat swarm optimization for wireless sensor networks," in *Advanced Technologies, Embedded and Multimedia for Human-Centric Computing*. Springer, 2014, pp. 311–318.

[32] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[33] J.-S. Pan, P.-W. Tsai, and Y.-B. Liao, "Fish migration optimization based on the fishy biology," in *2010 Fourth International Conference on Genetic and Evolutionary Computing*. IEEE, 2010, pp. 783–786.

[34] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, no. 1, pp. 17–35, 2013.

[35] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[36] J. R. Koza and J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*. MIT press, 1992, vol. 1.

[37] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[38] U. Premaratne, J. Samarabandu, and T. Sidhu, "A new biologically inspired optimization algorithm," in *2009 International Conference on Industrial and Information Systems (ICIIS)*. IEEE, 2009, pp. 279–284.

[39] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[40] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Applied Intelligence*, vol. 51, no. 3, pp. 1531–1551, 2021.

[41] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm," *Future Generation Computer Systems*, vol. 101, pp. 646–667, 2019.

[42] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowledge-Based Systems*, vol. 191, p. 105190, 2020.

[43] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "Gsa: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[44] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: charged system search," *Acta Mechanica*, vol. 213, no. 3, pp. 267–289, 2010.

[45] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition," in *2007 IEEE Congress on Evolutionary Computation*. IEEE, 2007, pp. 4661–4667.

[46] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[47] S. H. S. Moosavi and V. K. Bardsiri, "Poor and rich optimization algorithm: A new human-based and multi populations algorithm," *Engineering Applications of Artificial Intelligence*, vol. 86, pp. 165–181, 2019.

[48] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Human mental search: a new population-based metaheuristic optimization algorithm," *Applied Intelligence*, vol. 47, no. 3, pp. 850–887, 2017.

[49] A. Shabani, B. Asgarian, S. A. Gharebaghi, M. A. Salido, and A. Giret, "A new optimization algorithm based on search and rescue operations," *Mathematical Problems in Engineering*, vol. 2019, 2482543, 2019.

[50] P. Hu, J.-S. Pan, and S.-C. Chu, "Improved binary grey wolf optimizer and its application for feature selection," *Knowledge-Based Systems*, vol. 195, p. 105746, 2020.

[51] Z.-G. Du, J.-S. Pan, S.-C. Chu, and Y.-J. Chiu, "Improved binary symbiotic organism search algorithm with transfer functions for feature selection," *IEEE Access*, vol. 8, pp. 225 730–225 744, 2020.

[52] K. Deb and H.-G. Beyer, "Self-adaptive genetic algorithms with simulated binary crossover," *Evolutionary Computation*, vol. 9, no. 2, pp. 197–221, 2001.

[53] Y. Xue, J. Jiang, B. Zhao, and T. Ma, "A self-adaptive artificial bee colony algorithm based on global best for global optimization," *Soft Computing*, vol. 22, no. 9, pp. 2935–2952, 2018.

[54] A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi, "Firefly algorithm with chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 1, pp. 89–98, 2013.

[55] Z. Meng, J.-S. Pan, and K.-K. Tseng, "Pade: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization," *Knowledge-Based Systems*, vol. 168, pp. 80–99, 2019.

[56] F. Neri, G. Iacca, and E. Mininno, "Compact optimization," in *Handbook of Optimization.* Springer, 2013, pp. 337–364.

[57] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 287–297, 1999.

[58] E. Mininno, F. Neri, F. Cupertino, and D. Naso, "Compact differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 32–54, 2011.

[59] F. Neri, E. Mininno, and G. Lacca, "Compact particle swarm optimization," *Information Sciences*, vol. 239, pp. 96–121, 2013.

[60] J.-S. Pan, P.-C. Song, S.-C. Chu, and Y.-J. Peng, "Improved compact cuckoo search algorithm applied to location of drone logistics hub," *Mathematics*, vol. 8, no. 3, 333, 2020.

[61] T.-T. Nguyen, J.-S. Pan, and T.-K. Dao, "A compact bat algorithm for unequal clustering in wireless sensor networks," *Applied Sciences-Basel*, vol. 9, no. 10, 1973, 2019.

[62] K. Deb, "Multi-objective optimization," in *Search Methodologies.* Springer, 2014, pp. 403–449.

[63] S. Mirjalili, P. Jangir, and S. Saremi, "Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems," *Applied Intelligence*, vol. 46, no. 1, pp. 79–95, 2017.

[64] S. Z. Mirjalili, S. Mirjalili, S. Saremi, H. Faris, and I. Aljarah, "Grasshopper optimization algorithm for multi-objective optimization problems," *Applied Intelligence*, vol. 48, no. 4, pp. 805–820, 2018.

[65] P. K. Tripathi, S. Bandyopadhyay, and S. K. Pal, "Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients," *Information Sciences*, vol. 177, no. 22, pp. 5033–5049, 2007.

[66] K. Dao, T.-S. Pan, and J.-S. Pan, "A multi-objective optimal mobile robot path planning based on whale optimization algorithm," in *13th IEEE International Conference on Signal Processing (ICSP)*, ser. International Conference on Signal Processing, 2016, Conference Proceedings, pp. 337–342.

[67] J. Liu and J. Liu, "Applying multi-objective ant colony optimization algorithm for solving the unequal area facility layout problems," *Applied Soft Computing*, vol. 74, pp. 167–189, 2019.

[68] J.-S. Pan, S.-C. Chu, T.-K. Dao, and T.-G. Ngo, "A multi-objective ions motion optimization for robot path planning," in *International Conference on Engineering Research and Applications.* Springer, 2018, Conference Proceedings, pp. 46–54.

[69] J.-S. Pan, N. Liu, S.-C. Chu, and T. Lai, "An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems," *Information Sciences*, vol. 561, pp. 304–325, 2021.

[70] P. Hu, J.-S. Pan, S.-C. Chu, Q.-W. Chai, T. Liu, and Z.-C. Li, "New hybrid algorithms for prediction of daily load of power network," *Applied Sciences-Basel*, vol. 9, no. 21, 4514, 2019.

[71] S.-C. Chu, Z.-G. Du, and J.-S. Pan, "Discrete fish migration optimization for traveling salesman problem," *Data Science and Pattern Recognition*, vol. 4, no. 2, pp. 1–18, 2020.

[72] M. K. Sayadi, A. Hafezalkotob, and S. G. J. Naini, "Firefly-inspired algorithm for discrete optimization problems: An application to manufacturing cell formation," *Journal of Manufacturing Systems*, vol. 32, no. 1, pp. 78–84, 2013.

[73] Q.-K. Pan, M. F. Tasgetiren, P. N. Suganthan, and T. J. Chua, "A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem," *Information Sciences*, vol. 181, no. 12, pp. 2455–2468, 2011.

[74] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, vol. 1. IEEE, 2005, Conference Proceedings, pp. 695–701.

[75] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, "Opposition-based differential evolution algorithms," in *IEEE International Conference on Evolutionary Computation.* IEEE, 2006, Conference Proceedings, pp. 2010–2017.

[76] M. Abd Elaziz, D. Oliva, and S. Xiong, "An improved opposition-based sine cosine algorithm for global optimization," *Expert Systems with Applications*, vol. 90, pp. 484–500, 2017.

[77] A. A. Ewees, M. Abd Elaziz, and E. H. Houssein, "Improved grasshopper optimization algorithm using opposition-based learning," *Expert Systems with Applications*, vol. 112, pp. 156–172, 2018.

[78] M. J. Flynn, "Some computer organizations and their effectiveness," *IEEE Transactions on Computers*, vol. 100, no. 9, pp. 948–960, 1972.

[79] A. Jimeno-Morenilla, J.-L. Sanchez-Romero, H. Migallon, and H. Mora-Mora, "Jaya optimization algorithm with gpu acceleration," *The Journal of Supercomputing*, vol. 75, no. 3, pp. 1094–1106, 2019.

[80] M. Lin, Y. Zhong, and R. Chen, "Graphic process units-based chicken swarm optimization algorithm for function optimization problems," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 8, p. e5953, 2021.

[81] M. Pietroń, A. Byrski, and M. Kisiel-Dorohinicki, "Leveraging heterogeneous parallel platform in solving hard discrete optimization problems with metaheuristics," *Journal of Computational Science*, vol. 18, pp. 59–68, 2017.

[82] E. Rios, L. S. Ochi, C. Boeres, V. N. Coelho, I. M. Coelho, and R. Farias, "Exploring parallel multi-gpu local search strategies in a metaheuristic framework," *Journal of Parallel and Distributed Computing*, vol. 111, pp. 39–55, 2018.

[83] R. Skinderowicz, "The gpu-based parallel ant colony system," *Journal of Parallel and Distributed Computing*, vol. 98, pp. 48–60, 2016.

[84] P. Krömer, J. Platoš, and V. Snášel, "Nature-inspired meta-heuristics on modern gpus: state of the art and brief survey of selected algorithms," *International Journal of Parallel Programming*, vol. 42, no. 5, pp. 681–709, 2014.

[85] P. Szwed and W. Chmiel, "Multi-swarm pso algorithm for the quadratic assignment problem: a massive parallel implementation on the opencl platform," *arXiv preprint arXiv:1504.05158*, 2015.

[86] G.-H. Luo, S.-K. Huang, Y.-S. Chang, and S.-M. Yuan, "A parallel bees algorithm implementation on gpu," *Journal of Systems Architecture*, vol. 60, no. 3, pp. 271–279, 2014.

[87] A. Ouyang, X. Peng, Q. Wang, Y. Wang, and T. K. Truong, "A parallel improved iwo algorithm on gpu for solving large scale global optimization problems," *Journal of Intelligent & Fuzzy Systems*, vol. 31, no. 2, pp. 1041–1051, 2016.

[88] H. Rico-Garcia, J.-L. Sanchez-Romero, A. Jimeno-Morenilla, H. Migallon-Gomis, H. Mora-Mora, and R. V. Rao, "Comparison of high performance parallel implementations of tlbo and jaya optimization methods on manycore gpu," *IEEE Access*, vol. 7, pp. 133 822–133 831, 2019.

[89] J. M. Cecilia, A. Llanes, J. L. Abellan, J. Gomez-Luna, L.-W. Chang, and W.-M. W. Hwu, "High-throughput ant colony optimization on graphics processing units," *Journal of Parallel and Distributed Computing*, vol. 113, pp. 261–274, 2018.

[90] X. Zou, F. Tao, P. Jiang, S. Gu, K. Qiao, Y. Zuo, and L. Xu, "A new approach for data processing in supply chain network based on fpga," *The International Journal of Advanced Manufacturing Technology*, vol. 84, no. 1, pp. 249–260, 2016.

[91] H. Lee, K. Kim, Y. Kwon, and E. Hong, "Real-time particle swarm optimization on fpga for the optimal message-chain structure," *Electronics*, vol. 7, no. 11, p. 274, 2018.

[92] H.-C. Huang, "Fpga-based parallel metaheuristic pso algorithm and its application to global path planning for autonomous robot navigation," *Journal of Intelligent & Robotic Systems*, vol. 76, no. 3, pp. 475–488, 2014.

[93] X. Zou, L. Wang, Y. Tang, Y. Liu, S. Zhan, and F. Tao, "Parallel design of intelligent optimization algorithm based on fpga," *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9, pp. 3399–3412, 2018.

[94] S. S. Roy, F. Turan, K. Jarvinen, F. Vercauteren, and I. Verbauwhede, "Fpga-based high-performance parallel architecture for homomorphic computing on encrypted data," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2019, pp. 387–398.

[95] J. Cao, H. Cui, H. Shi, and L. Jiao, "Big data: A parallel particle swarm optimization-back-propagation neural network algorithm based on mapreduce," *Plos One*, vol. 11, no. 6, p. e0157551, 2016.

[96] E. Cantú-Paz, "A survey of parallel genetic algorithms," *Calculateurs Paralleles, Reseaux Et Systems Repartis*, vol. 10, no. 2, pp. 141–171, 1998.

[97] E. Cantu-Paz, *Efficient and accurate parallel genetic algorithms.* Springer Science & Business Media, 2000, vol. 1.

[98] Madhuri and K. Deep, "A state-of-the-art review of population-based parallel meta-heuristics," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. IEEE, 2009, pp. 1604–1607.

[99] M. Pedemonte, S. Nesmachnow, and H. Cancela, "A survey on parallel ant colony optimization," *Applied Soft Computing*, vol. 11, no. 8, pp. 5181–5197, 2011.

[100] E. Alba, G. Luque, and S. Nesmachnow, "Parallel metaheuristics: recent advances and new trends," *International Transactions in Operational Research*, vol. 20, no. 1, pp. 1–48, 2013.

[101] S. Lalwani, H. Sharma, S. C. Satapathy, K. Deep, and J. C. Bansal, "A survey on parallel particle swarm optimization algorithms," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 2899–2923, 2019.

[102] T. Dokeroglu, S. Pehlivan, and B. Avenoglu, "Robust parallel hybrid artificial bee colony algorithms for the multi-dimensional numerical optimization," *The Journal of Supercomputing*, vol. 76, no. 9, pp. 7026–7046, 2020.

[103] Ş. Gülcü and H. Kodaz, "A novel parallel multi-swarm algorithm based on comprehensive learning particle swarm optimization," *Engineering Applications of Artificial Intelligence*, vol. 45, pp. 33–45, 2015.

[104] T. Dokeroglu and A. Cosar, "Optimization of one-dimensional bin packing problem with island parallel grouping genetic algorithms," *Computers & Industrial Engineering*, vol. 75, pp. 176–186, 2014.

[105] E. Alba and M. Tomassini, "Parallelism and evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 443–462, 2002.

[106] G. Luque, E. Alba, and B. Dorronsoro, "An asynchronous parallel implementation of a cellular genetic algorithm for combinatorial optimization," in *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, 2009, pp. 1395–1402.

[107] L. Li, F. F. Zhang, X. Chu, and B. Niu, "Modified brain storm optimization algorithms based on topology structures," in *International Conference on Swarm Intelligence*, 2016, Conference Proceedings, pp. 408–415.

[108] X. Lai and Y. Zhou, "An adaptive parallel particle swarm optimization for numerical optimization problems," *Neural Computing and Applications*, vol. 31, no. 10, pp. 6449–6467, 2019.

[109] M. S. Turgut, O. E. Turgut, and D. T. Eliiyi, "Island-based crow search algorithm for solving optimal control problems," *Applied Soft Computing*, vol. 90, p. 106170, 2020.

[110] W. Li and G.-G. Wang, "Elephant herding optimization using dynamic topology and biogeography-based optimization based on learning for numerical optimization," *Engineering with Computers*, pp. 1–29, 2021.

[111] X. Yuan, T. Zhang, Y. Xiang, and X. Dai, "Parallel chaos optimization algorithm with migration and merging operation," *Applied Soft Computing*, vol. 35, pp. 591–604, 2015.

[112] Q. Liu, T. Li, C. Tang, Q. Liu, J. Zhu, X. Ding, and J. Wu, "Two phase parallel particle swarm algorithm based on regional and social study of object optimization," in *Third International Conference on Natural Computation (ICNC 2007)*, vol. 3. IEEE, 2007, Conference Proceedings, pp. 827–831.

[113] N. Lynn, M. Z. Ali, and P. N. Suganthan, "Population topologies for particle swarm optimization and differential evolution," *Swarm and Evolutionary Computation*, vol. 39, pp. 24–35, 2018.

[114] H. Migallón, A. Jimeno-Morenilla, J.-L. Sánchez-Romero, and A. Belazi, "Efficient parallel and fast convergence chaotic jaya algorithms," *Swarm and Evolutionary Computation*, vol. 56, p. 100698, 2020.

[115] Z. He, H. Peng, J. Chen, C. Deng, and Z. Wu, "A spark-based differential evolution with grouping topology model for large-scale global optimization," *Cluster Computing*, vol. 24, pp. 515–535, 2021.

[116] J. Shi and Q. Zhang, "A new cooperative framework for parallel trajectory-based metaheuristics," *Applied Soft Computing*, vol. 65, pp. 374–386, 2018.

[117] T. Kucukyilmaz and H. E. Kiziloz, "Cooperative parallel grouping genetic algorithm for the one-dimensional bin packing problem," *Computers & Industrial Engineering*, vol. 125, pp. 157–170, 2018.

[118] J. Torlapati and T. P. Clement, "Using parallel genetic algorithms for estimating model parameters in complex reactive transport problems," *Processes*, vol. 7, no. 10, p. 640, 2019.

[119] D. Mu, C. Wang, F. Zhao, and J. W. Sutherland, "Solving vehicle routing problem with simultaneous pickup and delivery using parallel simulated annealing algorithm," *International Journal of Shipping and Transport Logistics*, vol. 8, no. 1, pp. 81–106, 2016.

[120] M. Sambridge, "A parallel tempering algorithm for probabilistic sampling and multimodal optimization," *Geophysical Journal International*, vol. 196, no. 1, pp. 357–374, 2014.

[121] X. Yuan, T. Zhang, X. Dai, and L. Wu, "Master–slave model-based parallel chaos optimization algorithm for parameter identification problems," *Nonlinear Dynamics*, vol. 83, no. 3, pp. 1727–1741, 2016.

[122] J. Gou, F. Wang, and W. Luo, "Mining fuzzy association rules based on parallel particle swarm optimization algorithm," *Intelligent Automation & Soft Computing*, vol. 21, no. 2, pp. 147–162, 2015.

[123] J.-Y. Li, Z.-H. Zhan, R.-D. Liu, C. Wang, S. Kwong, and J. Zhang, "Generation-level parallelism for evolutionary computation: A pipeline-based parallel particle swarm optimization," *IEEE Transactions on Cybernetics*, vol. 51, no. 10, pp. 4848–4859, 2020.

[124] Z. Liu, Z. Li, P. Zhu, and W. Chen, "A parallel boundary search particle swarm optimization algorithm for constrained optimization problems," *Structural and Multidisciplinary Optimization*, vol. 58, no. 4, pp. 1505–1522, 2018.

[125] L. Asadzadeh, "A parallel artificial bee colony algorithm for the job shop scheduling problem with a dynamic migration strategy," *Computers & Industrial Engineering*, vol. 102, pp. 359–367, 2016.

[126] S. Li, W. Li, H. Zhang, and Z. Wang, "Research and implementation of parallel artificial bee colony algorithm based on ternary optical computer," *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, vol. 60, no. 4, pp. 422–431, 2019.

[127] R. S. Parpinelli, C. M. V. Benitez, and H. S. Lopes, "Parallel approaches for the artificial bee colony algorithm," in *Handbook of Swarm Intelligence*. Springer, 2011, pp. 329–345.

[128] Z.-h. Jia, Y.-l. Zhang, J. Y.-T. Leung, and K. Li, "Bi-criteria ant colony optimization algorithm for minimizing makespan and energy consumption on parallel batch machines," *Applied Soft Computing*, vol. 55, pp. 226–237, 2017.

[129] H. Liu, P. Li, and Y. Wen, "Parallel ant colony optimization algorithm," in *2006 6th World Congress on Intelligent Control and Automation*, vol. 1. IEEE, 2006, pp. 3222–3226.

[130] Ş. Gülcü, M. Mahi, Ö. K. Baykan, and H. Kodaz, "A parallel cooperative hybrid method based on ant colony optimization and 3-opt algorithm for solving traveling salesman problem," *Soft Computing*, vol. 22, no. 5, pp. 1669–1685, 2018.

[131] Q. Wang and H. Chen, "Optimization of parallel random forest algorithm based on distance weight," *Journal of Intelligent & Fuzzy Systems*, vol. 39, no. 2, pp. 1951–1963, 2020.

[132] Z. Yanpeng, "Hybrid kernel extreme learning machine for evaluation of athletes' competitive ability based on particle swarm optimization," *Computers & Electrical Engineering*, vol. 73, pp. 23–31, 2019.

[133] A. Lančinskas, P. M. Ortigosa, and J. Žilinskas, "Parallel optimization algorithm for competitive facility location," *Mathematical Modelling and Analysis*, vol. 20, no. 5, pp. 619–640, 2015.

[134] Z. Daoqing and J. Mingyan, "Parallel discrete lion swarm optimization algorithm for solving traveling salesman problem," *Journal of Systems Engineering and Electronics*, vol. 31, no. 4, pp. 751–760, 2020.

[135] Y. Laili, L. Zhang, and Y. Li, "Parallel transfer evolution algorithm," *Applied Soft Computing*, vol. 75, pp. 686–701, 2019.

[136] W. Li, Y. Bi, X. Zhu, C.-a. Yuan, and X.-b. Zhang, "Hybrid swarm intelligent parallel algorithm research based on multi-core clusters," *Microprocessors and Microsystems*, vol. 47, pp. 151–160, 2016.

[137] A. Majd, G. Sahebi, M. Daneshtalab, J. Plosila, S. Lotfi, and H. Tenhunen, "Parallel imperialist competitive algorithms," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 7, p. e4393, 2018.

[138] J. Klemsa and J. Řezáč, "Parallel low-memory quasi-newton optimization algorithm for molecular structure," *Chemical Physics Letters*, vol. 584, pp. 10–13, 2013.

[139] H. E. Kiziloz and T. Dokeroglu, "A robust and cooperative parallel tabu search algorithm for the maximum vertex weight clique problem," *Computers & Industrial Engineering*, vol. 118, pp. 54–66, 2018.

[140] Z. Lai, X. Feng, H. Yu, and F. Luo, "A parallel social spider optimization algorithm based on emotional learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 2, pp. 797–808, 2018.

[141] J. Yi, X. Li, C.-H. Chu, and L. Gao, "Parallel chaotic local search enhanced harmony search algorithm for engineering design optimization," *Journal of Intelligent Manufacturing*, vol. 30, no. 1, pp. 405–428, 2019.

[142] L. Hongtao and K. Fengju, "Adaptive chaos parallel clonal selection algorithm for objective optimization in wta application," *Optik*, vol. 127, no. 6, pp. 3459–3465, 2016.

[143] H.-C. Huang, "Fpga-based parallel metaheuristic pso algorithm and its application to global path planning for autonomous robot navigation," *Journal of Intelligent & Robotic Systems*, vol. 76, no. 3, pp. 475–488, 2014.

[144] D. Abramson and J. Abela, *A parallel genetic algorithm for solving the school timetabling problem.* Citeseer, 1991.

[145] E. Alba and J. M. Troya, "Analyzing synchronous and asynchronous parallel distributed genetic algorithms," *Future Generation Computer Systems*, vol. 17, no. 4, pp. 451–465, 2001.

[146] H. Alrezaamiri, A. Ebrahimnejad, and H. Motameni, "Parallel multi-objective artificial bee colony algorithm for software requirement optimization," *Requirements Engineering*, vol. 25, no. 3, pp. 363–380, 2020.

[147] D. I. Arkhipov, D. Wu, T. Wu, and A. C. Regan, "A parallel genetic algorithm framework for transportation planning and logistics management," *IEEE Access*, vol. 8, pp. 106 506–106 515, 2020.

[148] J. Berger and M. Barkaoui, "A parallel hybrid genetic algorithm for the vehicle routing problem with time windows," *Computers & Operations Research*, vol. 31, no. 12, pp. 2037–2053, 2004.

[149] S. Bhardwaj, G. B. D Amali, A. Phadke, K. Umadevi, and P. Balakrishnan, "A new parallel galactic swarm optimization algorithm for training artificial neural networks," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 5, pp. 6691–6701, 2020.

[150] S.-C. Chu, J. F. Roddick, and J.-S. Pan, "Ant colony system with communication strategies," *Information Sciences*, vol. 167, no. 1-4, pp. 63–76, 2004.

[151] T. Dokeroglu, S. Pehlivan, and B. Avenoglu, "Robust parallel hybrid artificial bee colony algorithms for the multi-dimensional numerical optimization," *The Journal of Supercomputing*, pp. 1–21, 2020.

[152] A. Hassanein, M. El-Abd, I. Damaj, and H. U. Rehman, "Parallel hardware implementation of the brain storm optimization algorithm using fpgas," *Microprocessors and Microsystems*, vol. 74, p. 103005, 2020.

[153] N. M. Hijazi, H. Faris, and I. Aljarah, "A parallel metaheuristic approach for ensemble feature selection based on multi-core architectures," *Expert Systems with Applications*, vol. 182, p. 115290, 2021.

[154] H.-C. Lu, F. Hwang, and Y.-H. Huang, "Parallel and distributed architecture of genetic algorithm on apache hadoop and spark," *Applied Soft Computing*, vol. 95, p. 106497, 2020.

[155] H. H. Mehne and S. Mirjalili, "A parallel numerical method for solving optimal control problems based on whale optimization algorithm," *Knowledge-Based Systems*, vol. 151, pp. 114–123, 2018.

[156] M. Mirhosseini, M. Fazlali, H. T. Malazi, S. K. Izadi, and H. Nezamabadi-pour, "Parallel quadrivalent quantum-inspired gravitational search algorithm on a heterogeneous platform for wireless sensor networks," *Computers & Electrical Engineering*, vol. 92, p. 107085, 2021.

[157] J. Pan, F. McInnes, and M. Jack, "Application of parallel genetic algorithm and property of multiple global optima to vq codevector index assignment for noisy channels," *Electronics Letters*, vol. 32, no. 4, pp. 296–297, 1996.

[158] R. M. Rizk-Allah, R. A. El-Sehiemy, and G.-G. Wang, "A novel parallel hurricane optimization algorithm for secure emission/economic load dispatch solution," *Applied Soft Computing*, vol. 63, pp. 206–222, 2018.

[159] B. Shanmugapriya and S. Meera, "A survey of parallel social spider optimization algorithm based on swarm intelligence for high dimensional datasets," *International Journal of Computational Intelligence Research*, vol. 13, no. 9, pp. 2259–2265, 2017.

[160] M. Waintraub, R. Schirru, and C. M. Pereira, "Multiprocessor modeling of parallel particle swarm optimization applied to nuclear engineering problems," *Progress in Nuclear Energy*, vol. 51, no. 6-7, pp. 680–688, 2009.

[161] Y. Zhou, F. He, N. Hou, and Y. Qiu, "Parallel ant colony optimization on multi-core simd cpus," *Future Generation Computer Systems*, vol. 79, pp. 473–487, 2018.

[162] C.-c. Deng, R.-m. Zhu, Y.-x. Li, and B. Xu, "A multi-group parallel genetic algorithm for tsp," *Computer Simulation*, vol. 25, pp. 187–190, 2008.

[163] P.-W. Tsai, J.-S. Pan, S.-M. Chen, B.-Y. Liao, and S.-P. Hao, "Parallel cat swarm optimization," in *2008 International Conference on Machine Learning and Cybernetics*, vol. 6. IEEE, 2008, pp. 3328–3333.

[164] R. Wang, Y. Ru, and Q. Long, "Improved adaptive and multi-group parallel genetic algorithm based on good-point set." *Journal of Software*, vol. 4, no. 4, pp. 348–356, 2009.

[165] X. Zhang, W. Chen, C. Dai, and W. Cai, "Dynamic multi-group self-adaptive differential evolution algorithm for reactive power optimization," *International Journal of Electrical Power & Energy Systems*, vol. 32, no. 5, pp. 351–357, 2010.

[166] L. Ozbakir, A. Baykasoglu, B. Gorkemli, and L. Gorkemli, "Multiple-colony ant algorithm for parallel assembly line balancing problem," *Applied Soft Computing*, vol. 11, no. 3, pp. 3186–3198, 2011.

[167] G. Du, Z. Jiang, X. Diao, and Y. Yao, "Knowledge extraction algorithm for variances handling of cp using integrated hybrid genetic double multi-group cooperative pso and dpso," *Journal of Medical Systems*, vol. 36, no. 2, pp. 979–994, 2012.

[168] A. El Dor, M. Clerc, and P. Siarry, "A multi-swarm pso using charged particles in a partitioned search space for continuous optimization," *Computational Optimization and Applications*, vol. 53, no. 1, pp. 271–295, 2012.

[169] P.-W. Tsai, J.-S. Pan, S.-M. Chen, and B.-Y. Liao, "Enhanced parallel cat swarm optimization based on the taguchi method," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6309–6319, 2012.

[170] Ş. Akpinar and A. Baykasoğlu, "Multiple colony bees algorithm for continuous spaces," *Applied Soft Computing*, vol. 24, pp. 829–841, 2014.

[171] C.-F. Tsai, T.-K. Dao, T.-S. Pan, and J.-F. Chang, "Parallel bat algorithm applied to the economic load dispatch problem," *Journal of Internet Technology*, vol. 17, no. 4, pp. 761–769, 2016.

[172] Z. Chun-na and L. Yi-ran, "Cloud computing scheduling strategy based on multi-group parallel particle swarm optimization," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 12, no. 2, pp. 195–204, 2017.

[173] S. Huang and V. Dinavahi, "Multi-group particle swarm optimisation for transmission expansion planning solution based on lu decomposition," *IET Generation, Transmission & Distribution*, vol. 11, no. 6, pp. 1434–1442, 2017.

[174] N. Tong, Q. Fu, C. Zhong, and P. Wang, "A multi-group firefly algorithm for numerical optimization," in *Journal of Physics: Conference Series*, vol. 887, no. 1. IOP Publishing, 2017, p. 012060.

[175] P.-W. Tsai, T.-T. Nguyen, J.-S. Pan, T.-K. Dao, and W.-M. Zheng, "A parallel optimization algorithm based on communication strategy of pollens and agents," in *Advances in Intelligent Information Hiding and Multimedia Signal Processing*. Springer, 2017, pp. 315–324.

[176] T.-K. Dao, T.-S. Pan, T.-T. Nguyen, and J.-S. Pan, "Parallel bat algorithm for optimizing makespan in job shop scheduling problems," *Journal of Intelligent Manufacturing*, vol. 29, no. 2, pp. 451–462, 2018.

[177] C. Guo, Z. Yang, X. Wu, T. Tan, and K. Zhao, "Application of an adaptive multi-population parallel genetic algorithm with constraints in electromagnetic tomography with incomplete projections," *Applied Sciences-Basel*, vol. 9, no. 13, p. 2611, 2019.

[178] B.-Q. Jiang and J.-S. Pan, "A parallel quasi-affine transformation evolution algorithm for global optimization." *Journal of Network Intelligence*, vol. 4, no. 2, pp. 30–46, 2019.

[179] N. Liu, J.-S. Pan, J. Wang, and T.-T. Nguyen, "An adaptation multi-group quasi-affine transformation evolutionary algorithm for global optimization and its application in node localization in wireless sensor networks," *Sensors*, vol. 19, no. 19, p. 4112, 2019.

[180] Z. Meng, J.-S. Pan, and K.-K. Tseng, "Pade: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization," *Knowledge-Based Systems*, vol. 168, pp. 80–99, 2019.

[181] T.-T. Nguyen, J.-S. Pan, and T.-K. Dao, "An improved flower pollination algorithm for optimizing layouts of nodes in wireless sensor network," *IEEE Access*, vol. 7, pp. 75 985–75 998, 2019.

[182] T.-T. Nguyen, J.-S. Pan, and T.-K. Dao, "A novel improved bat algorithm based on hybrid parallel and compact for balancing an energy consumption problem," *Information*, vol. 10, no. 6, p. 194, 2019.

[183] J.-S. Pan, P. Hu, and S.-C. Chu, "Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power," *Processes*, vol. 7, no. 11, p. 845, 2019.

[184] H. Zhang and X. You, "Multi-population ant colony optimization algorithm based on congestion factor and co-evolution mechanism," *IEEE Access*, vol. 7, pp. 158 160–158 169, 2019.

[185] Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo, and P. Hu, "Quasi-affine transformation evolutionary algorithm with communication schemes for application of rssi in wireless sensor networks," *IEEE Access*, vol. 8, pp. 8583–8594, 2020.

[186] Z. Feng, L. Chen, C.-H. Chen, M. Liu, and M.-e. Yuan, "Motion planning for redundant robotic manipulators using a novel multi-group particle swarm optimization," *Evolutionary Intelligence*, vol. 13, no. 4, pp. 677–686, 2020.

[187] P.-C. Song, J.-S. Pan, and S.-C. Chu, "A parallel compact cuckoo search algorithm for three-dimensional path planning," *Applied Soft Computing*, vol. 94, p. 106443, 2020.

[188] X. Sui, S.-C. Chu, J.-S. Pan, and H. Luo, "Parallel compact differential evolution for optimization applied to image segmentation," *Applied Sciences-Basel*, vol. 10, no. 6, p. 2195, 2020.

[189] N. Suryanto, H. Kang, Y. Kim, Y. Yun, H. T. Larasati, and H. Kim, "A distributed black-box adversarial attack based on multi-group particle swarm optimization," *Sensors*, vol. 20, no. 24, p. 7158, 2020.

[190] X. Wang, J.-S. Pan, and S.-C. Chu, "A parallel multi-verse optimizer for application in multilevel image segmentation," *IEEE Access*, vol. 8, pp. 32 018–32 030, 2020.

[191] Q. Yang, S.-C. Chu, J.-S. Pan, and C.-M. Chen, "Sine cosine algorithm with multigroup and multistrategy for solving cvrp," *Mathematical Problems in Engineering*, vol. 2020, 8184254, 2020.

[192] M. AlShabi, C. Ghenai, M. Bettayeb, F. F. Ahmad, and M. E. H. Assad, "Multi-group grey wolf optimizer (mg-gwo) for estimating photovoltaic solar cell model," *Journal of Thermal Analysis and Calorimetry*, vol. 144, no. 5, pp. 1655–1670, 2021.

[193] F. Fan, S.-C. Chu, J.-S. Pan, Q. Yang, and H. Zhao, "Parallel sine cosine algorithm for the dynamic deployment in wireless sensor networks," *Journal of Internet Technology*, vol. 22, no. 3, pp. 499–512, 2021.

[194] J. Li, M. Gao, J.-S. Pan, and S.-C. Chu, "A parallel compact cat swarm optimization and its application in dv-hop node localization for wireless sensor network," *Wireless Networks*, vol. 27, no. 3, pp. 2081–2101, 2021.

[195] S. Li, X. You, and S. Liu, "Multiple ant colony optimization using both novel lstm network and adaptive tanimoto communication strategy," *Applied Intelligence*, pp. 1–21, 2021.

[196] J.-S. Pan, J. Shan, S.-G. Zheng, S.-C. Chu, and C.-K. Chang, "Wind power prediction based on neural network with optimization of adaptive multi-group salp swarm algorithm," *Cluster Computing*, pp. 1–16, 2021.

[197] C. Wang, J. Li, H. Rao, A. Chen, J. Jiao, N. Zou, and L. Gu, "Multi-objective grasshopper optimization algorithm based on multi-group and co-evolution." *Mathematical Biosciences and Engineering: MBE*, vol. 18, no. 3, pp. 2527–2561, 2021.

[198] R.-B. Wang, W.-F. Wang, L. Xu, J.-S. Pan, and S.-C. Chu, "An adaptive parallel arithmetic optimization algorithm for robot path planning," *Journal of Advanced Transportation*, vol. 2021, 3606895, 2021.

[199] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Springer, 2010, pp. 65–74.

[200] M. B. Shafik, H. Chen, G. I. Rashed, and R. A. El-Sehiemy, "Adaptive multi objective parallel seeker optimization algorithm for incorporating tcsc devices into optimal power flow framework," *IEEE Access*, vol. 7, pp. 36 934–36 947, 2019.

[201] H. Zhang and Q. Hui, "Parallel multiagent coordination optimization algorithm: implementation, evaluation, and applications," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 984–995, 2016.

[202] A. Panda and S. Pani, "An orthogonal parallel symbiotic organism search algorithm embodied with augmented lagrange multiplier for solving constrained optimization problems," *Soft Computing*, vol. 22, no. 8, pp. 2429–2447, 2018.

[203] H. Migallón, A. Jimeno-Morenilla, H. Rico, J. Sánchez-Romero, and A. Belazi, "Multi-level parallel chaotic jaya optimization algorithms for solving constrained engineering design problems," *The Journal of Supercomputing*, vol. 77, no. 11, pp. 12 280–12 319, 2021.

[204] I. Diarrassouba, M. K. Labidi, and A. R. Mahjoub, "A parallel hybrid optimization algorithm for some network design problems," *Soft Computing*, vol. 23, no. 6, pp. 1947–1964, 2019.

[205] G. T. Pulido and C. A. C. Coello, "A constraint-handling mechanism for particle swarm optimization," in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)*, vol. 2. IEEE, 2004, pp. 1396–1403.

[206] B. Yang, K.-U. Bletzinger, Q. Zhang, and Z. Zhou, "Frame structural sizing and topological optimization via a parallel implementation of a modified particle swarm algorithm," *KSCE Journal of Civil Engineering*, vol. 17, no. 6, pp. 1359–1370, 2013.

[207] A. Borisov and J. Siek, "A parallel optimization algorithm for predictive control of marine vessel," in *Journal of Physics: Conference Series*, vol. 1333, no. 4. IOP Publishing, 2019, p. 042004.

[208] S. Santander-Jimenez and M. A. Vega-Rodriguez, "Parallel multiobjective metaheuristics for inferring phylogenies on multicore clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 6, pp. 1678–1692, 2014.

[209] H. F. Sheikh, I. Ahmad, and D. Fan, "An evolutionary technique for performance-energy-temperature optimized scheduling of parallel tasks on multi-core processors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 3, pp. 668–681, 2015.

[210] P. Yang, K. Tang, and X. Yao, "A parallel divide-and-conquer-based evolutionary algorithm for large-scale optimization," *IEEE Access*, vol. 7, pp. 163 105–163 118, 2019.

[211] Y. Su, Y. Dai, and Y. Liu, "A hybrid parallel harris hawks optimization algorithm for reusable launch vehicle reentry trajectory optimization with no-fly zones," *Soft Computing*, vol. 25, no. 23, pp. 14 597–14 617, 2021.

[212] A. A. Kawam and N. Mansour, "Metaheuristic optimization algorithms for training artificial neural networks," *International Journal of Computer and Information Technology*, vol. 1, no. 2, pp. 156–161, 2012.

[213] C. Voglis, P. E. Hadjidoukas, D. G. Papageorgiou, and I. E. Lagaris, "A parallel hybrid optimization algorithm for fitting interatomic potentials," *Applied Soft Computing*, vol. 13, no. 12, pp. 4481–4492, 2013.