**International Journal of Intelligent Information Technologies**
Volume 20 • Issue 1

# Distributed Data Real-Time Transaction Calculation Based on Collaborative Optimization and Multi-Objective Genetic Algorithm

Li Liao, Chongqing City Vocational College, China*


## ABSTRACT

Based on the above situation, this article elaborated on the methods that should be used to calculate the multi-information genetic algorithm (GA) in the current situation. The article mainly compared the non-dominated sorting genetic algorithm-II (NSGA-II) and multi-objective particle swarm optimization (MOPSO) with elite strategy. By measuring the solving speed and quality of the two algorithms, it was found that the NSGA-II had a greater advantage. Based on the NSGA-II, optimization processing was carried out. The NSGA-II was compared before and after optimization. After analyzing 48 data samples, it was found that the results of the NSGA-II before and after optimization showed that the algorithm tended to be more stable after optimization, thus indicating that the improved data was more accurate. The results indicated that the NSGA-II was necessary for its improvement, and its results were also reasonable.



## KEYWORDS

Distributed Data, MOPSO Algorithm, Multi-Objective Genetic Algorithm, NSGA-II Algorithm, Real-Time Transaction Computing


## 1. INTRODUCTION

The article mainly compares Non-dominated Sorting Genetic Algorithm-II (NSGA-II) and Multi-Objective Particle Swarm Optimization (MOPSO), the main purpose of which is to quickly generate value calculations for items through real-time transaction calculations and improve the efficiency of workers. About The NSGA-II with elite strategy can be implemented in various fields, and this algorithm is also commonly used to optimize data. Ransikarbum K conducted research and analysis on disaster relief allocation by utilizing the NSGA-II. He combined the NSGA-II with Pareto frontier analysis and proposed a comparison between the NSGA-II and the exact method. He also evaluated the results based on ultra large scale technology and computational time to demonstrate the efficiency of the NSGA-II (Ransikarbum & Mason, 2022). Babaeinesami A believed that the configuration

DOI: 10.4018/IJIIT.333632
*Corresponding Author

of the supply chain network was a key issue. This provided potential factors for reducing costs and improving service quality. Babaeinesami A designed a sales network according to customer requirements to minimize overall costs and CO2 emissions. In response to the complexity of this problem, he designed an adaptive NSGA-II. According to the constraint method, it was evaluated. He used design methods to adjust parameters to enhance the performance of the algorithm (Babaeinesami et al. 2022). It was also found that the solution time of the adaptive NSGA-II method was better than that of the constraint method.

Liu S believes that with the wide application of cloud computing technology, the services provided by cloud systems are becoming more and more diverse, so these systems are required to solve highly diverse and complex tasks. Liu S proposed a new Non-dominated Sorting Genetic Algorithm (NAGA). The improved crossover and mutation genetic operator enhances the optimization ability of the algorithm and greatly reduces the probability of the algorithm falling into local optimal solution. In addition, an upgraded fitness operator method was proposed by considering the main factors affecting the quality of service, such as task completion time, system load and network bandwidth. Finally, the improved genetic algorithm is proposed to effectively optimize the resource scheduling strategy, shorten the task completion time, promote system load balancing, and improve the quality of system service (Liu & Wang, 2020).

Zhou J carried out optimization analysis and dynamic parameter optimization through multi-objective collaborative optimization algorithm, aiming at the curve negotiation characteristics, comfort and tire eccentric wear of monorails. Zhou J finally concluded that the multi-objective cooperative optimization algorithm can successfully perform multi-objective optimization of monorail vehicles. Curve negotiation performance and comfort are improved, and optimized dynamic parameters reduce tire wear. In particular, a consistent objective function transformation strategy for multi-objective nonlinear convergence is proposed (Zhou et al. 2020).

Currently, there are many problems in the research of multi-objective optimization. Therefore, this article was based on an improved NSGA-II with elite strategy and multi-objective optimization method. By setting parameters for comparison, the pre optimization and post optimization were compared using parameters, and it was found that the computational speed and efficiency were improved (Ma et al. IEEE transactions on cybernetics).

For real-time data processing, it can be understood as: through the real-time data application platform of police business of the Public Security Department, a real-time control system can be built to serve the detection of public security cases; In the online loan application business of the financial industry, the real-time data application platform conducts data mining and analysis, and on the one hand, real-time feedback on user behavior; On the other hand, it helps enterprises make scientific analysis decisions in the shortest time, taking typical breakpoint marketing scenarios as examples; In the bank credit card business: the system application real-time data collection, cleaning, risk warning, help the bank to establish a credit card anti-fraud system, with the "channel-anti-fraud engine-host" implementation framework to judge fraudulent transactions and intercept (Liao & Ho, 2021).

The speed of data processing becomes particularly important when it comes to the value of a real-time data platform, which can be considered to diminish rapidly over time, and one of the key values of real-time processing is the ability to deliver data insights faster.

## 2. LITERATURE REVIEW

Cloud computing can flexibly adapt to high demand computing needs. The advantages provided by virtualization make cloud computing an attractive choice to meet the needs of high-performance computing and users.

Therefore, the article will be presented through Non-dominated Sorting Genetic Algorithm with Elite Strategy (NSGA-II) and Multi-Objective Particle Swarm Optimization (MOPSO). Algorithm-II

(NSGA-II) and Multi-Objective Particle Swarm Optimization (MOPSO). Are compared to measure the solution speed and solution quality of the two algorithms. The optimal result of collaborative optimization is one or a group of optimal solutions, which have the best performance in the case of multiple optimization objectives or variables. These solutions usually form a set called Pareto frontier, in which each solution is superior to other solutions in at least one optimization goal or variable, and cannot improve other goals.

In the context of scheduling processes assigned by users to virtual machines, reducing jobs is not only a focus of execution time, but also a focus of balancing energy and performance. In this work, Vila S proposed a Multi-objective Genetic Algorithm (MOGA) to determine the most suitable cloud let allocation to available virtual machines. This could generate scheduling decisions, and avoid system allocation, so as to provide new scheduling opportunities for the remaining cloud let to reduce overall execution time and energy consumption (Vila et al. 2019). Shaltout M L used MOGA to study the trade-off between competitive estimation errors. He found that compared with traditional statistical estimation methods, using MOGA for parameter estimation significantly improved accuracy (Shaltout et al. 2021). The development of GA in the current environment was becoming increasingly widespread, and many scholars conducted corresponding research and investigation on GA

Real time transaction computing is a common solution in the digital era, but most real-time data is used in transportation and finance, and transportation is mainly detected through real-time monitoring. Shen J processed and shared vehicle traffic data through real-time data, and provided feedback on incorrect traffic data. He found that incorrect data could reflect the situation of traffic safety; therefore, Shen J proposed a secure real-time traffic data aggregation scheme for vehicle clouds. In addition, Shen J also found that it outperformed previous solutions in terms of communication and computing costs (Shen et al. 2019). The rapid development of urbanization and independent industrial environment strengthened the demand for intelligent real-time video monitoring. Nawaratne R proposed an incremental spatiotemporal learner to address the challenges and limitations of anomaly detection and localization in real-time video surveillance. In Nawaratne R's view, the incremental spatiotemporal learner was a active learning based on fuzzy clustering, which could continuously update and identify new abnormal and normal states (Nawaratne et al. 2019). The results of these experiments validated the applicability of real-time video monitoring.

The growing needs of the financial services sector cannot be met by traditional intelligent business applications. They need uninterrupted access to intelligence that allows them to analyze large volumes of real-time events and gain rapid insight into the origin of events, i.e., respond to a changing world, often in an automated fashion. In other words, it is critical to be able to analyze and quickly respond to a continuous flow of high volume, real-time data. analysis and rapid response. In this regard the article will perform a multi-objective genetic algorithm for real-time trading computation.

## 3. MOGA UNDER DISTRIBUTED DATA

This section is the method introduction part of the article, mainly introducing Collaborative Optimization, Genetic Algorithm, Real-time transaction calculation of distributed data. The purpose is also to make the subsequent experimental part more clearly understand the meaning of various methods, so that readers can also understand the article more thoroughly. For example, Genetic Algorithm is based on the following Non-dominated Sorting Genetic Algorithm-II, and through the optimization of Genetic Algorithm, Non-dominated Sorting Genetic Algorithm-II is obtained, which is the purpose of this chapter.

With the popularity of the Internet and the development of digital technology, the amount of data generated in the world has increased exponentially. This data explosion provides huge data resources for the field of big data, attracting more people to invest in data analysis and processing.

### 3.1 Collaborative Optimization

The CO (Collaborative Optimization) method is a very promising cross regional optimization method (Shaam, 2020). This algorithm was discovered by researchers at Stanford University in the theory of subspace subsystems based on consistency constraints. In consistency constraint design algorithms, subsystems in each subspace can be designed but cannot be optimized. The CO algorithm not only designs all subspaces, but also achieves optimization through design.

CO is to divide some complex objective functions into simpler sub objective functions, and then achieve CO among these sub objective functions. Usually, when the values of variables are equal, the optimal solution can be considered. Although CO algorithm does not have local optimization problems, they have good convergence. This method can solve many nonlinear optimization and practical combinatorial optimization problems.

It is assumed that the variable $E(x_1, x_2, \cdots, x_n)$ of the objective function n can be abbreviated as $E(x)$, and CO can be used to decompose $E(x)$ into n single sub objective functions:

$$E(x) = E_1(x) + E_2(x) + \cdots + E_n(x) \tag{1}$$

If it is difficult to achieve consensus on each individual sub objective, $i = 1, 2, \cdots, n$ can be assumed and the optimal solution $x_i$ of $E_i(x)$ can be taken as the variable value, which is represented as $\tilde{x}_i$.

$$\tilde{x}_i = \arg \min_{x_i} \min_{X_i / x_i} E_i(x) \tag{2}$$

In the results of CO, if the value of each variable is the same as the solution result of each sub objective, it is the best solution.

The most important unsolved problem in modern optimization theory is the discovery of conventional global optimization conditions. This is because there are no conventional optimization conditions, so it is not known where to find the best solution or whether the existing solution is the best solution. Therefore, it is unclear how to organize the optimization process more effectively and when to interrupt the search in a timely manner. All conventional optimization conditions have theoretical importance and practical value. CO algorithm has solved this important problem on the basis of new optimization principles.

### 3.2 Multi-Objective Genetic Algorithm

GA is an optimization method based on biological evolution law first proposed in the 1970s. The computer model of biological evolution process is a method of imitating natural evolution process to find the best solution. The algorithm transforms the process of problem-solving from biological evolution to the process of crossover and mutation of chromosomal genetic factors through mathematical and computer simulation. When scholars solve complex combinatorial optimization problems, they can usually get better optimization results faster than some existing optimization algorithms (Rostami et al. 2021).

Pareto optimization type aims to find Pareto optimal solution set, also known as Pareto frontier. Pareto optimal solution is a set of solutions, none of which is superior to other solutions in all multiple objective functions. The trade-off type combines multiple objectives into a single objective by weighted summation of multiple objective functions, thus transforming the multi-objective problem into a single objective problem.

Genetic algorithm is an optimization algorithm based on biological evolution process. By simulating the evolution mechanism of biological population in nature, the optimal solution or near optimal solution of the problem is searched through continuous evolution and genetic operation. Genetic algorithm is a global optimization algorithm, which is usually used to solve complex optimization problems, especially those with huge search space and difficult to solve.

MOGA can be divided into two types: MOGA based on linear weighting; MOGA based on Pareto sorting (Sun et al. 2021).

The idea of a multifunctional GA based on linear weighted values is very simple. After converting multiple purposes into a single purpose using linear weighted values, traditional GA are applied to solve the problem, as shown in Formula (3):

$$fitness(I) = \sum_{k=1}^{K} w_k \times f_k(I) \tag{3}$$

Among them, $w_k$ in the formula represents the k-th target weight, and $f_k$ represents the k-th target value. Therefore, it can be seen that the core of this type of method is how to design weights. For example, Random Weight Genetic Algorithm (RWGA) uses the method of random weights. Each time fitness is calculated, and the weight values of different targets are randomly generated for each individual and then selected. Vector-Evaluated Genetic Algorithm (VEGA) also refers to a linearly weighted MOGA. Assuming there are K targets, VEGA would randomly divide the target population into K subgroups of the same size, with different subgroups selecting target values based on different objective functions, thus making selection operations. Therefore, the VEGA method is also a linearly weighted MOGA (Yashin et al. 2020). Vector evaluation genetic algorithm is a powerful optimization method, which can effectively search and improve the solution vector through the evolution process to solve multi-objective optimization problems and complex search space, and has wide application potential.

For the concept of MOGA based on Pareto sorting, it can be assumed that there are currently two unresolved problems, namely F1 and F2. If F1 has a higher goal than F2, it can be called F1 dominating F2. However, when F1 is not dominated by other solutions, it can be referred to as the Pareto solution. The set of Pareto solutions is called Pareto front. MOGA, Non-dominated Sorting Genetic Algorithm (NSGA), and NSGA-II with elite strategy are commonly used MOGA based on Pareto sorting.

### 3.3 Distributed Data Real-Time Transaction Calculation

Distributed computing is a method where different computers work together to solve common problems (Ramu et al. 2020). This makes the computer network look like a powerful calculator and provides a large amount of resources to handle complex challenges (Shabnnam, 2020). For example, distributed computing can encrypt large amounts of data, and solve physical and chemical formulas with multiple variables, so as to produce high-quality three-dimensional animations. Distribution systems, distribution programs, and distribution algorithms refer to other terms used to refer to distributed computing (Sangaiah et al. 2019).

The advantages of distributed data are as follows: scalability: Distributed systems can evolve based on workload and requirements, and add new nodes through data published on computer networks; availability: Even if one computer fails, distributed data would not cause the entire computer to crash, and this design improves the system's fault tolerance; consistency: In distributed data, there are shared resources between computers, but the entire system would be managed to maintain consistency between different computers (Pushppita, 2022). Without affecting data consistency, the advantage of fault tolerance can be enjoyed; transparency: Distributed computing systems provide logical separation between users and physical devices. Interacting with the system, distributed data works like a computer without worrying about the settings and configurations of individual machines (Verbraeken et al. 2020).

A distributed computing system composed of different hardware, middleware, software, and operating systems provides logical separation between users and physical devices; efficiency: Distributed systems provide faster performance by utilizing the best resources of basic hardware. Operating systems can improve efficiency by effectively managing multiple tasks and processes. This includes reasonable task switching, process priority scheduling and resource sharing. The operating system can also use memory paging, virtual memory and other technologies to minimize memory fragmentation and improve memory utilization.

Real time operations are typically performed on a large amount of information, typically in seconds. In the early days of the rise of big data analysis, although Hadoop did not provide a solution for real-time computing, the emergence of real-time computing frameworks (such as Storm, Spark Streaming and Kafka) made real-time computing more mature. Real-time streaming computing would be more fully applied in these areas (Zhao et al. 2020).

Fast dominance ranking is a ranking algorithm for solving multi-objective optimization problems. Its purpose is to classify a group of solutions according to their non-dominance, so as to construct Pareto frontier, that is, the set of non-dominance solutions.

The three characteristics of real-time computing: infinite data: Infinite data constantly increasing, and it is essentially an infinite dataset. This is called stream data, which is the opposite of a finite dataset. Infinite data processing refers to a series of data processing processes that enable processing engines to repeatedly process infinite data and break through the bottleneck of limited data processing engines; there is no clear definition of low latency. However, the value of data would decrease over time, and effectiveness is a problem that needs to be continuously addressed. At present, big data applications are a relatively popular field. For example, technical limitations may take one minute, one hour, or even more at the beginning of the practical system. This does not meet user needs. What users need is not the ability to perform offline processing but the ability to quickly process data.

## 4. EVALUATION OF SEVERAL MULTI-OBJECTIVE GENETIC ALGORITHM

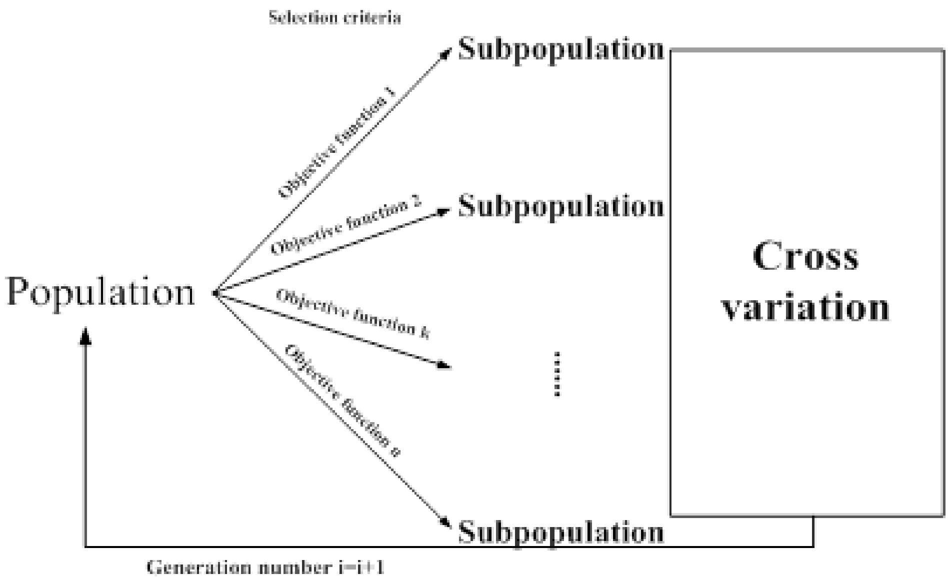
### 4.1 Parallel Selection Method

The basic idea: Population is evenly divided into subpopulations based on the number of sub objective functions. Each subpopulation is assigned an objective function and selected optimally, and individuals with high fitness are selected to form a new subpopulation. All these sub individuals are combined into complete individuals. Here, the crossover mutation operation is performed to create a complete next entity and a Pareto optimal solution, as shown in Figure 1.

The main steps: (1) Through objective function, individuals in the population are divided into subpopulations, and each subpopulation is assigned an objective function. (2) The individuals with higher fitness are selected based on the objective function through the individuals in the sub population, and then they are formed into a sub population. (3) Through subpopulation, mating and mutation are carried out to generate the next generation of father population, and the first step is repeated. The disadvantage of the parallel selection method is that it is easy to generate the ultimate optimal solution of a single objective function, and it is also difficult to generate a compromise solution that satisfies multiple objectives to a certain extent.

Parallel selection usually requires a lot of computing resources because it compares the fitness of multiple individuals at the same time. This may cause the algorithm to be very expensive in calculation, especially on large-scale problems. In the evolution of genetic algorithm, the fitness distribution of individuals may become uneven. If the parallel selection method is used, the individuals with high fitness may be over-selected, while the individuals with low fitness are ignored, thus falling into the local optimal solution.

**Figure 1. Parallel selection process**




### 4.2 Multi-Objective Particle Swarm Optimization

The MOPSO treats the individuals in the external archive as elite individuals. By controlling the evolution direction of the population through elite individuals, the population is guided to approach the real Pareto frontier. After the algorithm runs, the particles in the external archive are used as the approximate solution of the Pareto optimal solution obtained. The current MOPSO can be divided as follows: The first is to embed Pareto dominance relationships into Particle Swarm Optimization (PSO), which is used to determine the individual's historical best particles and global best particles; the second type can use the decomposition method to transform the multi-objective optimization problem into a set of single purpose optimization problems, and then directly apply PSO to solve the single purpose optimization problem, thus making full use of the convergence speed and global optimization ability of PSO, which can better solve the multi-objective optimization problem (Yuen et al. 2020).

Two core problems should be solved: 1. How to select Pbest and Gbest to update particle positions and velocities. 2. How to cut when the number of non dominated solutions exceeds the size of external archives. Figure 2 shows algorithm process of MOPSO.

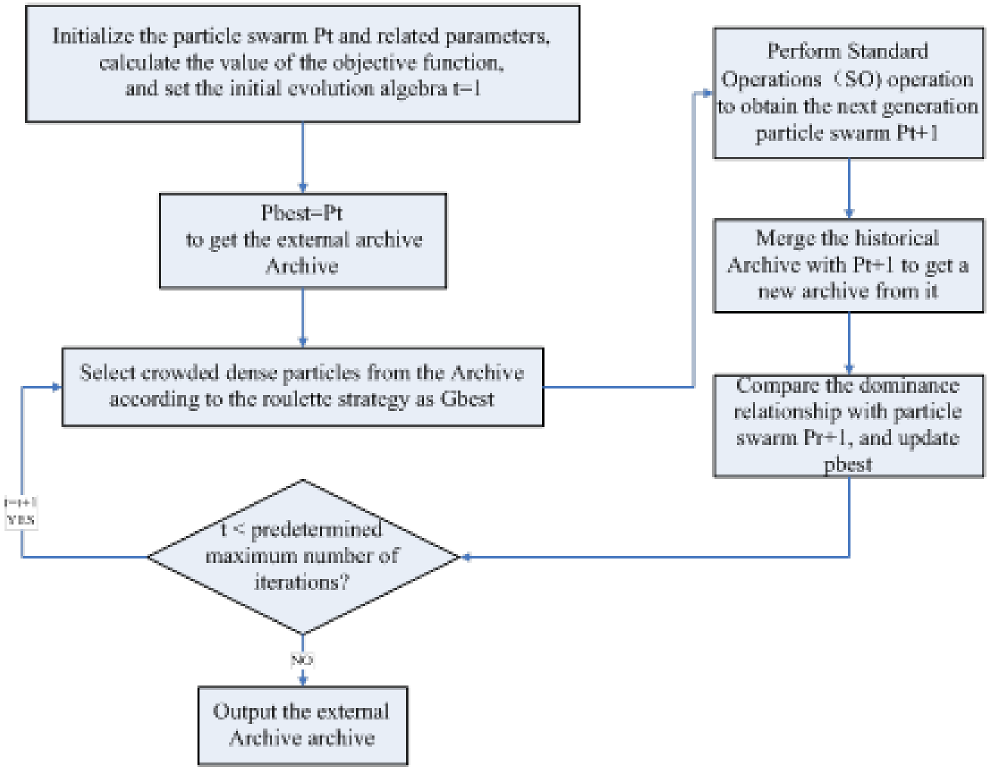
### 4.3 NSGA-II With Elite Strategy

NSGA-II is a further improvement of GA. Each generation randomly selects individuals from the current population and reproduces offspring through crossover and mutation into the next generation. Compared to GA, NSGA-II improves two choices in its main cycle: One is to evolve one population into another through crossover and mutation: The second is to determine elite candidates in a combination of two populations based on the criteria of dominance and crowding, which form the next generation. When applying the NSGA-II in practice, the final result (that is, the candidate of the last generation) is usually considered as the Pareto optimal solution. However, due to the number of iterations, these final results may not necessarily be the true Pareto optimal solutions (Chen et al. 2020).

Figure 3 shows specific process of NSGA-II with elite strategy:

The specific process of NSGA-II with elite strategy is as follows:

Figure 2. Process of MOPSO



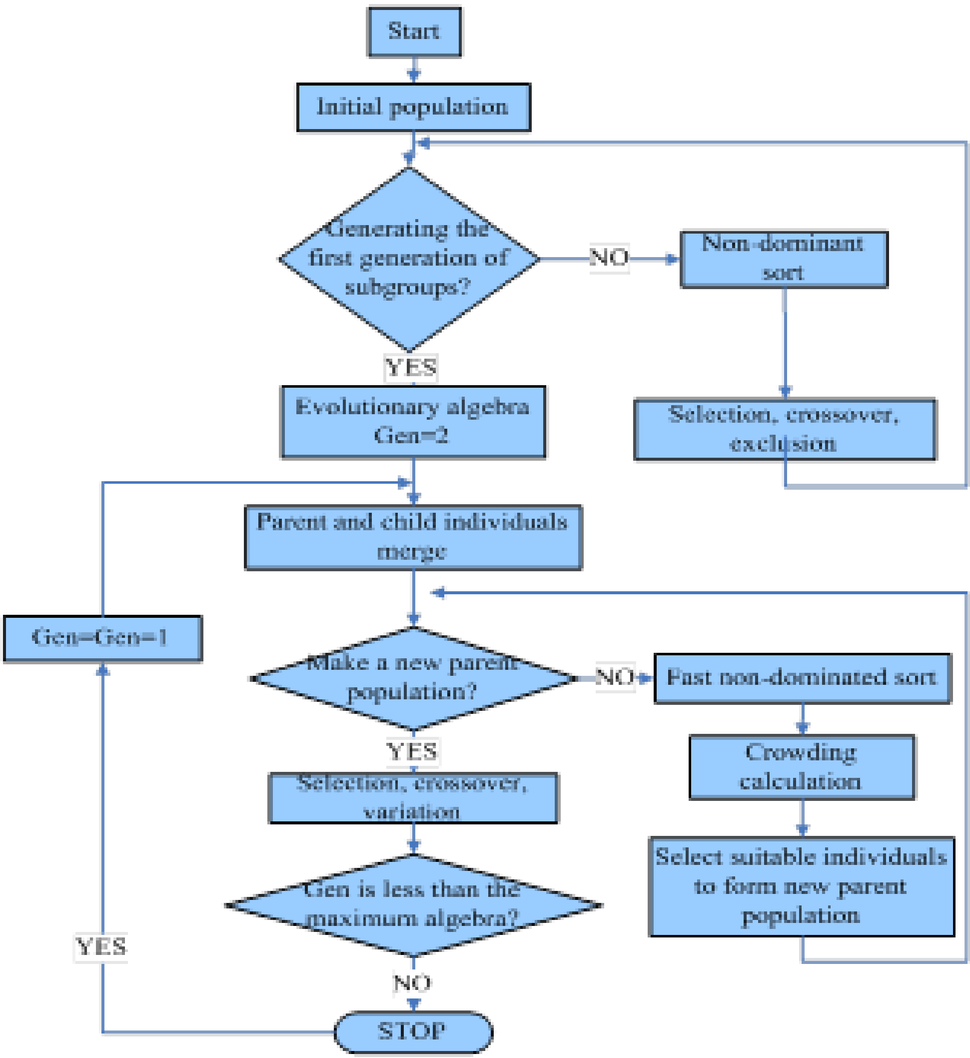

The initial population of N size is randomly generated. After non dominated sorting, the first generation subpopulation is obtained. Starting from the second generation, the parent population and offspring population merge and quickly perform non dominated sorting, while calculating the density of individuals in each non dominated layer. Appropriate individuals are selected to form a new parent population; through the basic operations of GA, new offspring populations are generated, and so on until the conditions for program completion are met (Zhao et al. 2018).

NSGA-II can maintain a Pareto frontier set, which contains the candidate set of non-dominated solutions. This frontier set provides multiple non-dominated optimal solutions to the problem, which gives decision makers more flexibility in different trade-offs and choices.

The characteristics of GA: The algorithm retrieves from a set of problem solutions. GA search within a set of strings, with a wide range that facilitates global selection. GA can simultaneously process multiple individuals in a population. The fitness function can be set in the field at will. GA uses probability theory transition rules to guide the direction of exploration. The genetic factor algorithm utilizes the information obtained during the evolution process to explore organization and discover individuals with higher fitness, thus resulting in a higher probability of survival. In addition, the algorithm itself also adopts dynamic adaptation technology. It is like a fuzzy adaptation method. The algorithm control parameters and encoding accuracy can be automatically adjusted during the evolution process (Wang et al. 2022).

NSGA-II is a genetic algorithm, which is specially used to solve multi-objective optimization problems. It maintains a fitness frontier set, which contains the candidate set of non-dominant solutions, by sorting the individuals in the population and allocating the crowding degree. MOPSO is a particle swarm optimization algorithm, which is also used for multi-objective optimization. In

Figure 3. NSGA-II process




MOPSO, each particle represents a solution, and the quality of the solution is continuously improved by simulating the movement of particles in the multidimensional search space.

After adjusting the parameters, experiments and performance evaluation are needed again to verify whether the performance of the algorithm is improved. This process may require many iterations. If the modification of parameters leads to better performance or closer to the optimization goal, then you can choose to adopt the modified parameters. If the proposed parameters can meet the requirements, they can be kept unchanged.

## 5. EVALUATION OF EXPERIMENTAL RESULTS OF GA

### 5.1 Comparison of NSGA-II and MOPSO Algorithms

The NSGA-II proposes fast non dominated sorting, which reduces computational complexity and merges the parent and child populations to select the next generation in twice the space, thereby preserving all outstanding individuals; elite strategy is introduced to ensure that certain dominant populations are not abandoned during the evolution process; a new algorithm is proposed, which not only solves the dependency on shared parameters in the NSGA algorithm, but also utilizes this operator to compare populations, enabling individuals in the quasi Pareto region to be evenly distributed throughout the entire region, thereby ensuring the diversity of the population (Hojjati et al. 2018).

The parameters involved in the MOPSO algorithm mainly include: the number of particle swarm; number of iterations; the speed range and position range of the example; fitness value; archive threshold; inertia factor; speed factor; grid. The selection of control parameters can affect the performance and efficiency of the algorithm. General parameter setting: There are 50 particles; the upper limit number of non inferior solution sets is 50; the individual learning factor C1 is 1; the group learning factor C2 is 2; the variation factor is 0.1; the maximum number of iterations is 200.

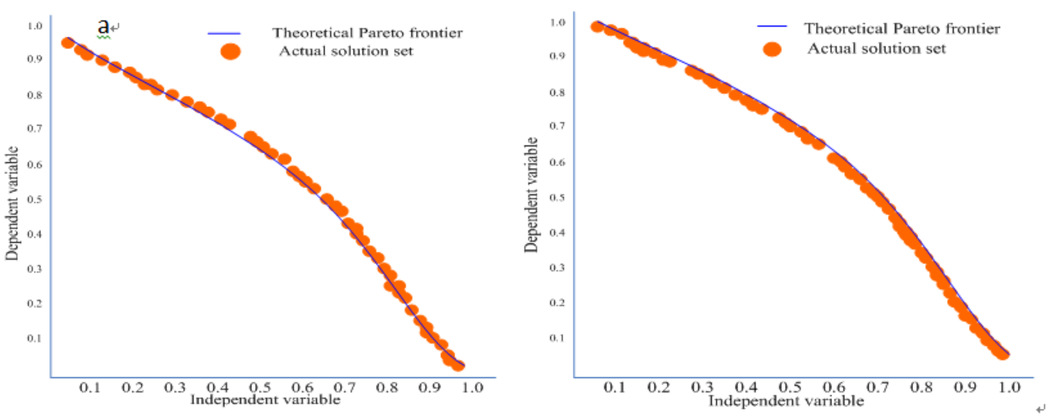
The algorithm solution results are shown in Figure 4.

In Figure 4 (a), the actual solution set of the NSGA-II was more uniform. Figure 4 (b) showed that the actual solution set of the MOPSO algorithm was not uniformly distributed. Two algorithms were compared and it was found that MOPSO had a slower computational speed and lower computational quality compared to NSGA-II (Rabbani et al. 2022). In multi-objective optimization, different optimization objectives may conflict, and improving one objective may lead to the deterioration of another.

## 5.2 NSGA-II Optimization Discussion

By comparing the two algorithms mentioned above, this article would choose the NSGA-II and conduct optimization research on it. Specific algorithm model steps: Firstly, the parameters are set, including global and local parameters. Global parameters include: number of chromosomes, number of genes per chromosome, range of gene values, and number of iterations. Local parameters include: fitting correlation coefficient threshold, selling on the Nth day after buying, yield base score, and unselected chromosome score. The above parameters are custom parameters and can be adjusted according to needs. The parts related to parameters are marked in quotation marks below. Secondly, the samples to be studied were selected and divided into training and testing sets. P chromosomes are initialized, with M genes per chromosome, which is a matrix of PxM. The generated chromosomes are transferred into the fitness function. Firstly, not every chromosome's morphology is consistent with the actual yield morphology. Therefore, the first-order least squares fitting method is used to compare the actual yield with each chromosome. If its correlation is greater than or equal to the "matching correlation threshold", it is scored. Otherwise, the score of this chromosome is "unselected chromosome score".

**Figure 4. Algorithm solution results: (a) NSGA-II, (b) MOPSO algorithm**



The generated chromosomes are used for transaction backtesting on the training and testing sets, and the backtesting results are written into the database. Based on the score of each chromosome, their probability of being selected is calculated, and the probability of being selected is proportional to the score. Through standard GA, chromosomes undergo the process of crossing, mutating, and replicating, and then new chromosomes are generated. The generated new chromosome is substituted into the next iteration until the set "iteration count" is reached. The results in the database are reviewed to find the best strategy. Figure 5 shows execution steps of the elite strategy.

To test the NSGA-II, the NSGA-II is set up with four sets of data, and the conclusion of the data is used to understand the NSGA-II. The main loop of the algorithm includes selection, crossover, and mutation, and the specific data is shown in Table 1.

To demonstrate whether the NSGA-II needs improvement, 48 sets of sample data were cited. The algorithm was optimized and compared before and after, and the comparison results were shown in Figure 6 and Figure 7. The X coordinate represented the number of samples, and the Y coordinate represented the output variable. The output variable F1 was the heating load, and the output variable F2 was the cooling load. Figure 6 (a) shows the traditional optimization of F1, and Figure 6 (b) shows the traditional optimization of F2. Figure 6 shows the optimized output of the traditional NSGA-II.

Figure 7 (a) shows F1 improved optimization, and Figure 7 (b) shows F2 improved optimization. Figure 7 shows the improved optimization output of the NSGA-II.

**Figure 5. Execution steps of elite strategy**

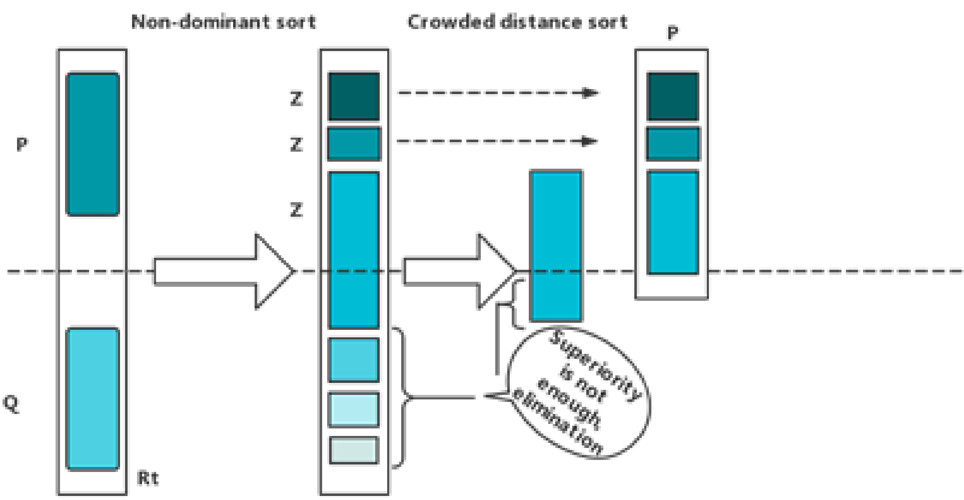
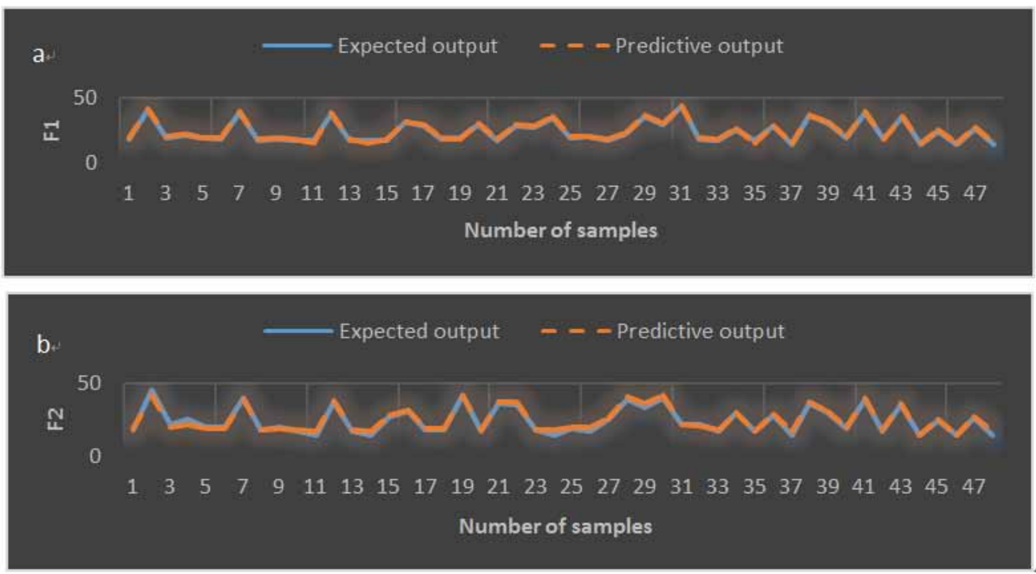



**Table 1. NSGA-II parameter settings**

| Parameter Item | NSGA-I (1) | NSGA-II (2) | NSGA-I (3) | NSGA-II (4) |
|---|---|---|---|---|
| Population | 100 | 100 | 100 | 100 |
| Crossover rate | 1.0 | 0.8 | 0.6 | 0.4 |
| Variation rate | 1/n | 1/n | 1/n | 1/n |
| Cross distribution index | 10 | 10 | 10 | 10 |
| Variation distribution index | 15 | 15 | 15 | 15 |

**Figure 6. Optimization output of traditional NSGA-II: (a) F1 traditional optimization, (b) F2 traditional optimization**

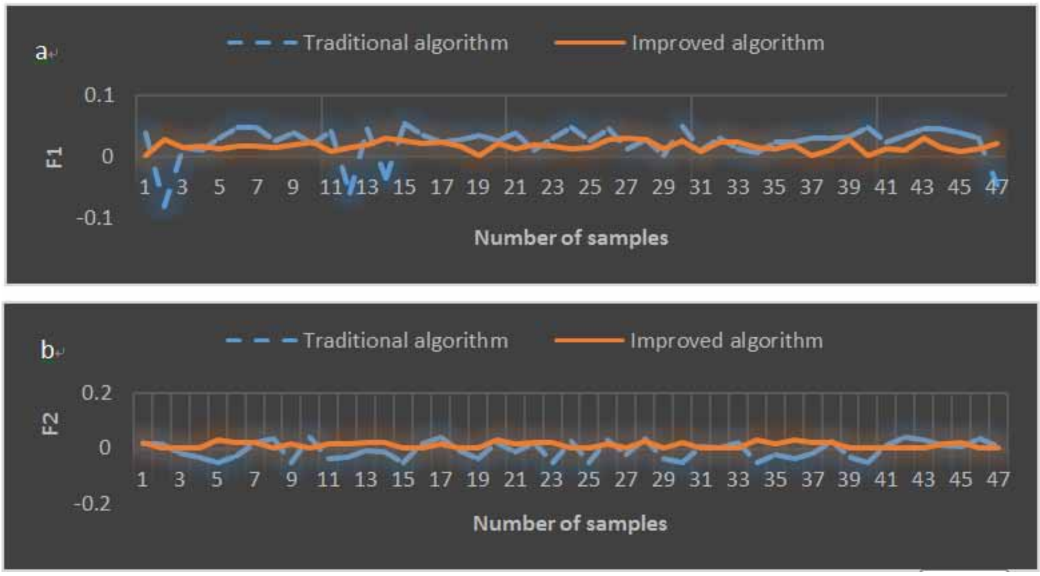

**Figure 7. Improved NSGA-II optimization output: (a) F1 improved algorithm optimization output, (b) F2 improved algorithm optimization output**




By comparing Figure 6 with Figure 7, it could be observed that the overall output of Figure 6 fluctuated significantly. Figure 7 showed the optimized output after improvement, and it could be seen that the overall output of Figure 7 was tending to a stable state. It could be found that the improved output was more stable than the pre improved output. The data showed that the actual predictions were close to the expected predictions, thus demonstrating stronger multi-objective optimization capabilities.

In order to make the data before and after the improvement more obvious, the data was compared by outputting the percentage before and after the improvement. The comparison results were shown in Figure 8. Figure 8 (a) showed that there was still a significant change in the traditional algorithm before and after the improvement compared to the improved algorithm. The F1 data of the traditional algorithm changed obviously before and after the improvement, and the data fluctuated greatly, while the F1 data of the improved algorithm was relatively stable. In Figure 8 (b), there was no significant fluctuation between the traditional algorithm and the improved algorithm before and after the improvement, and they could gradually converge. However, overall, it could be concluded that the improved data would always be more stable than the data before the improvement.

Figure 8. Output percentage of NSGA-II before and after improvement: (a) output percentage before and after F1 improvement, (b) output percentage before and after F2 improvement




After the above improvements, the percentage data could be observed that compared to traditional algorithms, the improved algorithm tended to be more stable. It could be concluded that the improved data was more accurate. The distribution of the optimal solution was also more uniform, which verified the necessity and rationality of improving the NSGA-II. The stability of GA was better than that of MOPSO algorithm, but it did not mean that MOPSO algorithm had no advantages.

Adjust and optimize the parameters of NSGA-II more carefully. This includes population size, crossover rate, mutation rate, non-dominant ranking parameters and so on. The performance of the algorithm can be improved by systematic parameter tuning. Consider introducing new genetic operation strategies, such as different crossover methods, mutation methods or selection methods. These new strategies may help to increase the diversity of search space, so as to better explore the potential solution space.

### 5.3 Discussion

From the above numerical simulation results, it could be seen that the MOPSO algorithm exhibited outstanding ability in global exploration and was suitable for solving various complex optimization problems. The parallel search mode could simultaneously search for multiple non inferior solutions, which was beneficial for obtaining the optimal solution set (Esmaeilion et al. 2021).

In addition, the algorithm itself had strong universality and low complexity, and it could be organically combined with conventional computing methods and intelligent algorithms. Compared to the NSGA-II, the main conclusion was that the algorithm was stable and was currently one of the most commonly used multi-functional GA. Its advantages were the reduced complexity of non column sorting GA and fast execution speed, thus making it the standard for other multi-purpose optimization algorithms.

## 6. CONCLUSION

The evolutionary algorithm NSGA-II is a classic method used to solve multi-objective problems. It is applied in many fields and is the first algorithm learned by many researchers who have just been exposed to multi-objective research fields. For researchers who are not familiar with the field of multi-objective optimization, algorithm ideas, parameter settings, and result analysis make it difficult for many to implement. Therefore, this paper would introduce the NSGA-II in a large space, so that researchers could quickly and clearly understand the field of multi-purpose evolutionary algorithms during the preparatory learning stage. In the initial stage, this article provided good guidance for the learning of evolutionary algorithms, but the goal of learning algorithms was to improve them based on actual target problems and solve them. Therefore, researchers need to further study in the future.

## FUNDING

This work was supported by Research and development of personalized intelligent course recommendation system based on big data project of science and technology research program of Chongqing Education Commission of China (No KJQN202203915) and Chongqing Yongchuan District Natural Science Foundation Project(Design and implementation of a real-time e-commerce analysis system based on Flink (No 2023yc-jckx20077).

## REFERENCES

Babaeinesami, A., Tohidi, H., Ghasemi, P., Goodarzian, F., & Tirkolaee, E. B. (2022). A closed-loop supply chain configuration considering environmental impacts: A self-adaptive NSGA-II algorithm. *Applied Intelligence*, *52*(12), 13478–13496. doi:10.1007/s10489-021-02944-9

Chen, G., Shi, J., Xu, H., Li, T., & Yang, W. (2020). An improved genetic algorithm in shared bicycle parking point allocation. *International Journal of Security and Networks*, *15*(3), 141–147. doi:10.1504/IJSN.2020.109710

Esmaeilion, F., Ahmadi, A., & Dashti, R. (2021). Exergy-economic-environment optimization of the waste-to-energy power plant using multi-objective particle-swarm optimization (MOPSO). *Scientia Iranica*, *28*(5), 2733–2750. doi:10.24200/sci.2021.55633.4323

Hojjati, A., Monadi, M., Faridhosseini, A., & Mohammadi, M. (2018). Application and comparison of NSGA-II and MOPSO in multi-objective optimization of water resources systems. *Journal of Hydrology and Hydromechanics*, *66*(3), 323–329. doi:10.2478/johh-2018-0006

Liao, S., & Ho, C. (2021). Mobile Payment and Mobile Application (App) Behavior for Online Recommendations. *Journal of Organizational and End User Computing*, *33*(6), 1–26. doi:10.4018/JOEUC.20211101.oa2

Liu, S., & Wang, N. (2020). Collaborative optimization scheduling of cloud service resources based on improved genetic algorithm. *IEEE Access : Practical Innovations, Open Solutions*, *8*, 150878–150890. doi:10.1109/ACCESS.2020.3016762

Ma, L., Li, N., Guo, Y., Wang, X., Yang, S., & Huang, M. (n.d.). Learning to optimize: reference vector reinforcement learning adaption to constrained many-objective optimization of industrial copper burdening system. *IEEE Transactions on Cybernetics*.

Nawaratne, R., Alahakoon, D., & De Silva, D. (2019). Spatiotemporal anomaly detection using deep learning for real-time video surveillance. *IEEE Transactions on Industrial Informatic*, *16*(1), 393-402. .10.1109/TII.2019.2938527

Pushppita, I. (2022). Distributed Computing Collaborative CAD System Based on COM / DCOM. *Distributed Processing System*, *3*(1), 46–53. doi:10.38007/DPS.2022.030106

Rabbani, M., Oladzad-Abbasabady, N., & Akbarian-Saravi, N. (2022). Ambulance routing in disaster response considering variable patient condition: NSGA-II and MOPSO algorithms. *Journal of Industrial and Management Optimization*, *18*(2), 1035–1062. doi:10.3934/jimo.2021007

Ramu, N., Pandi, V., Lazarus, J. D., & Radhakrishnan, S. (2020). A Novel Trust Model for Secure Group Communication in Distributed Computing. *Journal of Organizational and End User Computing*, *32*(3), 1–14. doi:10.4018/JOEUC.2020070101

Ransikarbum, K., & Mason, S. J. (2022). A bi-objective optimisation of post-disaster relief distribution and short-term network restoration using hybrid NSGA-II algorithm. *International Journal of Production Research*, *60*(19), 5769–5793. doi:10.1080/00207543.2021.1970846

Rostami, M., Berahmand, K., & Forouzandeh, S. (2021). A novel community detection based genetic algorithm for feature selection. *Journal of Big Data*, *8*(1), 1–27. doi:10.1186/s40537-020-00398-3 PMID:33425651

Sangaiah, A. K., Medhane, D. V., Han, T., Hossain, M. S., & Muhammad, G. (2019). Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics. *IEEE Transactions on Industrial Informatics*, *15*(7), 4189–4196. doi:10.1109/TII.2019.2898174

Shaam, D. (2020). Collaborative Optimization Method for Distributed Systems based on Deep Learning. *Distributed Processing System*, *1*(1), 34–41. doi:10.38007/DPS.2020.010105

Shabnnam, D. (2020). Distributed Fault-tolerant Computing Based on LMI Technology. *Distributed Processing System*, *1*(4), 10–17. doi:10.38007/DPS.2020.010402

Shaltout, M. L., Mostafa, M. A., & Metwalli, S. M. (2021). Enhancement of wind energy resources assessment using Multi-Objective Genetic algorithm: A case study at Gabal Al-Zayt wind farm in Egypt. *International Journal of Green Energy*, *18*(14), 1497–1509. doi:10.1080/15435075.2021.1904949

Shen, J., Liu, D., Chen, X., Li, J., Kumar, N., & Vijayakumar, P. (2019). Secure real-time traffic data aggregation with batch verification for vehicular cloud in VANETs. *IEEE Transactions on Vehicular Technology*, *69*(1), 807–817. doi:10.1109/TVT.2019.2946935

Sun, Y., Xue, B., Zhang, M., Yen, G. G., & Lv, J. (2020). Automatically designing CNN architectures using the genetic algorithm for image classification. *IEEE Transactions on Cybernetics*, *50*(9), 3840–3854. doi:10.1109/TCYB.2020.2983860 PMID:32324588

Verbraeken, J., Wolting, M., Katzy, J., Kloppenburg, J., Verbelen, T., & Rellermeyer, J. S. (2020). A survey on distributed machine learning. *ACM Computing Surveys*, *53*(2), 1–33. doi:10.1145/3377454

Vila, S., Guirado, F., Lerida, J. L., & Cores, F. (2019). Energy-saving scheduling on IaaS HPC cloud environments based on a multi-objective genetic algorithm. *The Journal of Supercomputing*, *75*(3), 1483–1495. doi:10.1007/s11227-018-2668-z

Wang, M., Zhang, Y., Lu, Y., Wan, X., Xu, B., & Yu, L. (2022). Comparison of multi-objective genetic algorithms for optimization of cascade reservoir systems. *Journal of Water and Climate Change*, *13*(11), 4069–4086. doi:10.2166/wcc.2022.290

Yashin, S., Yashina, N., Koshelev, E., Kashina, O., & Pronchatova-Rubtsova, N. (2020). Foresight of Volga Federal District innovation system development using a multi-objective genetic algorithm. *International Journal of Technology*, *11*(6), 1171–1180. doi:10.14716/ijtech.v11i6.4432

Yuen, M. C., Ng, S. C., & Leung, M. F. (2020). A competitive mechanism multi-objective particle swarm optimization algorithm and its application to signalized traffic problem. *Cybernetics and Systems*, *52*(1), 73–104. doi:10.1080/01969722.2020.1827795

Zhao, B., Xue, Y., Xu, B., Ma, T., & Liu, J. (2018). Multi-objective classification based on NSGA-II. *International Journal of Computing Science and Mathematics*, *9*(6), 539–546. doi:10.1504/IJCSM.2018.096325

Zhao, J., Sun, X., Li, Q., & Ma, X. (2020). Edge caching and computation management for real-time internet of vehicles: An online and distributed approach. *IEEE Transactions on Intelligent Transportation Systems*, *22*(4), 2183–2197. doi:10.1109/TITS.2020.3012966

Zhou, J., Du, Z., Yang, Z., & Xu, Z. (2020). Dynamic parameters optimization of straddle-type monorail vehicles based multiobjective collaborative optimization algorithm. *Vehicle System Dynamics*, *58*(3), 357–376. doi:10.1080/00423114.2019.1578384