Postprint: An Improved Bird Swarm Algorithm for Mobile Cloud Computing Task Scheduling

Authors: Chen Xuan, Zhao Wenjun, Long Dan

Date: 2020-09-28T00:00:00+00:00

Abstract

To address the issues of prolonged execution time and high device energy consumption in task scheduling for mobile cloud computing environments, a task scheduling strategy based on an improved bird swarm algorithm (IBSA) is proposed. First, a mobile cloud task scheduling model focusing on energy consumption and time is constructed. Second, adaptive perception coefficients and social coefficients are proposed to prevent the algorithm from falling into local optima; a learning factor is constructed to optimize flight behavior, ensuring individual search capability. Finally, the task scheduling objective function participates in the iterative update of the algorithm as the fitness function of bird swarm individuals. Simulation results demonstrate that compared with the ant colony algorithm, particle swarm algorithm, whale algorithm, and bird swarm algorithm, this algorithm achieves favorable performance in mobile cloud computing task scheduling, effectively saving time and reducing energy consumption.

Full Text

Research on Improved Bird Swarm Algorithm for Mobile Cloud Computing Task Scheduling

Chen Xuan[1], Zhao Wenjun[2], Long Dan[3]
[1]Zhejiang Industry Polytechnic College, Shaoxing, Zhejiang 312000, China
[2]Beijing Information Science & Technology University, Beijing 100101, China
[3]Zhejiang University, Hangzhou 310058, China

Abstract

Task scheduling in mobile cloud computing environments suffers from excessive time consumption and high device energy costs. To address these issues, this paper proposes a task scheduling strategy based on an Improved Bird Swarm Algorithm (IBSA). First, we construct a mobile cloud task scheduling model that primarily considers energy consumption and execution time. Second, we introduce adaptive sensing coefficients and social coefficients to prevent the algorithm from falling into local optima. We also optimize flight behavior through learning factors to ensure individual search capabilities. Finally, the task scheduling objective function serves as the fitness function for bird swarm individuals during iterative updates. Simulation results demonstrate that compared with the ant colony algorithm, particle swarm algorithm, whale algorithm, and standard bird swarm algorithm, the proposed algorithm achieves superior performance in mobile cloud computing task scheduling, effectively reducing both time consumption and energy expenditure.

Keywords: mobile cloud computing; bird swarm algorithm; adaptive; learning factor

0 Introduction

Wireless mobile computing enables people to work across geographic regions using portable devices and assists enterprises in resource allocation. However, due to constraints in battery capacity, storage, and computing power, this approach cannot compete with traditional servers and desktop devices—mobile hardware capabilities remain extremely limited. Mobile cloud computing technology has emerged as a promising solution to these challenges. Unlike conventional cloud computing, nodes in mobile cloud computing consist of numerous mobile terminals, whose attributes such as storage capacity, network connectivity status, and battery level affect overall cloud task scheduling.

Previous studies have focused on either task completion time or device energy consumption individually. Some researchers have considered joint optimization: for instance, literature [11] proposed a CCS algorithm for multi-task flows that improves task processing rates and reduces response times by increasing concurrency between task transmission and execution and enhancing task set integration probability, though it lacks comparison with other scheduling algorithms. Literature [12] examined a task scheduling model and objective function for joint execution time optimization, using simulated annealing to reduce time and energy consumption but at the cost of increased algorithmic complexity. Literature [13] employed a linear-time scheduling algorithm based on task migration for mobile cloud computing, though its practical effectiveness requires further validation. Literature [14] proposed an optimal task workflow scheduling scheme based on the whale algorithm, but this increased algorithm runtime. Literature [15] introduced a collaborative multi-task scheduling scheme based on ant colony optimization, yet it lacked comparison with newer bio-inspired algorithms. Literature [16] utilized bio-inspired algorithms to solve task scheduling problems for user mobile devices, demonstrating good scheduling performance in terms of energy efficiency and response time.

These studies indicate that bio-inspired algorithms can enhance task scheduling effectiveness in mobile cloud computing. Inspired by this observation, this paper applies an optimized bird swarm algorithm to mobile cloud computing task scheduling, specifically investigating task execution time and device energy consumption.

1.1 Mobile Cloud Computing Task Scheduling

The objective of task allocation in mobile cloud computing environments is to reasonably distribute N parallelizable subtasks among M mobile devices to minimize task execution time and system power consumption while improving execution efficiency and reducing overall system energy usage. Based on the task time and device energy calculation methods described in literature [17], this paper constructs both task time fitness functions and device energy fitness functions.

Let $T = \{T_1, T_2, ..., T_N\}$ represent the task set, where user-submitted tasks are divided into N subtasks that can be processed in parallel.

a) Task Completion Time Fitness Function

Mobile resources in cloud computing environments exhibit dynamic and heterogeneous characteristics. The same task may require different completion times on different mobile devices, depending on the device' s computing capability. Typically, for a task T_i assigned to device D_j, its execution time t^e_{ij} depends on the task' s length L_i and the device' s CPU processing capability C^{cpu}_j (in MIPS). The execution time is expressed as:

$$t^e_{ij} = \frac{L_i}{C^{cpu}_j} \times u_{cpu} \quad (1)$$

where u_{cpu} represents the mobile device' s CPU usage rate. Note that this consideration is based on the fact that CPU consumption time in both idle and busy states on mobile devices inevitably affects overall task execution time.

In addition to computing capability, task completion time also depends on the device's network transmission capability, as different mobile devices have varying network bandwidths that affect both the time to map tasks to the device and the time to return results, thereby influencing total completion time. For task T_i assigned to mobile device D_j, let t^{ts}_{ij} and t^{tr}_{ij} represent the task mapping time and result return time, respectively, which are determined by input data size d^i_{in}, output data size d^i_{out}, and the device' s network bandwidth B^n_j:

$$t^{ts}_{ij} = \frac{d^i_{in}}{B^n_j}, \quad t^{tr}_{ij} = \frac{d^i_{out}}{B^n_j}$$

Therefore, the total time for mobile device D_j to complete task T_i is:

$$t_{ij} = t^e_{ij} + t^{ts}_{ij} + t^{tr}_{ij}$$

The final completion time for the entire task set is determined by the subtask requiring the longest time:

$$T_{ij}^{max} = \max(t_{ij})$$

Thus, the task completion time fitness function is:

$$fit_{time} = \min(T_{ij}^{max})$$

b) Device Energy Consumption Fitness Function

Limited battery power represents a critical constraint in mobile cloud computing environments. Mobile devices are computing resources with finite energy, so task allocation must consider not only energy consumed during task execution but also the device' s baseline energy consumption. Each mobile device must first maintain power for essential control modules and programs, with remaining power allocated to scheduled tasks in the mobile cloud computing environment.

Based on literature [18] describing energy consumption of mobile device hardware modules (CPU, memory, WiFi, etc.), the energy consumption of mobile device D_j when executing task T_i is:

$$e_{ij}^{c} = \beta_{cpu} \times \mu_{cpu}^{ij} \times t_{ij}^{e} + \beta_{mem} \times \mu_{mem}^{ij} \times t_{ij}^{e}$$

where μ_{cpu}^{ij} and μ_{mem}^{ij} represent CPU and memory usage rates, respectively, while β_{cpu} and β_{mem} denote the power consumption coefficients for CPU and memory modules.

In addition to execution energy, data transmission between mobile devices and proxy servers also consumes energy, primarily due to different network connection methods employed by various devices. According to literature [20], mobile device data transmission energy is proportional to transmitted data size. The energy consumption e_{ij}^{d} caused by data transmission is:

$$e_{ij}^{d} = \delta_{net} \times (d_{in}^{i} \times t_{ij}^{ts} + d_{out}^{i} \times t_{ij}^{tr})$$

where δ_{net} is the network transmission module' s power consumption coefficient.

Therefore, the total energy consumption for mobile device D_j completing task T_i is:

$$e_{ij} = e_{ij}^{c} + e_{ij}^{d}$$

The final task completion energy is the sum of energy consumption across all tasks:

$$E_{ij} = \sum e_{ij}$$

Thus, the device energy consumption fitness function is:

$$fit_{energy} = \min(E_{ij})$$

1.2 Bird Swarm Algorithm

The Bird Swarm Algorithm (BSA), proposed by Xian-Bing Meng in 2015, is a novel swarm intelligence algorithm inspired by bird flocking behavior. The algorithm models three behaviors observed during bird foraging: foraging behavior, vigilance behavior, and flight behavior. Through information sharing mechanisms and search strategies, it achieves optimal solutions and demonstrates superior performance compared to ant colony and particle swarm algorithms for optimization problems.

Let the bird swarm population size be N, the search space be D-dimensional, and the position of the i-th bird in the j-th dimension be $X_i = [x_{i,1}, x_{i,2}, ..., x_{i,D}]$.

a) Foraging Behavior

During foraging, each bird individual updates its position based on both personal and population experience:

$$x_{i,j}^{t+1} = x_{i,j}^t + C \times \text{rand}() \times (p_{i,j} - x_{i,j}^t) + S \times \text{rand}() \times (g_j - x_{i,j}^t)$$

where $x_{i,j}^t$ and $x_{i,j}^{t+1}$ represent the i-th bird' s position in the j-th dimension at iterations t and $t+1$, respectively; C and S are the sensing coefficient and social driving coefficient; $p_{i,j}$ is the i-th bird' s best position in the j-th dimension; g_j is the swarm' s best position in the j-th dimension; and rand() denotes a random number between 0 and 1.

b) Vigilance Behavior

Birds moving toward the swarm center inevitably create competition, causing flight obstacles. The position update formula under vigilance behavior is:

$$x_{i,j}^{t+1} = x_{i,j}^t + A_1 \times (\text{mean}_j - x_{i,j}^t) \times \text{rand}() + A_2 \times (p_{k,j} - x_{i,j}^t) \times \text{rand}()$$

where:

$$A_1 = a_1 \times \exp\left(-\frac{pFit_i}{sumFit \times N + \varepsilon}\right)$$

$$A_2 = a_2 \times \exp\left(\frac{pFit_i - pFit_k}{|pFit_i - pFit_k| + \varepsilon} \times \frac{N \times pFit_i}{sumFit + \varepsilon}\right)$$

Here, $p_{k,j}$ is the k-th bird' s position in the j-th dimension, where k is a random positive integer in $[1, N]$ and $k \neq i$; a_1 and a_2 are constants in $[0, 2]$; $pFit_i$ is the i-th bird's best fitness value; mean_j is the average position in the j-th dimension; $sumFit$ is the sum of the swarm' s best fitness values; and ε is a small constant to prevent division by zero. A_1 and A_2 represent indirect and direct influences from the environment when individuals fly toward the swarm center. When $pFit_k$ is better than $pFit_i$, individual i experiences greater interference than k, potentially causing i to also move toward the swarm center.

c) Flight Behavior

During flight, birds may be disturbed by other flying animals and relocate. In new foraging areas, some birds act as producers searching for food, while others act as scroungers following producers. These roles are defined as:

Producer:

$$x_{i,j}^{t+1} = x_{i,j}^t + \text{randn}(0, 1) \times x_{i,j}^t$$

Scrounger:

$$x_{i,j}^{t+1} = x_{i,j}^t + \text{rand}() \times (x_{k,j}^t - x_{i,j}^t) \times FL$$

where $\text{randn}(0, 1)$ represents a Gaussian distribution with mean 0 and standard deviation 1, indicating that scroungers follow producers to find food. $FL \in (0, 2]$ represents the scroungers' following probability, and FQ is a positive integer representing the frequency of birds flying to other locations.

2 Improved Bird Swarm Algorithm for Mobile Cloud Computing Task Scheduling

The standard BSA algorithm suffers from susceptibility to local optima and premature convergence. While literature [22] improved the algorithm' s later-stage performance by introducing population similarity and aggregation concepts with random probability assignment for optimal positions, literature [23] enhanced performance by using current optimal individuals to replace random selection and incorporating step-size weighted averaging, and literature [24] employed migration and mutation strategies to improve local search capability, these approaches either increase algorithmic complexity or show limited convergence improvement.

2.1 Improved Bird Swarm Algorithm

a) Adaptive Sensing Coefficient and Social Coefficient

In foraging behavior, each individual updates its position using both personal and swarm positions. This paper optimizes the sensing coefficient C and social driving coefficient S by assigning different values to ensure distinct flight effects for each bird. The new coefficients are:

$$C = C_{\min} + (C_{\max} - C_{\min}) \times \frac{t}{t_{\max}}$$

$$S = S_{\min} + (S_{\max} - S_{\min}) \times \frac{t}{t_{\max}}$$

where $C_{\min}$ is the minimum sensing coefficient, t is the current iteration number, and $t_{\max}$ is the maximum iteration number. In the early algorithm stages, smaller C and S values emphasize global optimal search. As iterations progress, these values gradually increase to facilitate fine-grained search in later stages, improving convergence precision, helping birds escape local optima, and reducing the probability of missing optimal solutions.

b) Flight Behavior Based on Learning Factors

In flight behavior, producers continue foraging while scroungers probabilistically obtain food from designated producers. This approach neglects positional information among scroungers within subgroups, potentially causing scroungers to fall into local optima alongside producers. Therefore, this paper optimizes the scrounger position update formula by incorporating learning from other scroungers in the subgroup:

$$x_{i,j}^{t+1} = x_{i,j}^{t} + \mathrm{rand}() \times (x_{k,j}^{t} - x_{i,j}^{t}) \times FL \times M$$

where M is the learning factor, defined as:

$$M = \exp\left(\frac{fit'_i}{fit'_g}\right)$$

Here, fit'_i represents the current scrounger individual' s fitness function value, and fit'_g represents the best scrounger individual' s fitness in the current subgroup. This ratio reflects the association degree between the i-th scrounger and the best scrounger within the subgroup. A larger ratio yields a larger M value, indicating the current scrounger is closer to the optimal scrounger, thereby enhancing individual search capability and providing guarantees for subsequent flights.

2.2 Fitness Function

This paper employs both completion time and energy consumption fitness functions as task scheduling objective functions. Each bird individual corresponds to a task scheduling scheme, with the objective function serving as the algorithm' s fitness function. Leveraging the bird swarm algorithm' s inherent strengths, optimal scheduling schemes are obtained through fitness comparisons. However, due to significant differences in value ranges between time and energy fitness functions, normalization adjusts both to the $[0, 1]$ range.

Let $T_{\max}$ and $T_{\min}$ represent the maximum and minimum values of the time fitness function, and $E_{\max}$ and $E_{\min}$ represent those of the energy fitness function. The adjusted functions are:

$$T' = \frac{T_{\max} - T}{T_{\max} - T_{\min}}$$

$$E' = \frac{E_{\max} - E}{E_{\max} - E_{\min}}$$

Thus, the bird swarm algorithm' s fitness function combines the adjusted time and energy functions:

$$fit = \alpha \times T' + \beta \times E'$$

where α and β are weights satisfying $\alpha + \beta = 1$, representing the optimization objectives of the bird swarm algorithm.

2.3 Scheduling Process

The scheduling process follows these steps:

a) Initialize the bird swarm algorithm and related parameters, mapping mobile cloud computing task scheduling schemes to bird individuals, and set the maximum iteration count.

b) Optimize the foraging behavior of the bird swarm algorithm.

c) Optimize the flight behavior of the bird swarm algorithm.

d) Compare the current task objective function value with the previous iteration' s value. If improved, replace the original bird individual' s position; otherwise, maintain it.

e) If the iteration count is less than the maximum, return to step c); otherwise, proceed to step f).

f) Output the position of the bird individual with optimal fitness, representing the best cloud computing task scheduling scheme.

3.1 Experimental Environment and Parameters

Experiments were conducted on hardware with a 2.4 GHz CPU, 8 GB DDR3 RAM, and 1 TB hard drive, running Windows 7 with MATLAB 2012b simulation environment and Java Hotspot 64-bit Server VM. Primary experimental parameters are shown in , while comparative algorithm parameters are listed in .

TABLE:1 Task allocation related parameters
- CPU processing capability (MIPS): 3000-10000 (interval 1000)
- Network transmission capability (bit/s): 100-1000 (interval 100)
- CPU usage rate: 0.1-0.9 (interval 0.1)
- Task length (million instructions): 10000-20000 (interval 1000)
- Input/output data size: 100-1000 (interval 100)

TABLE:2 Comparative algorithm parameters
- **Ant Colony Algorithm (ACO):** Pheromone value 0.005, evaporation coefficient 0.01, path selection probability 0.5
- **Particle Swarm Algorithm (PSO):** Inertia weight 0.5, two learning factors 0.5, random weight 0.5
- **Whale Algorithm (WOA):** Decreasing coefficient a linearly decreasing in [2,0]
- **Bird Swarm Algorithm (BSA):** Sensing and social coefficients both 1.5, a_1 and a_2 as 1, FL as 3
- **Improved Bird Swarm Algorithm ABSA [22] (IBSA):** Sensing and social coefficients both 1.5, a_1 and a_2 as 1, $FL \in [0.5, 0.9]$, $FQ = 6$, $C_{\min} = 0.8$, $P = 0.3$, $\lambda = 0.1$, $\tau = 0.1$
- **Proposed IBSA:** Minimum sensing coefficient 1, a_1 and a_2 as 1, $FL \in [0.8, 1]$, $FQ = 0.5$, initial scale coefficient 1

3.2 Experimental Content

Experiments consist of three parts: (1) evaluation of task scheduling functions, (2) testing task completion time and mobile device energy consumption under different task quantities using the proposed algorithm, and (3) comparing the proposed algorithm with others under varying task quantities. Mobile device quantities range from 50 to 500, with task quantities set at 100, 200, 500, and 1000.

a) Task Scheduling Function Evaluation

[Figure 1: see original paper] displays the comparison of task scheduling evaluation function values among six algorithms in mobile cloud computing. The proposed IBSA algorithm rapidly converges to the smallest stable value compared to the other five algorithms, demonstrating better convergence performance than the standard BSA algorithm and validating its effectiveness for mobile cloud computing task scheduling.

b) Task Completion Time and Device Energy Consumption of the Proposed Algorithm

[Figure 2: see original paper] and [Figure 3: see original paper] show the proposed algorithm's performance in task completion time and energy consumption. [Figure 2: see original paper] reveals that for a fixed task quantity, completion time gradually decreases as the number of mobile devices increases. With larger task quantities, increasing device count significantly reduces completion time, while for smaller task quantities, the decreasing trend becomes less pronounced—indicating that additional devices provide limited benefit when tasks are few. [Figure 3: see original paper] demonstrates that for a fixed device count, energy consumption increases with task quantity. However, when task quantity remains constant, energy consumption does not grow substantially with increasing device count, suggesting that the energy required to schedule a fixed number of tasks remains relatively constant.

c) Comparison with Other Algorithms in Time and Energy Consumption

[Figure 4: see original paper] and [Figure 5: see original paper] present completion time and energy consumption comparisons among six algorithms for 100 tasks. With small task quantities, differences between algorithms are minimal, indicating the proposed algorithm holds no significant advantage. [Figure 6: see original paper] and [Figure 7: see original paper] show results for 1000 tasks. As mobile device count increases, the proposed algorithm' s completion time remains substantially lower than the other five algorithms, demonstrating its time efficiency due to enhanced performance. Regarding energy consumption, the proposed algorithm achieves the lowest system device energy, followed by ABSA, WOA, PSO, with ACO consuming the most energy—confirming that the proposed algorithm maximizes energy savings.

4 Conclusion

Addressing the problems of long completion times and high mobile device energy consumption in mobile cloud computing task scheduling, this paper maps feasible scheduling schemes to bird swarm individuals and employs the scheduling objective function as the algorithm' s fitness function. By optimizing the sensing and social coefficients and incorporating learning factors in flight behavior, the algorithm' s performance is enhanced. Simulation experiments demonstrate that the proposed algorithm effectively reduces scheduling time and energy consumption in mobile cloud computing task scheduling.

Future research will investigate the impact of cost factors on task scheduling to optimize the objective scheduling model for even better performance in mobile cloud computing environments.

References

[1] Chen Shuang, Wang Yanzhi, Pedram M. A semi-markovian decision process based control method for offloading tasks from mobile devices to the cloud [C]// Proc of the 2013 Int Conf on Global Communications. Piscataway, NJ: IEEE, 2013: 2885-2890.

[2] Deng Shuiguang, Huang Longtao, Taheri J, et al. Computation offloading

for service workflow in mobile cloud computing [J]. IEEE Trans on Parallel and Distributed Systems, 2015, 26(2): 3317-3329.

[3] Khan A, Ahirwar K. Mobile cloud computing as a future of mobile multimedia database [J]. International Journal of Computer Science and Communication, 2011, 2(1): 219-221.

[4] Miettien A P, Nurminen J K. Energy of mobile clients in cloud computing [C]// Proc of the USENIX Int Conf on Hot Topcis in Cloud Computing. Berkeley, CA: USENIX Association, 2010: 14-21.

[5] Balamurugan M, Akila V. Effective processor selection on heterogeneous computing [C]// Proc of the 2016 Int Conf on Science Technology Engineering and Management. Piscataway, NJ: IEEE, 2016: 13-16.

[6] Feng Bailong, Gao Jing. Distributed parallel Needleman Wunsch algorithm on heterogeneous cluster system [C]// Proc of the 2015 Int Conf on Network and Information Systems for Computers. Piscataway, NJ: IEEE, 2015: 12-18.

[7] Ra M, Sheth A, Mummert L, et al. Odessa: Enabling interactive perception applications on Mobile Systems, Applications, and Services [C]// Proc of the 8th Int Conf on Mobile Systems, Applications, and Services. New York: ACM, 2011: 43-56.

[8] Terzopoulos G, Karatza H. Dynamic voltage scaling scheduling on power-aware clusters under power constraints [C]// Proc of the 17th Int Conf on Distributed Simulation and Real Time Applications. New York: ACM, 2013: 72-78.

[9] Saravanan S, Venkatachalam V. Advance MapReduce task scheduling algorithm using mobile cloud multimedia services architecture [C]// Proc of the 6th Int Conf on Advanced Computing. Piscataway, NJ: IEEE, 2014.

[10] Deshmukh N, Deorankar A. Minimizing energy consumption in transmission efficient wireless sensor network [C]// Proc of the Int Conf on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics. Piscataway, NJ: IEEE, 2016: 475-479.

[11] Song Zuyao, Dai Yueming. Joint optimization method for multi-workflow task scheduling in mobile cloud computing environment [J/OL]. Application Research of Computers: 1-7 [2020-02-20]. https://doi.org/10.19734/j.issn.1001-3695.2019.02.0058.

[12] Hu Haiyang, Liu Runhua and Hu Hua. Multi-Objective Optimization for Task Scheduling in Mobile Cloud Computing [J]. Journal of Computer Research and Development, 2017, 54(9): 1909-1919.

[13] Lin X, Wang Y, Xie Q, et al. Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment [J]. IEEE Transactions on Services Computing, 2014, 8(2): 175-186.

[14] Peng H, Wen W S, Tseng M L, et al. Joint optimization method for task scheduling time and energy consumption in mobile cloud computing environment [J]. Applied Soft Computing, 2019, 80: 534-545.

[15] Wang T, Wei X, Tang C, et al. Efficient multi-tasks scheduling algorithm in mobile cloud computing with time constraints [J]. Peer-to-Peer Networking and Applications, 2018, 11(4): 793-807.

[16] Tang C, Hao M, Wei X, et al. Energy-aware task scheduling in mobile cloud computing [J]. Distributed and Parallel Databases, 2018, 36(3).

[17] Wang Zhulin. Research on Task Allocation Strategy in Mobile Cloud Computing [D]. Changchun: Jilin University, 2016: 34-38.

[18] Shye A, Scholbrock B, Memik G. Into the wild: studying real user activity patterns to guide power optimizations for mobile architectures [C]. IEEE/ACM International Symposium on Microarchitecture. ACM, 2009.

[19] Kansal A, Zhao F, Liu J, et al. Virtual machine power metering and provisioning [C]. Proceedings of the ACM International Symposium on Computer Architecture, 2010: 326-337.

[20] Li Jianzhou. Research on the Android Power Management and Evaluation Mechanism [D]. Xi' an: Xidian University, 2014: 36-37.

[21] Meng X B, Gao X Z, Lu L, et al. A new bio-inspired optimisation algorithm: Bird Swarm Algorithm [J]. Journal of Experimental & Theoretical Artificial Intelligence, 2016, 28(4): 673-687.

[22] Fan Xiaohong, Wan Renxia. An Improved Bird Swarm Algorithm of Multi-process Searching Based on Aggregation [J]. Science Technology and Engineering, 2019, 19(16): 180-186.

[23] Li Yanyan, Wan Renxia. An Improved Algorithm for Bird Swarm Optimization [J]. Microelectronics & Computer, 2018, 35(09): 79-84.

[24] Wang Jianwei, Peng Yigong. Improved Bird Swarm Algorithm Based on Migration and Mutation Strategy and Its Application in Parameter Estimation [J]. Journal of East China University of Science and Technology, 2018, 44(04): 617-624.

Note: Figure translations are in progress. See original paper for figures.