Multi Objective Optimization System for Bridge Design Based on Multi-objective Optimization Theory and Improved Ant Colony Algorithm

Qiqi Wen
Chongqing Leway civil engineering design Co., Ltd, China
E-mail address: loquat42@163.com

Keywords: multi-objective optimization, ant colony algorithm, bridge design, pareto solution set, genetic algorithm

Received: September 26, 2024

In the field of bridge design, multi-objective optimization problems have attracted much attention due to their complexity and multiple solutions. The limitations of existing optimization algorithms in dealing with multi-objective problems, especially the trade-off between multiple objectives such as cost, duration, safety and quality. Therefore, in order to achieve the balance and optimization of each optimization objective while satisfying the bridge design constraints, a multi-objective optimization system based on an improved ant colony algorithm is studied and developed. The study is conducted by modeling natural selection and genetic mechanisms to improve the global search capability and diversity of the algorithm. The results showed that the proposed system was significantly superior to the traditional methods in key performance indicators such as optimization speed, objective function value, and robustness. The accuracy, stability, and safety of the proposed system were as high as 92%, 95%, and 91%, respectively, while the corresponding indicators of the traditional system were only about 55%. Specifically, the optimization speed of the proposed system reached 0.95, which was significantly better than that of the traditional system of 0.70, indicating that the proposed system had a significant advantage in convergence speed. The objective function value of the proposed system was 0.92, which was better than 0.75 of the traditional system, indicating that the proposed system could achieve a more optimal solution when solving optimization problems. The proposed system is superior to the traditional system in all evaluation indices, which proves its superior performance in multi-objective optimization of bridge design. The study provides a new optimization strategy for bridge design, which helps to achieve a more efficient, economical and safer bridge design solution.

Povzetek: The study introduces a multi-objective optimization system for bridge design, utilizing an enhanced ant colony algorithm to balance factors like cost, duration, safety, and quality. Additionally, the system demonstrates a faster optimization speed and better objective function values, indicating its superior performance in multi-objective bridge design optimization.

1 Introduction

As an important transportation infrastructure, the design quality of bridges is directly related to public safety and economic development. With the acceleration of urbanization and the advancement of engineering technology, modern bridge design not only has to meet the basic load carrying and safety requirements, but also has to consider multiple factors such as economic benefits, construction efficiency, aesthetics, and environmental impact. These factors are intertwined with each other, constituting a complex multi-objective optimization (MOO) problem, which puts forward higher requirements for designers [1-2]. Traditional bridge design methods often focus on the optimization of a single objective, and it is difficult to consider multiple objectives at the same time. This leads to problems such as high cost, prolonged construction period or insufficient safety in practical application of design solutions. In addition, with the application of new materials and technologies, the complexity of bridge design further increases. Traditional optimization methods appear to be incompetent in dealing with such problems. In recent years, MOO algorithms such as genetic algorithm (GA), particle swarm optimization (PSO) and ant colony algorithm (ACA) have been gradually applied in bridge design. They provide new ways to solve problems by modeling natural phenomena [3-4].

For the MOO problem, where multiple optimal solutions (OSs) may exist. Pereira et al. did a thorough analysis of MOO algorithms, which have become popular in engineering in recent years. Their study revealed the effectiveness of traditional optimization methods in dealing with the number of variables, number of objectives and nonlinear problems. Although these algorithms were not commonly used in engineering practice, they showed significant potential for improvement in real-world applications [5]. Jangir et al. proposed a novel MOO algorithm for Pareto frontier problems with multiple conflicting objectives and features, including linear, nonlinear, continuous and discrete. The algorithm combined the elite non-dominated ordering and congestion distance mechanisms and was found to outperform some recognized good algorithms by experimental comparison [6]. An inventive multi-objective balancing optimizer was created by Premkumar et al. to address challenging optimization issues, such as engineering design. The study used a multi-objective balancing algorithm with a non-dominated sorting method. The study's findings demonstrated that, in contrast to existing algorithms, this new algorithm offered more competitive solutions [7]. Rao et al. optimized the original heuristic algorithm for the MOO problem. This improved algorithm utilized the dominance principle and congestion distance evaluation mechanism to handle multiple objectives simultaneously. Experimental results demonstrated that this optimization algorithm was not only concise but also robust and suitable for solving diverse engineering optimization problems [8]. In structural health monitoring, the optimization of sensor layout is very important to improve the diagnostic accuracy and reduce the computational burden. Yang et al. improved genetic and simulated annealing (SA) algorithms, developed adaptive simulated annealing GAs, and combined with strain modal criteria to achieve accurate sensor layout. Simulation showed that this algorithm had the lowest average false discovery rate and was superior to target detection and negative selection algorithms [9]. Aiming at the damage detection of large-span spatial lattice structures, Zhou et al. developed an improved GA damage detection model. Experiments confirmed that the model achieved a balance between recall rate and precision rate, with AUC value as high as 0.927, optimization error less than 0.4. It improved convergence performance, and achieved 100% convergence rate and fast iteration [10]. To achieve the minimization of power consumption, the shortest task completion time, and to guarantee the quality of service obtained by cloud service users, Elsedimy et al. developed an integrated multi-objective task scheduling model, which is based on an improved ant colony optimization (ACO) algorithm. The algorithm introduced adaptive distribution probabilities especially in global rule updating. It showed improved performance in terms of load balancing, energy economy, turnaround time, and completion time, according to experimental data [11]. For the two additional goals of decreasing transit time span and distance imbalance, Goel et al. suggested a meta-heuristic algorithm based on a multi-ant colony system. The algorithm was tested on several benchmark problems and showed advantages over other advanced methods as well as the non-dominated sorting genetic algorithm II (NSGA-II), which was commonly used in the MOO field [12]. Masoumi et al. suggested an improved multi-objective ACO algorithm to address this challenge. The results indicated that the algorithm achieved an acceptable level of reasonableness of the setup, reproducibility of the results, and runtime [13]. Conventional ACA usually used only one pheromone and updated the pheromone by a non-dominant solution to provide guidance for subsequent foraging behavior without utilizing the dominant solution. To utilize the dominant solution more effectively, a novel ACO algorithm was suggested by Ning et al. The algorithm was based on the multi-objective evolutionary algorithm of decomposition and combined the concepts of ACO and negative pheromone. The results of the study confirmed that the reasonable utilization of the relevant information of the dominant solution can significantly improve the efficiency of ACA [14]. The summary table of literature review is shown in Table 1.

Table 1: Summary of literature review

Reference	Algorithm multi-objective optimization algorithm	GoalsDeal with number of variables, number of targets, and nonlinear problems	Key performance indicator
[5]	Elite non-dominant sorting and congestion distance mechanism algorithm	Solve problems with multiple conflicting goals and features	It has the potential of effectiveness and improvement
[6]	Multi-objective balancing optimizer	Solve complex optimization problems including engineering design	The performance is better than the recognized excellent algorithm
[7]	Heuristic algorithm optimization	Adapt to the needs of multi-objective optimization problems	Provide more competitive solutions
[8]	Adaptive simulated annealing genetic algorithm	Optimal layout of sensors in structural health monitoring of civil engineering	Simple and robust
[9]	Improved genetic algorithm damage recognition model	Damage detection of long-span space grid structure	Average false detection rate, optimization error
[10]	Improved ant colony optimization algorithm	Minimize power consumption and task completion time to ensure quality of service	Recall rate, precision rate, AUC value, optimization error

[11]	Meta-heuristic algorithm based on multi-ant colony system	Minimize travel time span and distance imbalances	Completion time, turnaround time, energy efficiency, load balancing
[12]	Improved multi-objective ant colony optimization algorithm	User-centered path planning	Performance exceeds NSGA-II
[13]	Multi-objective evolutionary algorithm based on decomposition	Improve the efficiency of ant colony algorithm by using dominant solution	Setting rationality, repeatability of results, running time
[14]	Algorithm multi-objective optimization algorithm	GoalsDeal with number of variables, number of targets, and nonlinear problems	Efficiency improvement

In conclusion, these algorithms still have issues with sluggish convergence and a tendency to default to local optimization when handling large bridge design problems with several competing goals. To address these shortcomings, the study proposes an improved ACA (IACA). The method improves the global search capability and variety by incorporating mutation mechanisms and chromosome crossover, which are essential components of GA. In addition, the study introduces a pheromone decomposition mechanism and the concept of negative pheromone to utilize the dominant solution information more efficiently. The combination of GA's crossover and mutation operations, which enhances the algorithm's exploration capability and its capacity to avoid local optima (LO), is the study's unique contribution. To improve the algorithm's use of favorable solutions and optimize the pheromone updating rules, a decomposition-based multi-objective evolutionary technique is presented.

2 Methods and materials

The study firstly clarifies the logical connection between the optimization objectives (OOs) by constructing a functional relationship equation, and optimizes the bridge design multi-objective based on Pareto solution set. Finally, an MOO system is constructed through IACA to further enhance the overall performance and optimization effect of bridge design.

2.1 Bridge design MOO based on Pareto solution set

In the field of bridge design, the MOO problem can be constructed by constructing a functional relationship equation to clarify the logical connection between the OOs, and then construct a mathematical expression model based on the unified independent variables. The method of transforming a multi-objective problem into a mathematical model is an effective problem-solving strategy. Theoretically, all multi-objective problems can be solved by constructing a mathematical model, which provides a generalized solution to the problem [15-16]. The mathematical expression of an MOO problem usually consists of an objective function (OF) and constraints that satisfy specific conditions. The mathematical expression is shown in Equation (1).

$$\min F(x) = (f_1(x), f_2(x), ..., f_m(x)) \quad (1)$$

In Equation (1), $x \in \sigma \subset R^n$, $f_i(x)$ represent the sub-objectives that need to be optimized simultaneously. There is no direct consistency between them. The space consisting of the m -dimensional vector $(f_1(x), f_2(x), ..., f_m(x))$ is called the OF space, while $g_i(x) \le 0, (i = 1, 2, ..., p)$ is the constraints that must be satisfied during the model solution process. The MOO problem explored in the study refers specifically to the optimization problem in bridge design. That is, given the resources required for bridge design, how to achieve the relative OS of these objectives while satisfying the constraints of schedule, safety, quality, and cost [17-18]. Therefore, the goal of the study is to coordinate the various objectives in bridge design to achieve the best possible optimization under the given conditions, with the goal of achieving the best completion of the entire design project. When dealing with the MOO problem, various strategies can be used, such as the dominant objective strategy, the weighted sum method, and the Pareto efficiency method, especially the weighted sum method, as shown in Equation (2).

$$\min_{x \in X} \sum_{i=1}^{N} w_i f_i(x) \quad (2)$$

In Equation (2), $f_i(x)$ denotes the i th OO. w_i denotes the corresponding weight coefficient. X denotes the feasible optimization space. N denotes the total number of OOs. For single-objective optimization algorithms, their performance can be evaluated by several metrics, including but not limited to stability, effectiveness, convergence speed, coverage, and robustness. Robustness can be measured by running the algorithm multiple times with varying parameters, recording the OSs, and calculating the standard deviation (SD) of the OF values of these solutions. A smaller SD indicates higher robustness and this assessment is shown in Equation (3).

$$Rob = \sqrt{\frac{\sum_{i=1}^{n}(\frac{\sum_{i=1}^{n} f_i}{n} - f_i)^2}{n}} \tag{3}$$

In Equation (3), Rob denotes the measure of robustness, which reflects the consistency of the algorithm's performance under different parameter settings. f_i denotes the OF value of the OS found in i runs. n is the total runs. Convergence is evaluated by determining the convergence stage of the algorithm and calculating the SD of the OF value of the solution at that stage, as shown in Equation (4).

$$CON = \sqrt{\frac{\sum_{i=b}^{t}(\frac{\sum_{i=b}^{t} f_i}{n} - f_i)^2}{t-b+1}} \tag{4}$$

In Equation (4), CON represents the convergence measure. t is the total solutions found during the search. The solution found in the b th search is the value of the OF found by the optimization search. When evaluating the performance of the MOO algorithm, the main considerations are the set of Pareto OSs it generates and the time cost required [19-20]. To eliminate the variance of different objective units, the max-min normalization method is usually used. The objective vectors of the Pareto solutions are converted to standardized values, and then the evaluation metrics are calculated based on these normalized values (NV), as shown in Equation (5).

$$F_i^j = \frac{f_i^j - f_{\min}^j}{f_{\max}^j - f_{\min}^j} \tag{5}$$

In Equation (5), F_i^j is the NV of the i th solution of the j th objective. $f_{\max}^j$ denotes the actual maximum value of the j th objective in the algorithm's optimization process. f_i^j denotes the actual value of the j th objective for the i th solution. $f_{\min}^j$ denotes the actual minimum value of the j th objective obtained by the algorithm during the optimization process. In bridge design, in addition to evaluating the effectiveness of the algorithm, special attention must be paid to potential failure situations. If the OS obtained by the algorithm meets the user's requirements for accuracy, it can be considered a satisfactory solution [21-23]. Equation (6) illustrates how the relative difference between the OS and the genuine OS's OF value can be used to assess and compute the OS's quality.

$$\delta = \frac{|f' - f^*|}{f^*} \times 100\% \tag{6}$$

In Equation (6), δ is the quality of the OS. f' and f^* represent the OF values of the OS and the true OS obtained by the algorithm, respectively. The ability of the algorithm to find a satisfactory solution in finite time is called a successful optimization run. The ratio of the successes in multiple runs of the algorithm to the total runs is called the success rate, as shown in Equation (7).

$$\beta = \frac{N_{success}}{N_{all}} \times 100\% \tag{7}$$

In Equation (7), β denotes the success rate. N_{all} is the total optimization runs. $N_{success}$ is the successful runs. In the Pareto solution set, if the research considers two objectives f_1 and f_2 that need to be optimized, as shown in Figure 1.

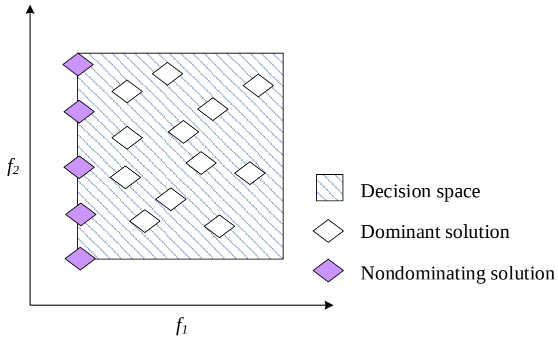


Figure 1: Definition of Pareto OS in multi-objective problems

In Figure 1, the non-dominated solutions on the boundary of the feasible domain are the Pareto-OSs, which provide multiple possibilities for trade-offs between different objectives and offer rich choices for decision makers. Within the MOO bridge design framework, the mathematical modeling of the project schedule is based on the accumulation of the time required for each construction phase, which is developed and refined through incremental refinement. The study starts by identifying all possible paths in the flowchart and calculating the elapsed time of these paths. All the factors that may affect the construction schedule are also considered, and finally a mathematical model of the duration target is established with the shortest total duration as the goal. The model is expressed as shown in Equation (8).

$$\begin{cases} \min T = \sum_{i \in I} t_u \\ s.t \quad t_{su} \le t_u \le t_{Iu} \end{cases} \tag{8}$$

In Equation (8), T represents the total construction time of the project. I represents the set of construction phases contained on the critical path in the construction flowchart. t_{Iu} is the shortest construction time of the stage. t_u is the actual construction time of the stage. t_{Iu} is the longest construction time for stage u . In the MOO study of bridge design, the construction of cost-effective optimization model is a key aspect. In constructing the cost optimization model, the actual duration of each construction stage is taken as the independent variable. Direct costs (DCs) cover costs directly related to construction, such as material costs, labor costs and equipment costs. Indirect costs (IDC), on the other hand, include costs that are not directly involved in production activities but are indispensable in the operation of the project, such as management fees and review fees. The relationship between DCs, IDCs and process time is shown in Figure 2.

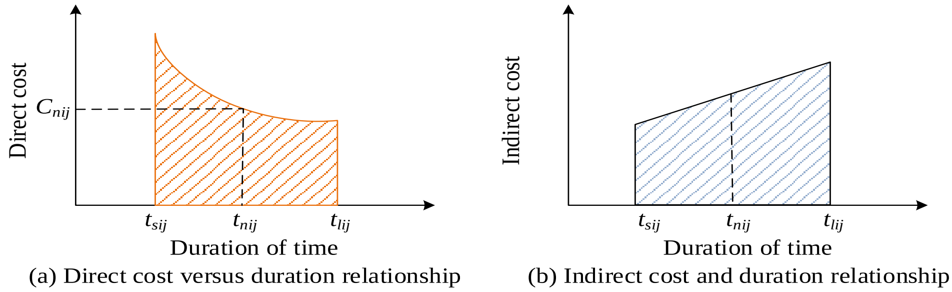


Figure 2: The relationship between DC, IDC and process time

In Figure 2, the relationship between DC and duration takes the form of a quadratic function. DCs are inversely proportional to the duration of the construction phase, but there is a minimum point. Beyond this point, costs increase due to labor and equipment idleness, depreciation, and other factors. IDCs are directly proportional to the duration of the construction phase and increase as construction time increases. The study completes the building of the construction phase cost optimization model by combining the relationship between direct and IDCs and construction duration. Equation (9) displays the created cost target model.

$$\min C = \sum_{i=1}^{n} (C_i + \lambda_i (t_i - t_{in})^2) + \varphi T_c \tag{9}$$

In Equation (9), C_i is the DC of the process at normal duration. λ_i is the marginal incremental factor associated with the cost. t_i is the actual construction time of the process. φ denotes the IDC per day. t_{in} denotes the theoretical construction time. T_c while denotes the total duration of the entire project. In the MOO framework for bridge design, ensuring the quality of the project is a crucial aspect. For this reason, the study proposes a method to assess the quality level based on the construction network diagram. In this method, each network node represents a process and its quality level is determined by a specific formula. This is shown in Figure 3.

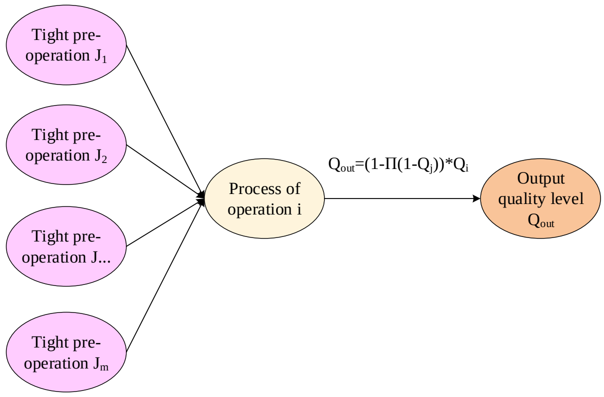


Figure 3: Construction network planning node

In Figure 3, Q_j is the quality level index of the initial input network node. For the intermediate processes in the network diagram, the calculation of their quality level needs to consider the effects of all immediately preceding processes. The immediately preceding process of process i is denoted by J_m. Where m is a natural number, the quality level of the immediately preceding process is Q_i. The output quality level of process i is Q_{out}. Ultimately, the quality level Q of the whole bridge design project is determined by the quality level Q_{out} of the outputs of all processes combined. The quality level of the final output process is shown in Equation (10).

$$Q = Q_{out}^{n} = (1 - \prod_{i=1}^{n} (1 - Q_{out})) * Q_n \tag{10}$$

In Equation (10), Q denotes the total quality level of the whole project, while Q_n denotes the quality level of the last process.

Through this method, it can ensure that the quality of each process is strictly controlled and optimized during the design and construction process, thus improving the quality of the whole bridge design. The OF is defined as a multi-dimensional vector covering the three key areas of cost, safety and construction time, which are linked by constructed mathematical models designed to find the best trade-offs between these objectives. These objectives and constraints interact in the Pareto solution space and are identified by non-dominant ordering, i.e., a solution is considered non-dominant if it is not inferior to another solution on all objectives and is superior on at least one objective. This sorting mechanism ensures that the solution set found in MOO can balance the trade-offs between the various objectives and increase the practicality of the solution.

2.2 IACA-based MOO system construction

The study explores the bridge design MOO problem based on Pareto solution set and provides a systematic solution for bridge design through mathematical modeling and optimization strategies. Next, it will introduce how to construct an MOO system through IACA to further improve the overall performance and optimization of bridge design. It is assumed that there are M ants searching for food, the probability of the k th ant transferring from node p to node q at each iteration t times of them is shown in Equation (11).

$$p_{pq}^{k}(t)=\begin{cases}\dfrac{\tau_{pq}^{\alpha}\cdot\eta_{pq}^{\beta}}{\sum_{q\in allowed_k}\tau_{pq}^{\alpha}\cdot\eta_{pq}^{\beta}}, q\in allowed_k \\ 0, other\end{cases} \quad (11)$$

In Equation (11), τ_{pq} denotes the initial pheromone (IP) between nodes. β is the expectation heuristic factor (EHF). $allowed_k$ denotes the set of next nodes. η_{pq} denotes the heuristic information. α denotes the information heuristic factor. This probability depends on the information heuristic factor and the set of next nodes, and is also influenced by the EHF, which reflects the visibility of the paths between nodes, as well as the values of the heuristic information and the IP [24-26]. The heuristic information η_{pq} is specifically shown in Equation (12).

$$\eta_{pq}=\frac{1}{d_{pq}} \quad (12)$$

In Equation (12), d_{pq} denotes the Euclidean distance between nodes. The IP evaporates as the iteration proceeds to the $t+1$ th iteration. The IP is shown in Equation (13).

$$\begin{cases}\tau_{pq}(t+1)=(1-\rho)\tau_{pq}+\Delta\tau_{pq}(t) \\ \Delta\tau_{pq}(t)=\sum_{k=1}^{M}\Delta\tau_{pq}^{k}(t)\end{cases} \quad (13)$$

In Equation (13), $\Delta\tau_{pq}^{k}(t)$ denotes the pheromone increment of the k th ant after the t th iteration. ρ denotes the pheromone volatilization factor. Equation (14) illustrates how an enhancement treatment is applied to the optimal path discovered after each iteration to increase its appeal by raising the pheromone in order to improve the efficiency and solution quality of the algorithm.

$$\tau_{pq}(t+1)=(1-\rho)\tau_{pq}+\sum_{k=1}^{M}\Delta\tau_{pq}^{k}(t)+\Delta\tau_{pq}^{*}(t) \quad (14)$$

In Equation (14), $\Delta\tau_{pq}^{k}(t)$ is modeled as an ant-week system, as shown in Equation (15).

$$\Delta\tau_{pq}^{*}(t)=\begin{cases}\sigma\cdot\dfrac{Q}{L_{best}}, Ant\ path(\text{p,q}) \\ 0, otherwise\end{cases} \quad (15)$$

In Equation (15), L_{best} denotes the optimal path length. σ denotes the number of elite ants. The flow of ACA is shown in Figure 4.

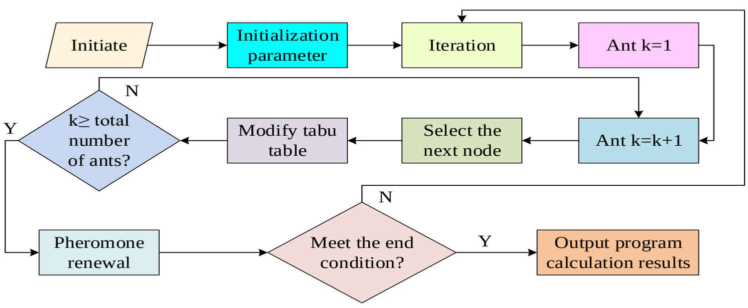
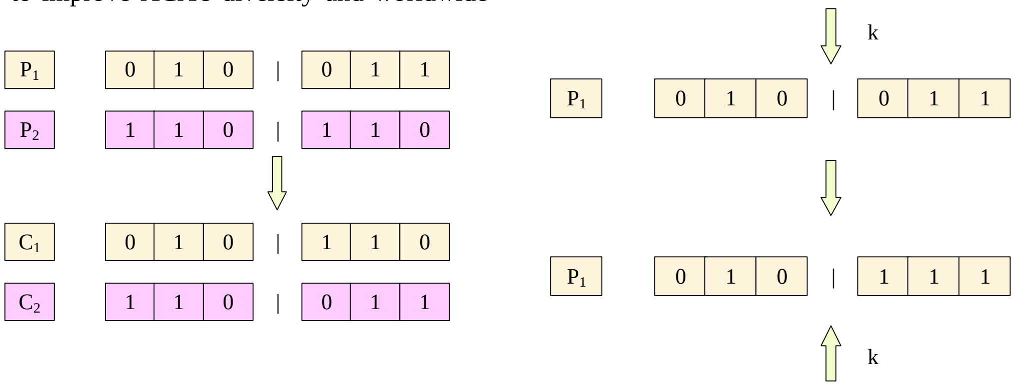


Figure 4: Ant colony algorithm flow chart

In Figure 4, the algorithm starts with ants selecting nodes to pass through based on a specific probability distribution and leaving pheromone traces on the selected paths. As the algorithm iterates, the accumulation of pheromone makes the ants increasingly inclined to select paths that already have a high pheromone concentration due to a positive feedback mechanism. While this phenomenon can facilitate fast convergence, it can also lead to a concentration of the search process on a small number of paths, thus increasing the risk of falling into LO. To get around this restriction, the research adds GA components to improve ACA's diversity and worldwide search capabilities. By combining the advantages of the two algorithms, not only the solution speed is accelerated, but also the problem of premature convergence of the algorithm to a local optimum solution is effectively avoided. This improved algorithm increases the possibility of exploring new potential solutions by mimicking natural selection and genetic mechanisms in the MOO problem of bridge design. This enables a more comprehensive search of the solution space to find a more balanced and optimized design solution. The chromosome crossover and mutation process of GA is shown in Figure 5.



(a) Chromosome crossing process (b) Chromosomal variation process

Figure 5: Chromosome crossover and mutation process of genetic algorithm

In Figure 5, the mutation operation in GA injects the necessary genetic diversity into the algorithm by implementing small random adjustments on the coding strings of the solutions. This effectively avoids the stagnation of the algorithm on the local OS and promotes the in-depth exploration of the global solution space. Meanwhile, the combination of mutation and recombination operations provides the GA with an efficient navigation capability in the solution space to find global or near-global OSs. The IACA-based MOO process is shown in Figure 6.

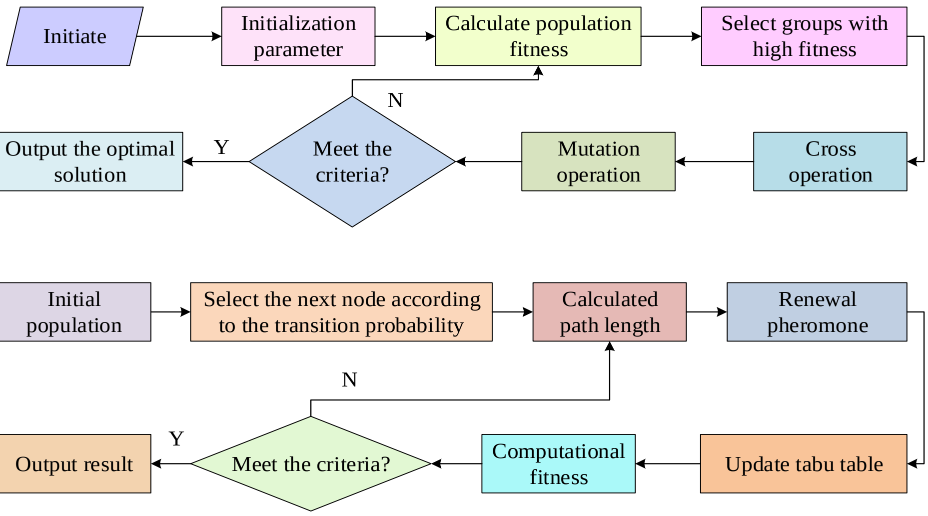


Figure 6: MO optimization process based on improved ant colony algorithm

In Figure 6, in the initial stage, the algorithm performs parameter setting and the construction of fitness function. Subsequently, the fitness is calculated and high fitness individuals are screened by genetic iteration, and the population is optimized by applying OX crossover method and mutation operation. Based on the fitness ordering, the top n individuals are chosen to build a new generation population by combining the optimized and original populations. Next, check whether the termination condition is met. If it is not met, iteration continues, and if it is met, the most optimal population is applied to ACA. In the ACA phase, the ants select paths based on specific formulas and update the pheromone and contraindication tables. Eventually, if the ants complete the search and the result satisfies the output condition, the algorithm terminates and outputs the result, otherwise, the iteration continues until a solution that satisfies the condition is found. In IACA, the choice of parameters is very important for the exploration ability and convergence performance of the algorithm. The pheromone volatility factor controls the decay rate of the pheromone, which is set to 0.1. A lower volatility factor helps maintain the persistence of the pheromone, thus promoting global exploration in the early stages of the algorithm and avoiding premature convergence. The pseudo-random factor affects the balance between randomness and pheromone intensity guidance when the ants choose the path, and is set to 0.9, which allows the algorithm to use the pheromone while maintaining enough randomness to explore new paths and increase the diversity of the algorithm. The detailed IACA process is shown in Figure 7.

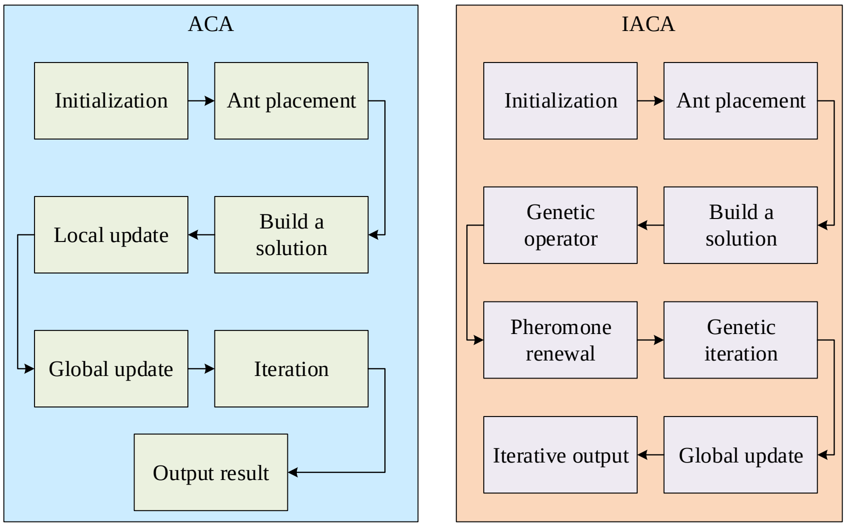


Figure 7: Detailed flow chart of IACA

In Figure 7, IACA adds the operation of GA based on traditional ACA, as well as the improvement of pheromone update mechanism, which helps to improve the search ability and the quality of the algorithm.

3 Results

The study first validates the performance of IACA through a series of experiments, including the convergence of the algorithm, precision rate, recall rate, and comparing the results of the error drop rate. Finally, the study evaluates the performance of the proposed bridge design MOO system and compares it with the traditional MOO system in various metrics.

3.1 IACA-based performance experiments

The effectiveness of the design process and the caliber of the output are directly impacted by the suitability of parameter setup in the field of bridge design. The parameters in the optimization method must be carefully chosen, taking into account both the particular requirements of the design project and the limitations of the algorithm itself, when it is used to solve the actual multi-objective bridge design problem. The experiments are conducted in a computing environment equipped with an Intel Core i7 processor and 16 GB RAM, the operating system is Windows 10, all algorithms are implemented in Python 3.8 environment, using NumPy and SciPy libraries. The experiment is repeated 30 times to ensure the stability of the results, and the parameter settings are consistent for each run. To ensure the replicability of the study, the data pre-processing steps include data cleaning to remove missing and outliers, and data standardization to ensure consistency of algorithmic inputs. The hyperparameter tuning process uses a grid search strategy to systematically traverse the predefined parameter space to find the optimal parameter combination. For each iteration of the algorithm, the setting of the initial conditions follows a randomization process in which the initial position of the ant and the initial concentration of the pheromone are randomly generated to ensure independence of each iteration and diversity of results. Table 2 displays the parameter settings.

Table 2: Parameter settings

Parameter name	Symbols	Parameter value
Pheromone volatile factor	ρ	0.1
Population size	m	80
Pseudo-random factor	q_0	0.9

Pheromone heuristic factor	α	2
Expectation heuristic factor	β	3
Pheromone local volatile factor	ξ	0.1
Number of iterative updates	GEN	100
Pheromone concentration	τ_0	0.8

For the purpose of balancing exploration and exploitation, Table 2's population size is set to 80. The heuristic factors α and β are set to 2 and 3, respectively, which guide the ants to avoid LO and effectively utilize pheromones during the search process. The pheromone volatilization factor is set to 0.1 to maintain a moderate volatilization of pheromone concentration and facilitate global search. The iterations is set to 100 to ensure that the algorithm converges within a limited number of iterations. The IP concentration and the local volatilization factor are set to 0.8 and 0.1, respectively, which accelerate the initial exploration and maintain the global search capability. The pseudo-randomization factor is set to 0.9 to ensure that the algorithm is both fast and accurate in the solution process. genetic algorithm improves ant colony optimization (GA-ACO) is compared with ACO, GA, SA, PSO for comparative analysis. The convergence of the five algorithms is shown in Figure 8. In Figure 8, the GA-ACO algorithm reaches the OS and the OF value is minimized at the 18th iteration, and no further change occurs after that. This shows that the GA-ACO algorithm not only converges quickly, but also has good stability. It can find the OS of the problem in a shorter time, and can keep this state stable after finding the OS. This fast convergence property is very important for solving the MOO problem. It enables the algorithms to handle complex optimization tasks with high efficiency and reliability. The precision and recall of the five algorithms are shown in Figure 9.

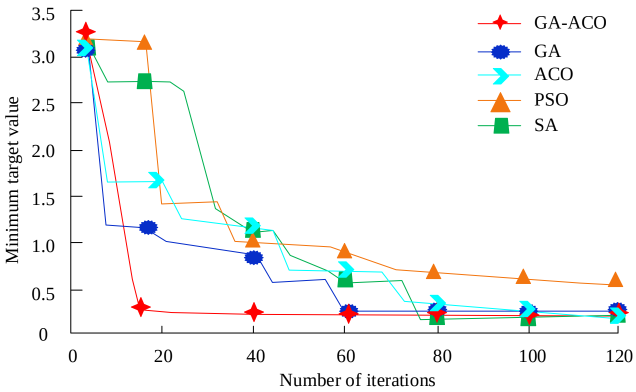
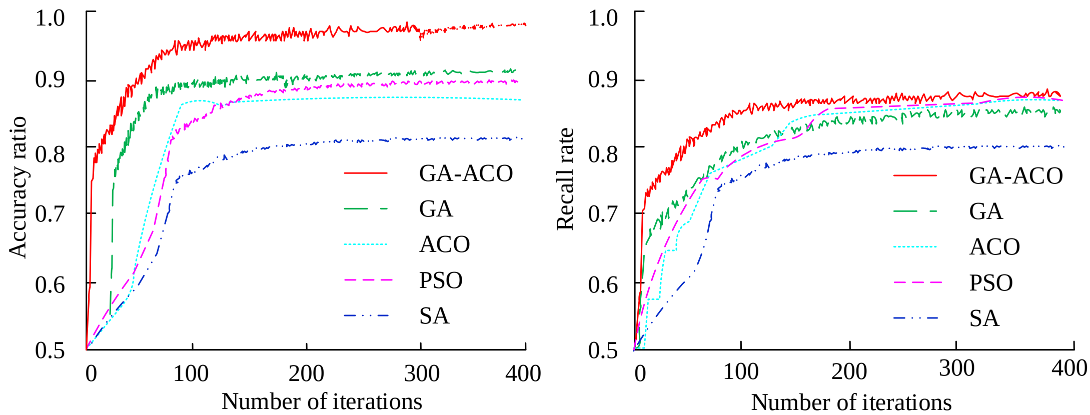


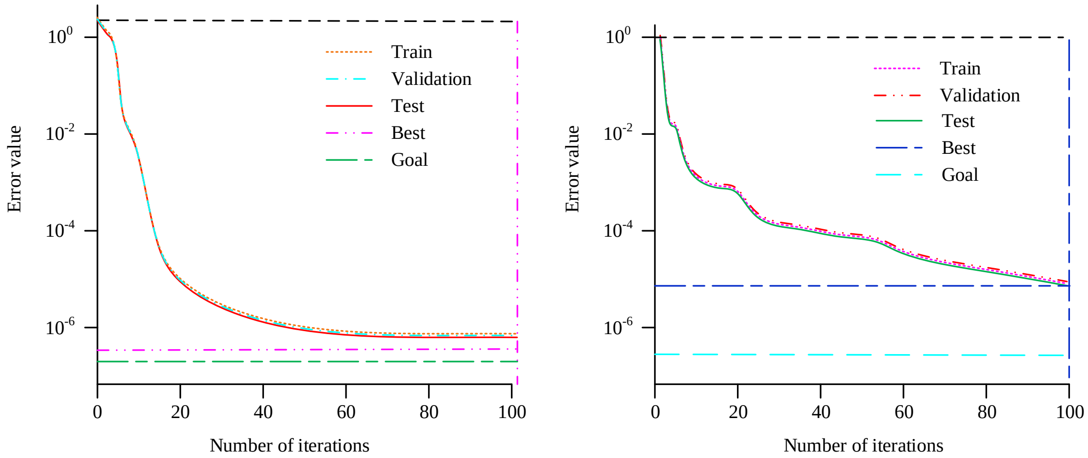
Figure 8: Convergence of five algorithms



(a) Accuracy ratio (b) Recall rate

Figure 9: Comparison of accuracy rate and recall rate of five algorithms

In Figure 9(a), the accuracy ratio of the GA-ACO algorithm reaches more than 0.9 in the first 50 iterations or so, and eventually stabilizes at around 0.98. In Figure 9(b), the recall ratio of GA-ACO algorithm is also higher than the other four algorithms, and eventually stabilizes at around 0.90. It shows that the GA-ACO algorithm not only has the property of fast convergence in the MOO problem, but also performs well in the two key performance indexes of precision and recall. It proves the efficiency and reliability of GA-ACO algorithm in solving complex optimization problems. A comparison of the error drop rate results between the GA-ACO algorithm and the ACO algorithm is shown in Figure 10.



(a) Error decline rate of GA-ACO algorithm

(b) Error decline rate of ACO algorithm

Figure 10: Error decline rate comparison

Figure 10(a) and Figure 10(b) demonstrate the iterations required by the GA-ACO algorithm versus the traditional ACO algorithm in reaching the target value. The GA-ACO algorithm has reached the target value after 50 iterations, while the ACO algorithm requires 100 iterations to reach the target value for the first time. This shows that the GA-ACO algorithm has a faster convergence rate. By including the GA mechanism, this enhanced GA-ACO algorithm enhances the efficiency of the algorithm and optimizes the ACA search process. Due to the reduction in the iterations, the algorithm is more efficient in learning and exploring the solution space, which helps to quickly identify the key features and patterns of the problem. To enhance the rigor of the results in comparison with other algorithms, statistical significance tests are employed to ascertain whether the observed performance differences are statistically significant. Concurrently, confidence intervals are calculated to furnish a range of uncertainty for performance comparisons. The specifics are presented in Table 3.

Table 3: Significance tests and confidence interval statistics

Performance index	**Rate of convergence**	**Accuracy ratio**	**Recall rate**
GA-ACO mean	0.95	0.98	0.90
ACO mean	0.70	0.55	0.55
Standard deviation of GA-ACO	0.02	0.01	0.02
Standard deviation of ACO	0.05	0.10	0.08
T-test result (*P*-value)	<0.05	<0.05	<0.05
Confidence interval (95%)	(0.10, 0.20)	(0.03, 0.07)	(0.05, 0.10)

In Table 3, the P-value in the T-test results is less than 0.05, indicating that the difference between GA-ACO and ACO on this measure is statistically significant. A confidence interval provides a range of uncertainty for comparing algorithm performance, and a 95% confidence interval means that there is 95% confidence that the true difference lies within that interval. These statistical methods make performance comparisons more precise and ensure that the conclusions drawn are not due to random variation. Their ability to evaluate the performance of different algorithms with greater confidence and to determine that new algorithms are statistically significantly better than existing ones.

3.2 Performance evaluation of MOO systems for bridge design

The study concludes by analyzing the performance of the bridge design MOO system in real situations. The bridge design MOO system proposed in the study (System 1) is compared with the conventional MOO system (System 2). The metrics include optimization speed and so on, and the metrics are normalized. To evaluate the robustness of MOO algorithms, the process entails running the algorithm on multiple occasions and recording the OS obtained on each occasion. The specific method is to collect the OF values of the OS in each run and then calculate the standard deviation of these values to measure the consistency of the algorithm's performance under different running conditions. The smaller the standard deviation, the higher the robustness of the algorithm, i.e. the better the stability of the algorithm in different operations. Table 4 displays the ultimate outcomes.

Table 4: Comparison of indicators of system 1 and system 2

Indicators optimization speed	System 1	System 2
Objective function value	0.95	0.70
Robustness mass of solution	0.92	0.75
Success rate	0.96	0.65
The convergence diversity resource consumption	0.94	0.72
Indicators optimization speed	0.93	0.68
Objective function value	0.91	0.63
Robustness mass of solution	0.97	0.60
Success rate	0.90	0.80

In Table 4, the optimization speed of System 1 is 0.95, which is significantly better than System 2's 0.70, indicating that System 1 has a significant advantage in convergence speed. The OF value of System 1 is 0.92, which is better than System 2's 0.75, indicating that System 1 is able to achieve a solution closer to the optimum when solving the optimization problem. System 1 outperforms System 2 in all evaluation metrics, proving its superior performance in the bridge design MOO problem. The accuracy, stability, and safety of System 1 and System 2 are specifically shown in Figure 11.

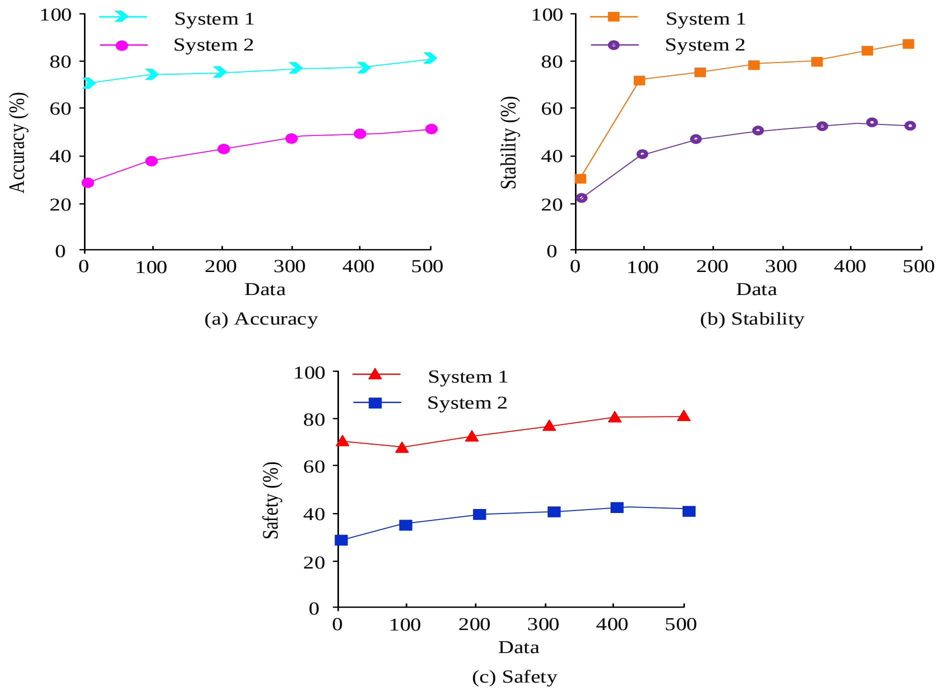


Figure 11: System accuracy, stability, security comparison

In Figure 11, the accuracy, stability, and security of System 1 can reach up to 92%, 95%, and 91%, while that of System 2 is only about 55%. It shows that System 1 has a high correctness rate in identifying and processing the bridge design optimization problem. It is able to maintain consistent performance under different operating conditions or when faced with different datasets. Moreover, it is able to comply well with safety norms and standards in the design process to reduce potential risks.

4 Discussion

Compared with ACA and other advanced algorithms, such as the MOO algorithm proposed by Pereira et al. [5], the elite non-dominance sorting and congestion distance mechanism algorithm proposed by Jangir et al. [6], and the dominance principle and heuristic algorithm of congestion distance evaluation mechanism proposed by Rao et al. [8], the IACA was more effective than the ACA. It showed significant performance improvement. Specifically, IACA achieved an optimization speed of 0.95, showing a faster convergence rate compared to 0.70 in Pereira et al.'s algorithm and 0.65 in Jangir et al.'s algorithm. In terms of robustness, the standard deviation of IACA was 0.02, which was lower than the 0.04 of Rao et al.'s algorithm, indicating that it had higher stability and consistency over multiple runs. IACA also excelled in solution accuracy, with an accuracy rate of 0.98, higher than Jangir et al.'s 0.80 and Rao et al.'s 0.85. The IACA greatly enhanced the global search capability and diversity of the algorithm by integrating key elements of GAs, such as chromosome crossing and mutation mechanisms. This enhanced search capability allowed IACA to more effectively avoid local optimizations and thus found better solutions in MOO problems. In addition, the concept of negative pheromones introduced into the IACA helped to suppress the influence of bad solutions, further enhancing the robustness of the algorithm. The IACA provided a new optimization strategy to achieve more efficient, economical, and safer bridge design solutions. These improvements were not only innovative in theory, but also had important technical value in practice, providing a new solution to bridge design and related engineering optimization problems.

5 Conclusion

Aiming at the increasing demand of MOO in the field of bridge design, the study proposed an IACA-based MOO system for bridge design to improve the design efficiency and quality. The results of the study indicated that the accuracy ratio of the GA-ACO algorithm reached more than 0.9 in the first 50 iterations or so and eventually stabilized at about 0.98. The GA-ACO algorithm achieved the target value after 50 iterations, while the ACO algorithm required 100 iterations to reach the target value for the first time. The research successfully developed an IACA-based MOO system for bridge design, which outperformed the conventional optimization system in several evaluation metrics. By introducing the mechanism of GA, the new system demonstrated significant performance improvement in terms of optimization speed, OF value, and robustness. Specifically, the accuracy, stability, and security of System 1 reached 92%, 95%, and 91%, respectively, much higher than that of System 2 at 55%. Although IACAs showed faster convergence and better optimization performance in experiments, their increased complexity could lead to challenges in parameter tuning and computational resource requirements, and there was a risk of overfitting on smaller datasets. In addition, while the current study demonstrated the effectiveness of IACA on specific bridge design problems, its scalability and applicability to more complex bridge design problems or larger data sets needed to be further validated and investigated. These considerations provide directions for future improvement and application of the algorithm, ensuring the comprehensiveness and practicality of the research results. Further research could investigate the application of the algorithm in diverse infrastructure contexts, including high-rise structural design and transportation network optimization. Additionally, the adaptability of the algorithm to dynamic environmental changes and its performance on large-scale datasets warrant further examination. Furthermore, the automatic parameter adjustment mechanism of the algorithm can be subjected to further study with a view to enhancing its generalization ability and robustness. This would provide a clear development direction and practical application guidance for subsequent research.

References

[1] Abdollahzadeh B, Gharehchopogh F S. A multi-objective optimization algorithm for feature selection problems. Engineering with Computers, 2022, 38(3): 1845-1863. Doi:10.1007/s00366-021-01369-9.

[2] Sadeghi Hesar A, Kamel S R, Houshmand M. A quantum multi-objective optimization algorithm based on harmony search method. Soft Computing, 2021, 25(14): 9427-9439. Doi:10.1007/s00500-021-05799-x.

[3] Hong W J, Yang P, Tang K. Evolutionary computation for large-scale multi-objective optimization: A decade of progresses. International Journal of Automation and Computing, 2021, 18(2): 155-169. Doi:10.1007/s11633-020-1253-0.

[4] Karmakar K, Das R K, Khatua S. An ACO-based multi-objective optimization for cooperating VM placement in cloud data center. The Journal of Supercomputing, 2022, 78(3): 3093-3121. Doi:10.1007/s11227-021-03978-z.

[5] Pereira J L J, Oliver G A, Francisco M B, Cunha Jr S S, Gomes G F. A review of multi-objective optimization: methods and algorithms in mechanical engineering problems. Archives of Computational Methods in Engineering, 2022, 29(4): 2285-2308. Doi:10.1007/s11831-021-09663-x.

[6] Jangir P, Buch H, Mirjalili S, Manoharan P. MOMPA: Multi-objective marine predator algorithm for solving multi-objective optimization problems. Evolutionary Intelligence, 2023, 16(1): 169-195. Doi:10.1007/s12065-021-00649-z.

[7] Premkumar M, Jangir P, Sowmya R, Alhelou H H, Mirjalili S, Kumar B S. Multi-objective equilibrium optimizer: Framework and development for solving multi-objective optimization problems. Journal of Computational Design and Engineering, 2022, 9(1): 24-50. Doi:10.1093/jcde/qwab065.

[8] Rao R V, Keesari H S. Rao algorithms for multi-objective optimization of selected thermodynamic cycles. Engineering with Computers, 2021, 37(4): 3409-3437. Doi:10.1007/s00366-020-01008-9.

[9] Yang K, Wang Z. Health Monitoring of Civil Engineering Structures Using Simulated Annealing Genetic Algorithm. Informatica, 2024, 48(18). Doi:10.31449/inf.v48i18.6435.

[10] Zhou Y. Structural Damage Identification of Large-Span Spatial Grid Structures Based on Genetic Algorithm. Informatica, 2024, 48(17). Doi:10.31449/inf.v48i17.6428.

[11] Elsedimy E, Algarni F. MOTS-ACO: An improved ant colony optimiser for multi-objective task scheduling optimisation problem in cloud data centres. IET Networks, 2022, 11(2): 43-57. Doi:10.1049/ntw2.12033.

[12] Goel R, Maini R. Improved multi-ant-colony algorithm for solving multi-objective vehicle routing problems. Scientia Iranica, 2021, 28(6): 3412-3428. Doi:10.24200/SCI.2019.51899.2414.

[13] Masoumi Z, Van Genderen J, Sadeghi Niaraki A. An improved ant colony optimization-based algorithm for user-centric multi-objective path planning for ubiquitous environments. Geocarto international, 2021, 36(2): 137-154. Doi:10.1080/10106049.2019.1595176.

[14] Ning J, Zhao Q, Sun P, Feng Y. A multi-objective decomposition-based ant colony optimisation algorithm with negative pheromone. Journal of Experimental & Theoretical Artificial Intelligence, 2021, 33(5): 827-845. Doi:10.1080/0952813X.2020.1789753.

[15] Li X. RREASO Building Structure Physical Parameter Identification Algorithm for Structural Damage Identification. Informatica, 2024, 48(18). Doi:10.31449/inf.v48i18.6156.

[16] Mohammadzadeh A, Masdari M. Scientific workflow scheduling in multi-cloud computing using a hybrid multi-objective optimization algorithm. Journal of Ambient Intelligence and Humanized Computing, 2023, 14(4): 3509-3529. Doi:10.1007/s12652-021-03482-5.
[17] Hua Y, Liu Q, Hao K. A survey of evolutionary algorithms for multi-objective optimization problems with irregular Pareto fronts. IEEE/CAA Journal of Automatica Sinica, 2021, 8(2): 303-318. Doi:10.1109/JAS.2021.1003817.
[18] Zhai L, Yan X, Liu G. Cost Impact Factors and Control Measures of Road and Bridge Projects Based on Linear Regression Model. Informatica, 2024, 48(17). Doi:10.31449/inf.v48i17.6370.
[19] Sharma S, Kumar V. A comprehensive review on multi-objective optimization techniques: Past, present and future. Archives of Computational Methods in Engineering, 2022, 29(7): 5605-5633. Doi:10.1007/s11831-022-09778-9.
[20] Hao X. Intelligent User Experience Design in Digital Media Art under Internet of Things Environment. Informatica, 2024, 48(15). Doi:10.31449/inf.v48i15.6405.
[21] Zhang D, Luo R, Yin Y, Zou S L. Multi-objective path planning for mobile robot in nuclear accident environment based on improved ant colony optimization with modified A∗. Nuclear Engineering and Technology, 2023, 55(5): 1838-1854. Doi: 10.1016/j.net.2023.02.005.
[22] Abualigah L, Diabat A. A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments. Cluster Computing, 2021, 24(1): 205-223. Doi:10.1007/s10586-020-03075-5.
[23] De R, Nanda I. Network/Security Threats and Countermeasures For Cloud Computing. Acta Electronica Malaysia. 2022; 7(1): 01-03. http://doi.org/10.26480/aem.01.2022.01.03
[24] Sun G, Li J, Liu Y, Liang S, Kang H. Time and energy minimization communications based on collaborative beamforming for UAV networks: A multi-objective optimization method. IEEE Journal on Selected Areas in Communications, 2021, 39(11): 3555-3572. Doi:10.1109/JSAC.2021.3088720.
[25] Singh O, Rishiwal V, Chaudhry R, Yadav M. Multi-objective optimization in WSN: Opportunities and challenges. Wireless Personal Communications, 2021, 121(1): 127-152. Doi:10.1007/s11277-021-08627-5.
[26] Choudhuri S, Adeniye S, Sen A. Distribution Alignment Using Complement Entropy Objective and Adaptive Consensus-Based Label Refinement for Partial Domain Adaptation. Artificial Intelligence and Applications. 2023, 1(1): 43-51. Doi:10.47852/bonviewAIA2202524.