Postprint: Research on Knowledge Search Algorithm for Cloud Computing Industry Alliance Based on Improved Particle Swarm Optimization

Authors: Gao Changyuan, Yu Jianping, He Xiaoyan

Date: 2017-11-08T00:00:00+00:00

Abstract

【Objective】To utilize improved particle swarm optimization for knowledge search in cloud computing industry alliances, thereby enhancing search accuracy and efficiency. 【Method】First, the Map function in MapReduce is employed to group particles for parallel processing, and subsequently the Reduce function is utilized to reduce particle search results, thereby shortening search time. During the particle search process, information exchange among particles within groups is conducted based on the average of optimal positions within each group, preventing premature convergence to a local optimum. 【Results】Through three sets of simulation experiments comparing the improved particle swarm optimization algorithm with the standard particle swarm optimization algorithm, the results demonstrate that the improved algorithm exhibits clear superiority in both efficiency and accuracy. 【Limitations】The sample data contains interfering data, which requires improvement. 【Conclusion】This method can improve the accuracy of knowledge search in cloud computing industry alliances and enhance search efficiency.

Full Text

Knowledge Search for Cloud Computing Industry Alliance: An Algorithm Based on Improved Particle Swarm Optimization

Gao Changyuan[1,2], Yu Jianping[1], He Xiaoyan[1,2]
[1](College of Management, Harbin University of Science and Technology, Harbin 150040, China)
[2](High-tech Industrial Development Research Center, Harbin University of Science and Technology, Harbin 150040, China)

Abstract

[**Objective**] This paper employs an improved particle swarm optimization algorithm for knowledge search in cloud computing industry alliances, aiming to enhance search accuracy and efficiency. [**Methods**] First, the Map function in MapReduce is utilized to group particles for parallel processing, followed by the Reduce function to aggregate search results, thereby shortening search time. During the particle search process, information interaction among particles within each group is guided by the average of optimal positions within the group, preventing premature convergence to a local optimum. [**Results**] Through three sets of simulation experiments comparing the improved particle swarm algorithm with the standard version, results demonstrate that the improved algorithm exhibits significant superiority in both efficiency and accuracy. [**Limitations**] The sample data contains interfering data that requires improvement. [**Conclusions**] This method can improve the accuracy of knowledge search in cloud computing industry alliances while enhancing search efficiency.

Keywords: Cloud Computing Industry Alliance; Knowledge Search; Particle Swarm Optimization Algorithm; MapReduce

Since its inception, cloud computing has developed rapidly, with countries worldwide paying close attention to its broad market prospects and significant industrial opportunities. According to the *White Paper on Cloud Computing Industry Development (2015 Edition)* released by the China Center for Information Industry Development, the global cloud computing service market reached $180 billion in 2015, representing an 18% growth rate, as IT companies globally transitioned toward cloud computing and the industry scale expanded rapidly. The establishment of cloud computing industry alliances aims to promote cloud computing ecosystem development and shared cloud databases, enabling knowledge search within the alliance through access to cloud databases via the alliance' s knowledge platform. Consequently, a critical challenge is how alliance members can quickly and accurately locate required knowledge within cloud databases.

To address this problem and effectively handle the massive volume and diversity of knowledge in cloud computing industry alliances, as well as the distributed and dynamically scalable nature of cloud databases [2], a search algorithm is needed to enable alliance members to acquire knowledge accurately and rapidly. Algorithm selection primarily follows two considerations: (1) **Search time constraints**. As cloud computing industry alliances operate and membership grows, cloud database volume and quantity continuously increase, creating difficulties for member searches, thus requiring methods that improve computational efficiency even at the cost of optimal precision [3]. (2) **Global optimality**. Cloud databases in the alliance form a massive resource pool, and conventional search algorithms such as ant colony algorithms tend to converge to local optimal solutions [4], preventing alliance members from finding more precise knowledge.

Particle swarm optimization is a heuristic algorithm capable of simultaneously solving static and dynamic combinatorial optimization problems [5]. The knowledge search process for alliance members aligns closely with the principle of birds foraging in particle swarms. However, cloud databases in the alliance are numerous and complexly distributed, and using only the standard particle swarm algorithm for knowledge search results in insufficient information communication and cooperation among particles, ultimately leading to entrapment in local optimal solutions and incomplete access to cloud databases. To prevent premature convergence in particle swarm algorithms, many improvement methods have been proposed, including adaptive parameter selection strategies such as dynamic nonlinear functions for parameter control [6], Q-learning for parameter adjustment [7], and adaptive critic strategies for parameter tuning [8]. Additionally, Wang Yanyan et al. proposed a dynamic grouping particle swarm optimization algorithm where the number of subpopulations is randomly selected from a specific set, which can improve overall solution quality but remains inadequate for complex problems [9]. Increasing numbers of scholars are focusing on particle swarm algorithms, which have been applied in cloud computing due to their excellent performance. Li Zhijie et al. used an improved particle swarm algorithm to handle grid resource allocation problems, achieving optimal grid resource allocation under storage constraints [10]. Li Yuanyuan et al. proposed a fast-converging improved particle swarm optimization algorithm for cloud platform scheduling, completing all tasks with high efficiency and accuracy [11]. These studies demonstrate that improved particle swarm algorithms offer advantages and receive consistent recognition in solving cloud computing problems, though most research focuses on modifying the algorithm' s parameters without considering the dispersed and diverse characteristics of knowledge within cloud computing industry alliances.

Based on the above analysis, this paper proposes a MapReduce-based improved particle swarm algorithm tailored to the characteristics of knowledge and cloud databases within cloud computing industry alliances. On one hand, addressing the dynamic incremental nature of knowledge in the alliance, MapReduce functions are combined to complete particle swarm grouping and achieve parallel processing, improving search efficiency and avoiding the one-sidedness problem caused by random initial position generation. On the other hand, the original particle swarm algorithm is improved by grouping particles and using the average of optimal positions within each group for information interaction, preventing algorithm convergence to a single local optimum and thereby enhancing search efficiency and accuracy.

2.1 Analysis of Cloud Computing Industry Alliance Members

Alliance members include software providers, network operators, hardware infrastructure providers, content providers, research institutions, and cloud computing service providers. Alliance members send requests to cloud computing service providers through networks and application software, which then return the required services. Software service providers supply software support to other members and provide software applications for the entire alliance' s operation, enabling members to access the cloud through software application connections. Network operators primarily provide network services, content services, and value-added services to other members, offering network support for the entire alliance's operation and enabling user access to the cloud via networks. Hardware infrastructure providers supply hardware facilities to cloud computing service providers, helping build cloud platforms and applications and providing the hardware foundation for alliance operations. Research institutions primarily provide theoretical and technical support, including alliance management, operations, risk control, and profit distribution design. Cloud computing service providers constitute the core of alliance operations, offering cloud resource services, platform services, and application services, relying on support from various members while providing cloud services to them.

Based on the operation of cloud computing industry alliances and members' inherent attributes, knowledge types in the alliance are primarily categorized as: operator knowledge, hardware equipment knowledge, software knowledge, content knowledge, cloud computing service knowledge, scientific research knowledge, and alliance innovation knowledge, as shown in Table 1 .

Table 1 Knowledge Types in Cloud Computing Industry Alliance

Knowledge Type	Description
Operator Knowledge	Knowledge generated during operations by China Mobile, China Unicom, and China Telecom
Hardware Equipment Knowledge	Knowledge generated during hardware manufacturing processes
Software Knowledge	Knowledge from software design, operation, and maintenance
Content Knowledge	Various media content including text, images, audio, and video
Cloud Computing Service Knowledge	Includes cloud platform knowledge, cloud resource knowledge, and cloud application knowledge
Scientific Research Knowledge	Theoretical and technical support provided by research institutes and universities
Alliance Innovation Knowledge	Knowledge generated during alliance operations

2.2 Analysis of Knowledge Characteristics in Cloud Computing Industry Alliance

(1) Dispersion: Cloud computing industry alliance members are widely distributed across different geographical locations, with each member' s knowledge stored in their respective cloud databases according to different organizational forms based on knowledge type. **(2) Diversity:** Knowledge in the alliance includes not only operator knowledge, hardware equipment knowledge, software knowledge, content knowledge, cloud computing service knowledge, and scientific research knowledge, but also innovative knowledge about the alliance, alliance maintenance knowledge, management skills knowledge, and alliance public relations knowledge. **(3) Massiveness:** Various types of knowledge in the alliance are continuously stored in different cloud databases, and massive amounts of interaction and exchange knowledge are generated through various interactions and collaborations among alliance members. **(4) Incrementality:** As cloud computing industry alliances develop and operate, knowledge in the alliance is generated and accumulated in large quantities through dynamic growth, and alliance members' own knowledge stock also continuously grows through knowledge interaction. **(5) Asset nature:** Knowledge can be regarded as intangible assets. Different enterprises submit their respective knowledge through the cloud computing industry alliance platform to jointly serve the alliance, continuously innovating and providing knowledge as a service or commodity to different users. Therefore, the value of knowledge lies in innovative knowledge products (services) developed through continuous exploration and processing.

The primary goal of establishing cloud computing industry alliances is to achieve knowledge sharing. However, the diverse, massive, and incremental characteristics of alliance knowledge determine the difficulty of knowledge sharing among alliance members. Knowledge search within the alliance becomes particularly important as members acquire knowledge to enhance their competitiveness and innovation capabilities, thereby gaining advantages in fierce market competition.

2.3 Analysis of Knowledge Search Optimization Process in Cloud Computing Industry Alliance

Members within cloud computing industry alliances possess the technical foundation to form a cloud environment. Member enterprises provide their own infrastructure or servers to form a cloud database for storing each member' s knowledge and external knowledge, and jointly build a cloud computing industry alliance knowledge platform. This platform enables search of cloud databases, allowing alliance members to quickly and conveniently acquire required knowledge. Knowledge that alliance members can provide to other members for free and knowledge requiring payment are expressed through ontologies and stored in cloud databases using cloud storage technology, with different knowledge types expressed through different ontologies. When conducting knowledge searches, alliance members select the corresponding ontology representation of knowledge types from the engine database as the knowledge carried by particles' initial positions based on their search requirements, and access cloud databases through API interfaces.

The large number of members in cloud computing industry alliances results in massive cloud databases storing member knowledge, and the dynamic joining and leaving of members makes the number of alliance cloud databases dynamically variable. Knowledge stored in alliance cloud databases exhibits diversity, massiveness, and incrementality. Since the standard particle swarm algorithm randomly generates initial search positions, it faces significant computational difficulty and long search times when conducting knowledge searches in the alliance using the original serial search method. Therefore, MapReduce functions, which are effective for processing massive knowledge, are introduced to accurately calculate particle initialization positions and help achieve parallel processing of knowledge search. Furthermore, the standard particle swarm algorithm requires all particles to update simultaneously during each iteration and demands extensive optimal position sharing and interaction to complete iterations, which becomes particularly difficult in the distributed environment of cloud computing industry alliances. Therefore, in the cloud computing industry alliance environment, we propose grouping particle swarms and conducting information interaction among particles within each group based on the average of optimal positions within the group. Based on the cloud computing industry alliance context, this paper innovatively proposes a knowledge search process using the improved particle swarm algorithm by comprehensively considering particle swarms and MapReduce functions: (1) According to the real-time number of cloud databases in the alliance, the Map function in MapReduce is used to group the entire particle swarm, transforming the original serial search into parallel search to reduce search time and improve efficiency. (2) After particles enter cloud databases, they continuously compare their carried knowledge with knowledge in the databases, selecting the most suitable knowledge according to alliance members' needs while using the average of optimal positions within the group as the basis for particle movement. (3) When particle search concludes, the Reduce function in MapReduce aggregates search results from different groups and returns one or more aggregated results to alliance members. The specific search process is shown in Figure 1 [Figure 1: see original paper].

3.1 MapReduce Function Construction

When using particle swarm algorithms for search, more particles yield higher algorithm efficiency and more significant effects. However, more particles also occupy more CPU resources. Leveraging the advantages of cloud computing MapReduce distributed computing and distributed storage provides sufficient memory space for particles while accelerating iteration counts. First, the Map function is applied to group the entire particle swarm and allocate it to different nodes for execution, achieving parallel processing. The Reduce function aggregates search results, where each node represents an independent database in the cloud database system. This solves the problem in traditional particle swarm algorithms where unidirectional local convergence occurs too quickly without reaching global optimum, effectively improving search time while better accommodating dynamically changing alliance knowledge. This paper first uses the Map function to divide a complete particle population into subpopulations based on the number of nodes and allocate them to different cloud databases for parallel processing. After particles complete their search, the Reduce function merges particles from different groups. The specific process is shown in Figure 2 [Figure 2: see original paper].

3.2 Improved Particle Swarm Algorithm Based on MapReduce

Particle Swarm Optimization (PSO), proposed by Dr. J. Kennedy and Dr. R.C. Eberhart in 1995, immediately became a research hotspot due to its algorithmic simplicity, minimal parameter tuning requirements, strong global optimization capability, and high efficiency [12]. The mathematical description of the standard particle swarm algorithm is as follows [13]: m particles search in a D-dimensional space, where the position of the i-th particle is represented by vector x_i, and its flight velocity is represented by v_i. The optimal position found by the i-th particle is denoted as $pbest_i$, and the optimal position found by the entire population is denoted as $gbest$. The velocity and position update for each particle are shown in equations (1) and (2). Here, c_1 and c_2 are two non-negative learning factors, r_1 and r_2 are random numbers between [0,1], k is the iteration number, and ω is the inertia weight [14].

The standard particle swarm algorithm requires all particles to update simultaneously during each iteration, which becomes particularly difficult in the distributed cloud computing environment. Moreover, during particle iteration, each movement requires all particles to share optimal positions, with optimal values potentially updating at any time, meaning particles need extensive optimal position sharing and interaction to iterate—unsuitable for cloud computing environments. Therefore, in cloud computing distributed environments, enabling asynchronous particle iteration and reducing data transmission of shared positions among particles is key to solving the problem. Although the standard particle swarm algorithm features strong robustness, particle distribution is relatively dispersed during the early search stage, requiring long search times despite extensive space coverage. After searching for some time, as particles' historical optimal positions and population optimal positions become nearly identical, particles move slowly, lose optimization capability, and the algorithm prematurely converges to local optimal values. Therefore, another improvement aspect is maintaining particle population diversity [15] to enhance search capability.

The improvements in this paper are as follows: After particle swarm grouping, each group conducts searches independently. For the j-th group, its velocity and position updates are shown in equations (3) and (4). Here, k represents the iteration number for the j-th group, $gbest_j$ represents the global extremum of the j-th group, and $pbest_{avg}$ represents the average of optimal positions within the group at time k; other parameters share the same meaning as in the standard particle swarm algorithm. These improvements aim to enable algorithm implementation in cloud computing distributed environments while leveraging cloud computing advantages to improve algorithm speed, promptly solving premature convergence problems and improving the ability to escape local optima.

3.3 Improved Particle Swarm Algorithm Steps

In the improved particle swarm algorithm, particles serve as knowledge carriers, achieving search through continuous comparison and selection with different types of knowledge in databases. When the algorithm terminates, the optimal particle obtained represents the knowledge required by cloud computing industry alliance members, completing the search.

The knowledge search process of the improved PSO algorithm (CPSO) is as follows: (1) According to alliance members' search conditions, determine the ontology-expressed knowledge categories related to members' input keywords from the engine database as the initial population. (2) Use the Map function in MapReduce to map and group the particle swarm based on the number of cloud databases. (3) Update particle velocities and positions using equations (3) and (4) to change each group' s particle velocities and positions. (4) Conduct knowledge selection: In each group, compare each particle' s fitness value with its group' s average optimal position $pbest_{avg}$; if the particle' s current fitness value is superior to $pbest_{avg}$, recalculate the group' s $pbest_{avg}$. Compare particle fitness values with the group' s optimal position $gbest_j$; if a particle' s fitness value is superior to $gbest_j$, reset the particle' s position to $gbest_j$. (5) When particles find results or the algorithm reaches maximum iteration times max, use the Reduce function to aggregate the final optimal solutions from different populations. (6) Return the optimal solution or solution set to alliance members.

To better validate the effectiveness of the improved particle swarm algorithm in cloud computing industry alliance knowledge search, this paper conducts three sets of simulation experiments. These experiments compare the improved particle swarm algorithm, its average throughput in cloud database searches, grouping delay, and search performance in cloud databases. The first experiment uses MATLAB 7.0 simulation software; the second selects the University of Southern California' s Information Sciences Institute (ISI) NS-2 as the simulation platform [16]; and the third builds cloud databases and platforms using Hadoop 2.0 on Linux systems. Experimental simulation data comes from the UCI KDD Archive developed by the University of California, a database platform for machine learning and knowledge discovery research. Based on the characteristics of cloud computing industry alliances and alliance knowledge, datasets matching alliance knowledge features are selected for simulation.

Three benchmark functions are used to test the standard particle swarm algorithm and the improved particle swarm algorithm: (1) Rastrigin function: $f(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$, $x_i \in [-5.12, 5.12]$; (2) Griewank function: $f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$, $x_i \in [-600, 600]$; (3) Rosenbrock function: $f(x) = \sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$, $x_i \in [-30, 30]$. Both Rastrigin [9] and Griewank [14] are multimodal functions that are difficult to optimize using general algorithms; Rosenbrock is a unimodal function with a gentle trend that makes optimal solution search challenging. All three functions have a dimension of 10 and an optimal solution of 0.

The function test iteration count is 1,000, with each function running independently 100 times. The simulation uses 10 computing units, each with 50 particles, equivalent to a population size of 500 particles. The standard PSO algorithm also uses 500 particles. Standard PSO algorithm parameters are set as $\omega = 0.9$, $c_1 = c_2 = 2$. For comparison convenience, the improved algorithm uses the same parameter values for c_1, c_2, and ω as the standard algorithm. Experimental results are shown in Table 2 .

Table 2 Comparison of Function Test Results

Algorithm	Rastrigin (Avg/Best/Worst)	Griewank (Avg/Best/Worst)	Rosenbrock (Avg/Best/Worst)
Standard PSO	18.34 / 12.45 / 25.67	0.81 / 0.45 / 1.23	156.78 / 98.34 / 234.56
Improved PSO	5.20 / 2.15 / 8.45	0.23 / 0.08 / 0.45	45.67 / 23.45 / 78.90

The experimental results show that with the same number of particles, the improved particle swarm algorithm significantly enhances performance for the multimodal Rastrigin function, with the average, best, and worst values all improving (average from 18.34 to 5.20), demonstrating clear improvement in global search capability. The multimodal Griewank function also shows improvement in average, best, and worst values, though due to the function' s concentrated value range, the average only improves from 0.81 to 0.23. The Rosenbrock function is a complex non-convex, ill-conditioned unimodal function where global search capability is crucial. The function provides limited information, making it difficult for traditional particle swarm algorithms to identify search direction quickly, often falling into local optima and resulting in large variations in single-run outcomes. The improved particle swarm algorithm' s global traversal and uniform distribution enable it to find the correct search direction quickly, enhancing its ability to escape local optima.

4.2 Cloud Database Throughput and Grouping Delay Simulation Analysis

The appropriateness of experimental network data depends on many interacting factors in nonlinear and unpredictable ways, making meaningful test environment selection very difficult. The approach follows methods defining finite sets composed of various classifications of adjustable components [17]. The University of Southern California' s Information Sciences Institute NS-2 is selected as the experimental platform. Throughput and data packet delay are used as effectiveness measures, with experimental data representing averages from 50 experiments, each simulation lasting 1,000 virtual seconds.

Experimental results in Figure 3 [Figure 3: see original paper] show that the improved particle swarm algorithm performs better, maintaining correct transmission at 1000Kb/s, while the standard particle swarm algorithm performs significantly lower. Results in Figure 4 [Figure 4: see original paper] demonstrate that the improved particle swarm algorithm has lower time delay.

4.3 Search Time and Accuracy Analysis

This study focuses on knowledge search in cloud computing industry alliances. To verify search accuracy and timeliness, an experimental environment for cloud computing industry alliance knowledge search was built using virtual machines. Under the virtual machine environment, 20 Linux environments were established, with each Linux system using Hadoop 2.0 to build cloud databases representing knowledge storage for alliance members and cloud platforms for knowledge search.

In the first simulation experiment, both the standard particle swarm algorithm (PSO) and the improved particle swarm algorithm (CPSO) were applied sequentially in the built cloud computing industry alliance environment for knowledge search. Figure 5 [Figure 5: see original paper] shows that the improved particle swarm algorithm achieves higher accuracy with less time consumption. This is because, in the simulated environment with numerous cloud databases, the standard particle swarm algorithm's randomly generated initial search positions and serial search approach consume more time. In later search stages, as particles' historical optimal positions and population optimal positions converge, particle movement becomes too slow, essentially losing optimization capability and resulting in lower accuracy. In the second simulation experiment, the dynamic grouping particle swarm algorithm (DGPSO) proposed in literature [9] and the improved particle swarm algorithm (CPSO) were applied sequentially in the same environment. Figure 6 [Figure 6: see original paper] shows that the improved particle swarm algorithm still achieves higher accuracy with less time consumption. Comparing Figures 5 and 6 reveals that the dynamic grouping particle swarm algorithm achieves higher accuracy than the standard version with improved performance and lower time consumption, but its search capability remains weaker than the improved algorithm proposed in this paper under the complex cloud computing industry alliance environment.

Conclusion

To address the challenge of knowledge search in cloud databases caused by the characteristics of knowledge in cloud computing industry alliances, this paper proposes an improved particle swarm algorithm for knowledge search in alliance cloud databases. By integrating MapReduce parallel computing concepts with the particle swarm algorithm, search time for the alliance's massive, rich, and dynamic knowledge is reduced. Simultaneously, particle grouping improves knowledge search accuracy. Three simulation experiments comparing the improved and standard particle swarm algorithms demonstrate that the improved algorithm exhibits clear superiority in efficiency and accuracy. Future research will focus on screening and matching knowledge search results to ensure effective knowledge acquisition for cloud computing industry alliance members.

References

[1] Zhai Lili, Liu Yufeng, Wang Jing, et al. Research on Evolutionary Game on Trust Among Software Industrial Virtual Cluster' s Enterprises [J]. *Chinese Journal of Management Science*, 2014, 22(12): 118-125.

[2] Shi Hengliang, Ren Chongguang, Bai Guangyi, et al. Cloud Database Dynamic Route Query Based on Self-adaptive Ant Colony Optimization [J]. *Computer Engineering and Applications*, 2010, 46(9): 10-12.

[3] Hiremath N C, Sahu S, Tiwari M K. Multi Objective Outbound Logistics Network Design for a Manufacturing Supply Chain[J]. *Journal of Intelligent Manufacturing*, 2013, 24(6): 1071-1084.

[4] Su Yi, Li Baizhou, Liu Xiaojing. A Model Fitting Analysis of the R&D Expenditure Based on Ant Colony Algorithm[J]. *Journal of Intelligence*, 2012, 31(5): 198-201.

[5] Wang Hongfeng, Wang Na, Wang Dingwei, et al. Improved Species-based Particle Swarm Optimizer for Multi-modal Optimization Problems [J]. *Journal of Systems Engineering*, 2012, 27(6): 854-864.

[6] Chatterjee A, Siarry P. Nonlinear Inertia Weight Variation for Dynamic Adaptation in Particle Swarm Optimization[J]. *Computers & Operations Research*, 2006, 33(3): 859-871.

[7] Stefan J, Martin M. A Hierarchical Particle Swarm Optimizer and Its Adaptive Variant[J]. *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics*, 2005, 35(6): 1213-1223.

[8] Doctor S, Venayagamoorthy G K. Improving the Performance of Particle Swarm Optimization Using Adaptive Critics Designs[J]. *Swarm Intelligence Symposium*, 2005, 9(9): 154-158.

[9] Wang Yanyan, Ge Hongwei, Wang Juanjuan, et al. A Particle Swarm Optimization Algorithm of Dynamic Grouping [J]. *Computer Engineering*, 2015, 41(1): 180-185, 189.

[10] Li Zhijie, Liu Xiangdong, Duan Xiaodong. Optimization of Grid Resource Allocation Using Improved Particle Swarm Optimization Algorithm [J]. *Computer Integrated Manufacturing Systems*, 2009, 15(12): 2375-2382.

[11] Li Yuanyuan, Qu Wenyu, Li Zhiyang, et al. Particle Swarm Optimization Algorithm with Fast Convergence Applied in Cloud Computing[J]. *Journal of Huazhong University of Science and Technology: Natural Science Edition*, 2012, 40(S1): 34-37.

[12] Shelokar P S, Siarry P, Jayaraman V K. Particle Swarm and Ant Colony Algorithms Hybridized for Improved Continuous Optimization[J]. *Applied Mathematics and Computation*, 2007, 188(1): 129-142.

[13] Fan Chengli, Xing Qinghua, Fu Qiang, et al. A Hybrid Intelligent Algorithm by Combining Particle Swarm Optimization with Variable Neighborhood Search for Solving Nonlinear Bilevel Programming Problems [J]. *Systems Engineering-Theory & Practice*, 2015, 35(2): 473-480.

[14] Luo Hong. Research on the Book Covers Search Engine Based on Improved PSO[J]. *Information Science*, 2014, 32(4): 106-108, 121.

[15] Li Guohui, Feng Mingyue, Yi Xianqing. Sensor Scheduling Method Based on Grouping Particle Swarm Optimization[J]. *Systems Engineering and Electronics*, 2010, 32(3): 598-602.

[16] Ye Xiaoguo. NS-2-Based Simulation Module Extension Method for Wireless Sensor Networks[J]. *Journal of Computer Research and Development*, 2011, 48(S2): 302-306.

[17] Liu Y, Chen J, Wang J. On Counting 3-D Matching of Size K[J]. *Algorithmic*, 2009, 54(10): 530-543.

Author Contributions

Gao Changyuan: Proposed research ideas, designed research plan, revised final manuscript version.
Yu Jianping: Designed and revised research plan, drafted and revised manuscript.
He Xiaoyan: Designed research plan, revised multiple manuscript versions.

Conflict of Interest Statement

All authors declare no conflict of interest.

Supporting Data

Supporting data is available in the journal's online version at http://www.infotech.ac.cn.
[1] Gao Changyuan, Yu Jianping, He Xiaoyan. text.xlsx. Test data.
[2] Gao Changyuan, Yu Jianping, He Xiaoyan. train.xlsx. Training data.

Received: June 27, 2016
Revised: November 27, 2016

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