

Research Article

A Novel Fused Optimization Algorithm of Genetic Algorithm and Ant Colony Optimization

FuTao Zhao,¹ Zhong Yao,¹ Jing Luan,¹ and Xin Song²

¹*School of Economics and Management, Beihang University, Beijing 100191, China*

²*School of Computer Science and Engineering, Beihang University, Beijing 100191, China*

Correspondence should be addressed to Zhong Yao; iszhyao@buaa.edu.cn

Received 31 May 2016; Accepted 31 July 2016

Academic Editor: Anna Vila

Copyright © 2016 FuTao Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A novel fused algorithm that delivers the benefits of both genetic algorithms (GAs) and ant colony optimization (ACO) is proposed to solve the supplier selection problem. The proposed method combines the evolutionary effect of GAs and the cooperative effect of ACO. A GA with a great global converging rate aims to produce an initial optimum for allocating initial pheromones of ACO. An ACO with great parallelism and effective feedback is then served to obtain the optimal solution. In this paper, the approach has been applied to the supplier selection problem. By conducting a numerical experiment, parameters of ACO are optimized using a traditional method and another hybrid algorithm of a GA and ACO, and the results of the supplier selection problem demonstrate the quality and efficiency improvement of the novel fused method with optimal parameters, verifying its feasibility and effectiveness. Adopting a fused algorithm of a GA and ACO to solve the supplier selection problem is an innovative solution that presents a clear methodological contribution to optimization algorithm research and can serve as a practical approach and management reference for various companies.

1. Introduction

Inspired by Darwin's evolution theory and Mendel's heredity theory, Holland first proposed genetic algorithms (GAs) in 1975 [1]. A GA is a biotic, general-purpose search optimization strategy designed to imitate the evolutionary processes of natural selection in biotic populations. The decision variables of the problem must be coded as chromosomes, and genetic operations of copying, crossover, and mutation are employed as the simulated gene pool changes over time in response to environmental pressures that enable optimal solutions to survive to the next generation. All GA optimization processes are based on these conceptions of chromosomes and biotic populations; GAs conform to the genetic and evolutionary principle of "survival of the fittest." GAs provide self-organization, self-adaption, and useful global search ability. As a global optimization method, GAs can handle all types of objective functions and constraints without the mathematical limitations that plague numerous approaches to optimization problems; therefore, GAs have been widely utilized in various applications. However, GAs do not have

a practicable feedback mechanism, so a large number of redundant iterations are produced when solutions are within a certain scope, resulting in low efficiency [2]. Additionally, GAs require long search times for Big Data problems [3].

Ant colony optimization (ACO) is a class of simulative evolutionary algorithms mimicking the foraging behavior of ants in nature, first proposed by Dorigo and Gambardella [4], and has successfully solved complicated optimization problems such as the traveling salesman problem (TSP), quadratic assignment problem, and job shop scheduling problem. Real-world ants apply stigmergy to their foraging process: when an ant forages, it marks the path that it has chosen by releasing pheromones as it walks. When an ant encounters a fork with no detectable pheromones, that ant will randomly choose one path, but when an ant encounters forking paths marked with pheromones, the ant's decision is not entirely random; the decision is influenced by the accumulation of pheromones on the paths. Regardless of which route the ant chooses, the pheromone that the ant releases will influence the decisions of other ants. The probability for an ant to choose a path depends on the number of ants that previously

chose that path. Therefore, in the absence of volatilization, a pheromone trail on a popular path will accumulate rapidly and help attract evergrowing numbers of ants to follow that path (positive feedback) [5]. Through this natural stigmergic process, without any prior knowledge, real-world ant colonies establish optimal foraging paths through exchanges of information between individuals and mutual cooperation. As a swarm intelligence optimization algorithm, ACO offers the advantages of parallel computation, self-learning, and effective information feedback. However, during the initial stages of ACO searches, little or no information is available; therefore, ACO searches often converge slowly.

The concept of integrating a GA and ACO was first proposed by Abbattista et al. [6] for exploiting the cooperative effect of ACO with the evolutionary effect of a GA. They integrated these search methods by using GAs to evolve optimal parameter values for ACO. Following their initial publication, numerous efforts have hybridized GAs and ACO, and now developed fusing approaches can be roughly divided into four categories. Approaches in the first category, such as the work of Acan et al. [7, 8], apply a GA to select ACO parameter values that optimize the performance of ant populations. Methods in the second category generate initial pheromone distributions through GAs, which are subsequently optimized with ACO [9–13]. Methods in the third category add genetic operations to ACO to diversify solutions [14–16]. Approaches in the fourth category combine the initialization from the second category and the diversification from the third category [17]. Approaches of the second category fuse the GA and ACO to take advantage of the GA's rapid convergence and ACO's parallelism and effective feedback; this fusion inspires the present study. The present study improves the search for an appropriate fusing time, thus enhancing the performance of the GA and ACO. Through a numerical experiment of the supplier selection problem, we demonstrated the feasibility and efficiency of this novel fused algorithm.

The supplier selection problem is an essential topic in supply chain management because the selection of proper supply partners can substantially improve a firm's competitive advantages and can further influence the quality and prices of the final products offered to customers [5]. In the context of previous research, the supplier selection problem can be described as a multigoal combinatorial optimization problem with the objectives of achieving production targets and maximum profits by selecting appropriate suppliers for each material under the condition of limited resources. Because the supplier selection problem is a typical combinatorial optimization problem, we consider that solving it with our fused GA and ACO algorithm could be noteworthy and shed new light on contemporary problems encountered in research on supplier selection.

The remainder of this paper is organized as follows: Section 2 reviews the literature regarding the development of the GA, ACO, and the fused GA and ACO algorithms. Section 3 describes the key points of our fused algorithm and elaborates on the modifications of fusing time, the GA, and ACO. Section 4 explains the design of a numerical experiment on the supplier selection problem and the optimization

of parameters; the performance of the new fused algorithm is evaluated. In the final section, we offer the conclusions and directions for future research.

2. Literature Review

2.1. Genetic Algorithms and Ant Colony Optimization. GAs and ACO are popular classes of intelligent heuristic algorithms, which have been broadly applied in the field of optimization. The initial versions were immediately improved upon, and improvements have continued to advance the performance of GAs and ACO. For GAs, substantial ameliorative work has been performed to optimize search performance, with features such as improved selection mechanism strategy, adaptive mutation probability [18], GA operators [19], and elitism selection mechanisms. To improve the global search capability and convergence performance of GAs, Wang et al. [20] proposed four types of improved GAs, namely, hierarchic GAs, simulated annealing GAs, simulated annealing hierarchic GAs, and adaptable GAs; these methods can overcome the defects of traditional GAs by combining GAs with simulated annealing algorithms and modifying various coding methods. In terms of ACO, the main features of its improvement involve mechanisms to intensify the search involving high-quality solutions and preserve a sufficient search space [21]. Niu et al. [22] stated that as a typical greedy heuristic algorithm, ACO tends to become trapped in local optima. They proposed a method for guiding the search away from local optima by adding a perturbation into the original probability. Moreover, a coefficient representing the effects of an average pheromone trail updated pheromones to reduce the effects of the parameter Q .

GA and ACO are promising because they can substantially increase the possibility of determining high-quality solutions for some complex combinatorial optimization problems, such as the supplier selection problem. To manage an integrated multi-item supplier selection problem for maximizing the annual income of an entire supply chain, Aliabadi et al. [23] presented a two-level GA (2LGA) model based on two types of variables—binary variables and real variables—of which the first layer was used for selecting suppliers and the following layer was used for ordering them. Simić et al. [24] proposed a GA performance value constraint model that used a grading variable for assessing the performance of suppliers. Yang et al. [25] applied a GA to a stochastic-demand multiproduct supplier selection model with constraints of service level and budget, where the highest value of the average expected profit and the lowest value of the standard deviation were achieved through different combinations of crossover and mutation rates. In reference to the attribute-based ant colony system (AACS), Tsai et al. [5] reported an examination of the critical factors; the criteria factors and weights were incorporated in the pheromone update rule and the AACS was used to obtain the optimal supplier according to a quantitative decision policy.

2.2. Fused Algorithm of a GA and ACO. A review of the extant algorithms introduced to solve combinatorial optimization

problems shows that intelligence optimization algorithms are gradually prevailing. Such algorithms include GAs and ACO, which are inspired by the behavior or processes present in nature. Each of these has its own advantages and disadvantages; thus, numerous researchers have considered investigations of multiple methods to be notable and hold promise for overcoming the defects of individual algorithms as well as achieving complementary advantages. The hybridization of a GA and ACO has been applied to solve numerous complex combinatorial optimization problems, such as the capacitated vehicle routing problem [26], logistics distribution route optimization [9], the 0-1 knapsack problem and quality of service [10], optimization of cloud database route scheduling [11], the virtual enterprise partner selection problem [12, 13], and some NP-complete problems, including the satisfaction problem, the tripartite matching problem, and the TSP [27].

In the relevant literature, the key to hybridizing GAs and ACO is to combine the population diversity and global searching ability of GAs with the feedback mechanism and rapid convergence of ACO to maximize accuracy and efficiency. In Zhang and Wu [17], the fused algorithm has two procedures; first, it approximates the global maximum by using a GA, and it then searches for the optimal solution by using ACO with GA operators. Two fusion ideas were proposed in Xiao and Tan [14]: in some cases, a GA is used to search for rough initial pheromone solutions, which initialize ACO information, and ACO subsequently seeks an optimal solution; however, a GA can be used to add crossover operators into ACO to prevent stagnation at local optima, thereby enhancing the global searching ability of ACO. In Liu [28], a GA was used to optimize the coefficients of pheromones, heuristics, and pheromone volatilization in ACO; thus, GAs and ACO were integrated to improve the efficiency of ACO. With a different approach to fusing a GA and ACO, Li et al. [15] added a heuristic factor of genetic information into an initial fixed heredity proportion to determine the transition probability of ACO; this was intended to minimize computational effort and increase the convergence rate during the path search.

3. Concept of Fusing a Genetic Algorithm and Ant Colony Optimization

3.1. The Concept of Fusing a GA and ACO. In this paper, the basic concept of the dynamic integration of a GA and ACO comes from Yao et al. [12, 13] and Xiong et al. [29]. We adopted a GA to generate available solutions and update initial pheromone values. An ACO implementation searches until the optimum is reached. Xiong et al. [29] presented a speed-time curve of a GA and ACO (Figure 1), where t_a is the optimal fusing time. In order to achieve a fusion time approximately equal to t_a , they proposed a dynamic integration strategy that set a minimum iteration Ge_{\min} (t_b moment), a maximum iteration Ge_{\max} (t_c moment), and a constant Ge_{die} for their GA. If the evolutionary loop returned a result that was less than a constant for Ge_{die} generations, the hybrid algorithm would terminate the GA loop and initiate the ACO search.

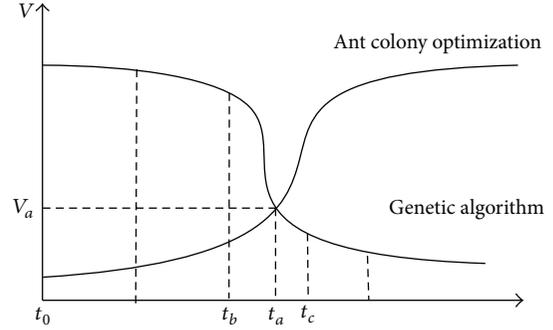


FIGURE 1: Difference of speed-time curve of GA and ACO.

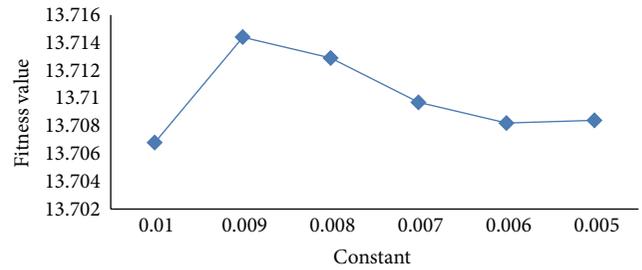


FIGURE 2: Variation of optimal fitness value among different constants.

3.2. Improvements of the Fused Algorithm. In this paper, we improved traditional GAs and ACO to enhance the performance of a hybrid algorithm.

3.2.1. Fusing Time of a GA and ACO. Based on the concept of setting the fusing time of the integrated algorithm reported in Xiong et al. [29], in this paper, we define the evolutionary rate as the variation rate of optimal fitness values between two successive iterations. When the evolutionary rate is detected to be less than a certain constant for three iterations of the loop ($Ge_{\text{die}} = 3$), the efficiency of the GA is considered to be low enough to end the GA loop and to engage ACO. To determine the constant, we compared the optimal fitness values among different constants ranging from 0.005 to 0.01 according to a value distribution of the evolutionary rate. Figure 2 shows the average fitness values of 10 iterations under different constants, where 0.009 is clearly the optimal constant.

3.2.2. Genetic Algorithm with Self-Adaptive Crossover and Mutation Probability. For a general GA, the crossover probability and mutation probability are constants. Although the algorithm may initially show a high convergence rate, if it lacks an explicit feedback mechanism, its efficiency gradually degenerates. In the context of Ma [30], self-adaptive crossover and mutation probabilities are introduced in our algorithm. By adjusting crossover and mutation probabilities automatically, the enhanced GA successfully avoids redundant iterations and low search efficiency in its later stages. The

self-adaptive crossover and mutation probability functions are as follows:

$$P_c = \begin{cases} P_{c_0}, & f \leq \bar{f}, \\ P_{c_1} \left(\frac{P_{c_0}}{P_{c_1}} \right)^{\frac{(f_{\max}-f)}{(f_{\max}-\bar{f})}}, & f > \bar{f}, \end{cases} \quad (1)$$

$$P_m = \begin{cases} P_{m_0}, & f' \leq \bar{f}, \\ P_{m_1} \left(\frac{P_{m_0}}{P_{m_1}} \right)^{\frac{(f_{\max}-f')}{(f_{\max}-\bar{f})}}, & f' > \bar{f}, \end{cases}$$

where P_{c_0} and P_{m_0} represent the higher crossover and mutation probabilities, P_{c_1} ($P_{c_1} < P_{c_0}$) and P_{m_1} ($P_{m_1} < P_{m_0}$) are the lower probabilities, f and f' are the lower fitness values of individuals, and f_{\max} , \bar{f} are the optimal and average fitness values in the population.

3.2.3. Updating Mechanism of the Pheromone in Ant Colony Optimization. Pheromone updating is a critical process of ACO. Stützle and Hoos [31] presented Max-Min ACO, which updates only the pheromones of the optimal solution after each iteration. This concept simplifies the pheromone updating method compared with traditional ACO, which updates the pheromone levels of all solutions. The pheromone constant Q affects the performance of ACO. In general, Q has an artificial initial value and cannot be changed as the search proceeds, and thus a general ACO implementation is at risk of stagnating at local optima. Therefore, a self-adaptive Q is introduced in this paper, where Q is not a constant, but varies according to a step function. Based on this, the functions of pheromone updating are as follows:

$$\tau_{S_{ij}}(t+n) = (1-\rho)\tau_{S_{ij}}(t) + \rho\Delta\tau_{S_{ij}}(t), \quad (2)$$

$$\Delta\tau_{S_{ij}}(t) = \sum_{k=1}^m \Delta\tau_{S_{ij}}^k(t), \quad (3)$$

$$\Delta\tau_{S_{ij}}^k(t) = \begin{cases} \frac{Q}{f_{\max}}, & S_{ij} \in f_{\max}, \\ 0, & \text{else,} \end{cases} \quad (4)$$

$$Q = Q_0 * \left(1 - \frac{w * N_t}{\max N} \right), \quad (5)$$

where $\rho \in [0, 1]$ is pheromone volatilization coefficient, $\Delta\tau_{S_{ij}}(t)$ is pheromone variation of the optimal solution, and $\Delta\tau_{S_{ij}}^k(t)$ is the pheromone left by each ant on the traversed nodes of the optimal solution. Equation (5) is the step function for Q , where Q_0 is the initial value of Q , $w \in [0, 1]$ is the adjustment coefficient, N_t is the current iteration, and $\max N$ is the maximum number of iterations.

3.3. Algorithm Processes

3.3.1. Genetic Algorithm

- (1) Initialize the control parameters of the GA, including population size N , high crossover and mutation probabilities P_{c_0} and P_{m_0} , lower crossover and mutation probabilities P_{c_1} and P_{m_1} , the end condition of the GA, namely, Ge_{\min} , Ge_{\max} , and Ge_{die} , and evolutionary rate Er .
- (2) Randomly generate initial population $G(0)$ in accordance with constraints, and set the index of generations g as $g = 0$.
- (3) Calculate the individual fitness value in $G(g)$ and the maximal and average fitness value in $MaxFit$ and $AvgFit$.
- (4) According to the individual fitness value and roulette choice strategy, set $P(i)$ as the choice probability of each individual in $G(g)$.
- (5) For ($N_t = 0$; $N_t < N$; $N_t = N_t + 2$)
 - (a) according to $P(i)$, select two individuals of $G(g)$ as fathers;
 - (b) calculate crossover probability P_c and mutation probability P_m ;
 - (c) generate random number $r = \text{random}[0, 1]$;
 - (d) if ($r \leq P_m$), implement a mutation operation on the two fathers; if the fitness value of the new individual is higher than that of its father, insert it into the next generation group $G(g+1)$;
 - (e) if ($P_m < r \leq P_m + P_c$), implement a crossover operation; if the fitness value of the member of the new generation is higher than its father, insert it into next generational group $G(g+1)$;
 - (f) otherwise insert the two fathers into the next generation group $G(g+1)$.

- (6) Calculate and update the individual fitness value, $MaxFit$, $AvgFit$, and $g = g + 1$.
- (7) Judge whether Er has been invariant for Ge_{die} generations or $g > Ge_{\max}$; if either test is true, the algorithm enters the ant colony optimization steps; if neither test is true, proceed to Step (4).

3.3.2. Ant Colony Optimization

- (1) Set the initial pheromones for the routes of ACO according to the results of the GA.
- (2) Set the $iter = 1$ ($iter$ is the index of search iterations), randomness coefficient w , optimal value of objective function $f = 0$, initial number of ants m , and path length n ; all the ants start from the beginning.
- (3) Initiate the feasible sets $allow_k$ (the allowable nodes for ant k) and solution sets tab_k (the nodes chosen by ant k for J types of materials).

- (4) According to the transition probability P , ant k moves to the next node and adds the selected node into tab , updating the feasible set $allow_k$.
- (5) After n iterations, all ants have traversed n nodes, and one round of the search process is complete. Calculate the fitness value f_k for all the solutions, marking the maximum of f_k as f_{max} and the corresponding solution as tab_{best} .
- (6) Update the pheromone in the optimal path, and set $iter = iter + 1$.
- (7) Judge whether $f < f_{max}$ and $iter \leq iter_{max}$; if so, set $f = f_{max}$ and return all the ants back to their starting nodes; then proceed to Step (3); if not, test whether $iter > iter_{max}$; if so, the search ends, resulting in the optimal known solution f and tab_{best} ; if not, return all the ants back to their starting nodes and proceed to Step (3).

4. Numerical Experiments

4.1. Instance Description. Supplier selection is a multigoal combinatorial optimization problem; it is an appropriate problem for swarm intelligence optimization algorithms, such as our proposed fused algorithm of a GA and ACO. In our numerical experiment, we defined N types of raw materials and components to be purchased; we defined J qualified suppliers. All suppliers are grouped into N categories according to the raw materials or components they can provide, and the task is to choose one supplier for each raw material. To ensure product quality, each raw material and component part can be offered by exactly one supplier, and each supplier can only offer a limited number of material types. Quality (Q), cost (C), delivery capability and flexibility (T), and innovation and development capability (D) are categorized and considered as the evaluation indices for the selection of suppliers. The objective of selecting suppliers is to maximize quality, delivery capability and flexibility, innovation, and development capability and to minimize cost, denoted as $\max\{Q, T, D, -C\}$. With the increase of J , the supplier selection problem clearly becomes a combinatorially explosive problem. Because effective selection of suppliers to meet all requirements is difficult, the problem must be transformed into a single-objective optimization problem. Here, we adopt the technique for order of preference by similarity to ideal solution (TOPSIS), a very effective method in multiobjective decision analysis. Its core concept is to compare distances between each evaluation option and positive/negative ideal solutions and to evaluate the available options. In terms of TOPSIS, for the i th material, the synthetic goal of its j th supplier $f_{ij}(t)$ can be written as (6), where $d^+(t)$ and $d^-(t)$ are the distances between each index value and positive/negative ideal values, (C^+, Q^+, T^+, D^+) and (C^-, Q^-, T^-, D^-) are the positive and negative ideal values of the four indices, respectively, q_{ij} , t_{ij} , d_{ij} , and c_{ij} represent the four index values of candidate j for i th material, and w_q^i, w_d^i, w_c^i , and w_t^i denote the weights of indices Q, D, C , and T for i th material. Thus, the objective function for this

supplier selection problem can be described as (9), where M_i are the numbers of potential suppliers for i th material. Based on TOPSIS, we converted a multiobjective combinatorial optimization problem to a single-objective form.

$$f_{ij} = \frac{d_{ij}^-}{d_{ij}^+ + d_{ij}^-}, \tag{6}$$

$$d_{ij}^+ = w_c^i \frac{|c_{ij} - C^+|}{C^+ + C^-} + w_q^i \frac{|q_{ij} - Q^+|}{Q^+ + Q^-} + w_t^i \frac{|t_{ij} - T^+|}{T^+ + T^-} + w_d^i \frac{|d_{ij} - D^+|}{D^+ + D^-}, \tag{7}$$

$$d_{ij}^- = w_c^i \frac{|c_{ij} - C^-|}{C^+ + C^-} + w_q^i \frac{|q_{ij} - Q^-|}{Q^+ + Q^-} + w_t^i \frac{|t_{ij} - T^-|}{T^+ + T^-} + w_d^i \frac{|d_{ij} - D^-|}{D^+ + D^-}, \tag{8}$$

$$\max f = \sum_{i=1}^N f_{ij}(t), \quad j = 1, \dots, M_i. \tag{9}$$

To examine the time and optimization performance of the hybrid algorithm, we coded a simulation case based on the supplier selection problem described previously. In our case, a middle-scale automobile enterprise was required to purchase 15 types of accessories in a market with 15 qualified suppliers for each accessory. To ensure the efficiency of suppliers, we supposed that each material could be supplied by only one supplier and that each supplier could offer only one material. The fitness value of potential suppliers by TOPSIS and partial data of simulation case are shown in *Appendix Table 1* and *Appendix Table 2*, respectively, in Supplementary Material available online at <http://dx.doi.org/10.1155/2016/2167413>. This numerical experiment comprised two parts. Parameter optimization was conducted to improve the efficiency of the novel fused algorithm. Given those optimal parameters, the GA, ACO, and our fused algorithm were applied separately to solve this supplier selection problem.

4.2. Parameter Optimization. Because of the lack of criteria for setting parameters in ACO, the main objective of parameter optimization is to adjust ACO parameters to approximate or reach optimal values. These parameters include ant number ant_Num , pheromone coefficient a , heuristic coefficient b , and pheromone volatilization coefficient r . Generally, ACO parameters are optimized by trials of their feasible values and empirical selection of values that approximate the optimal solution, as shown in Figure 3.

The number of ants can greatly affect the search efficiency. Figure 4 shows the performance levels of our hybrid algorithm with ant populations of 5 and 10. The maximal fitness values are plotted against the number of iterations in the third panel; we can conclude that the optimizing capacity of 10 ants is superior to that of 5 ants. Generally, within practical limits, when numbers of ants increase, the convergence speed increases; however, the improvement cannot be extended

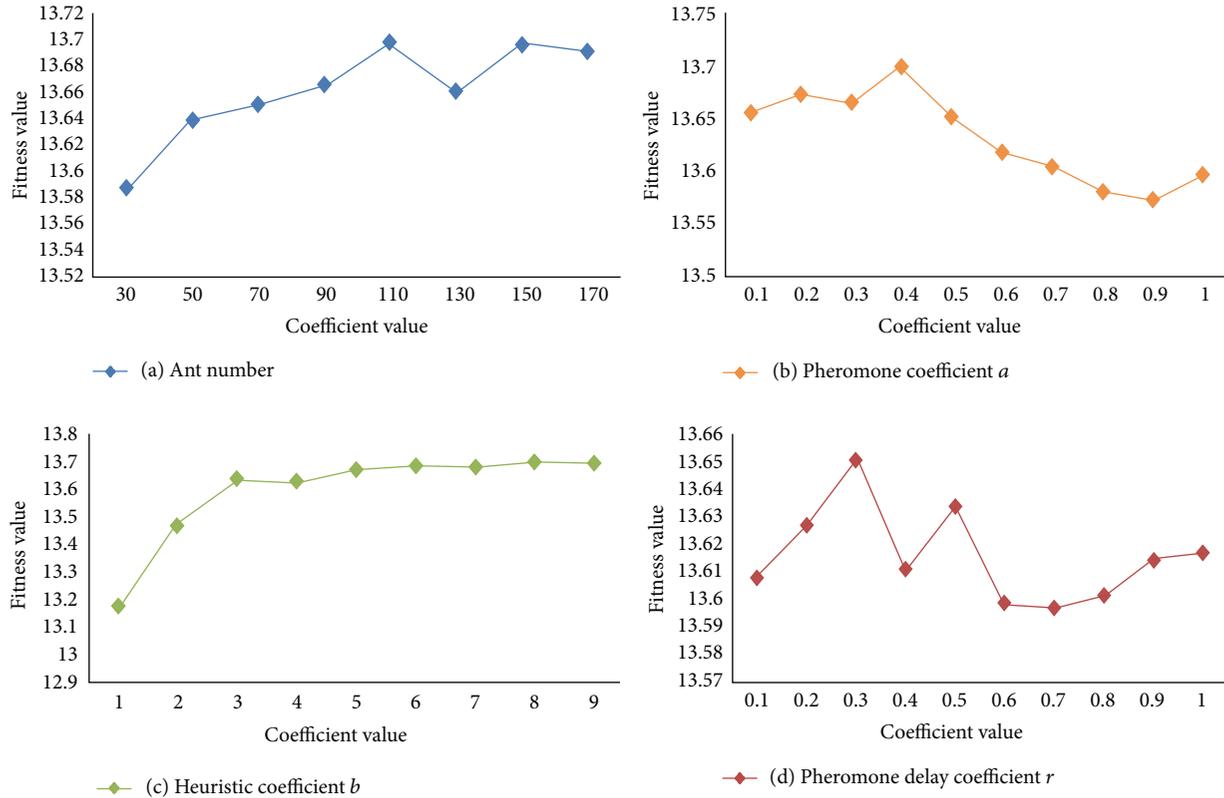


FIGURE 3: The influence of parameters.

indefinitely. Figure 3(a) shows that 110 is a pivotal point. The blue line shows the average optimal values for 10 iterations with ant populations ranging from 30 to 170. Given 110 ants, the iterative optimal value and performance of novel algorithm are optimal.

Regarding the pheromone coefficient a , which can cause the search to stagnate at local optima, the larger its value is, the more influence it exerts on transition probability P . The orange line in Figure 3(b) shows the influence of this pheromone coefficient on the optimal fitness value, and $a = 0.4$ performed better. The heuristic coefficient b reflects the effect of the heuristic on algorithm efficiency. As the green line in Figure 3(c) indicates, $b = 8$ is the most suitable value for our algorithm. The pheromone volatilization coefficient r determines the degree of pheromone volatilization. Specifically, the greater r is, the more the pheromones are left and the more easily the algorithm can stagnate. If r is excessively low, the pheromones volatilize too rapidly and the traces of an optimal path disappear before the ants can reinforce that path. In Figure 3(d), the red line demonstrates that $r = 0.3$ is the proper value.

Traditional parametric optimization involves setting all the other variables constant and only adjusting one parameter, but this traditional method requires excessive time and computational workload. By adopting another feasible fusion of a GA and ACO, Liu [28] used a GA to search for the optimal ACO parameter combination; the GA was applied to generate a parameter combination and parameter performance was

evaluated by comparing ACO solutions premised on those parameters. In this study, we also attempted to utilize the fusion of a GA and ACO to optimize parameters. Specifically, parameters a , b , and r were coded as chromosomes in the GA. Seven-digit codes were used for each parameter; each chromosome had 21 digits in total. The parameter combination generated by the GA was converted to decimal numbers according to the parameter scope and was applied by ACO for solving the supplier selection problem. The specific coding and converting scheme is shown in Table 1, and the results are displayed in Figure 5. The optimal values of pheromone coefficient a , heuristic coefficient b , and pheromone volatilization coefficient r were 0.4, 8, and 0.3, respectively; these results were equivalent to those of the traditional method, but they were reached after 60 iterations. Moreover, we discovered that the optimal fitness value obtained from the fused algorithm was inferior to that of the integrated algorithm of the GA and ACO for supplier selection (13.238 versus 13.729). This may have been caused by the influence of parameter uncertainty; this disparity indicates the pivotal role that parameters play in ACO.

4.3. Simulation Results and Analysis. To solve the problem described previously, we conducted a GA, ACO, and our fused algorithm with the optimal parameters. We used JAVA6 to code the algorithm and simulated the numerical example on a Windows 7 Ultimate platform. Figures 6 and 7 show the

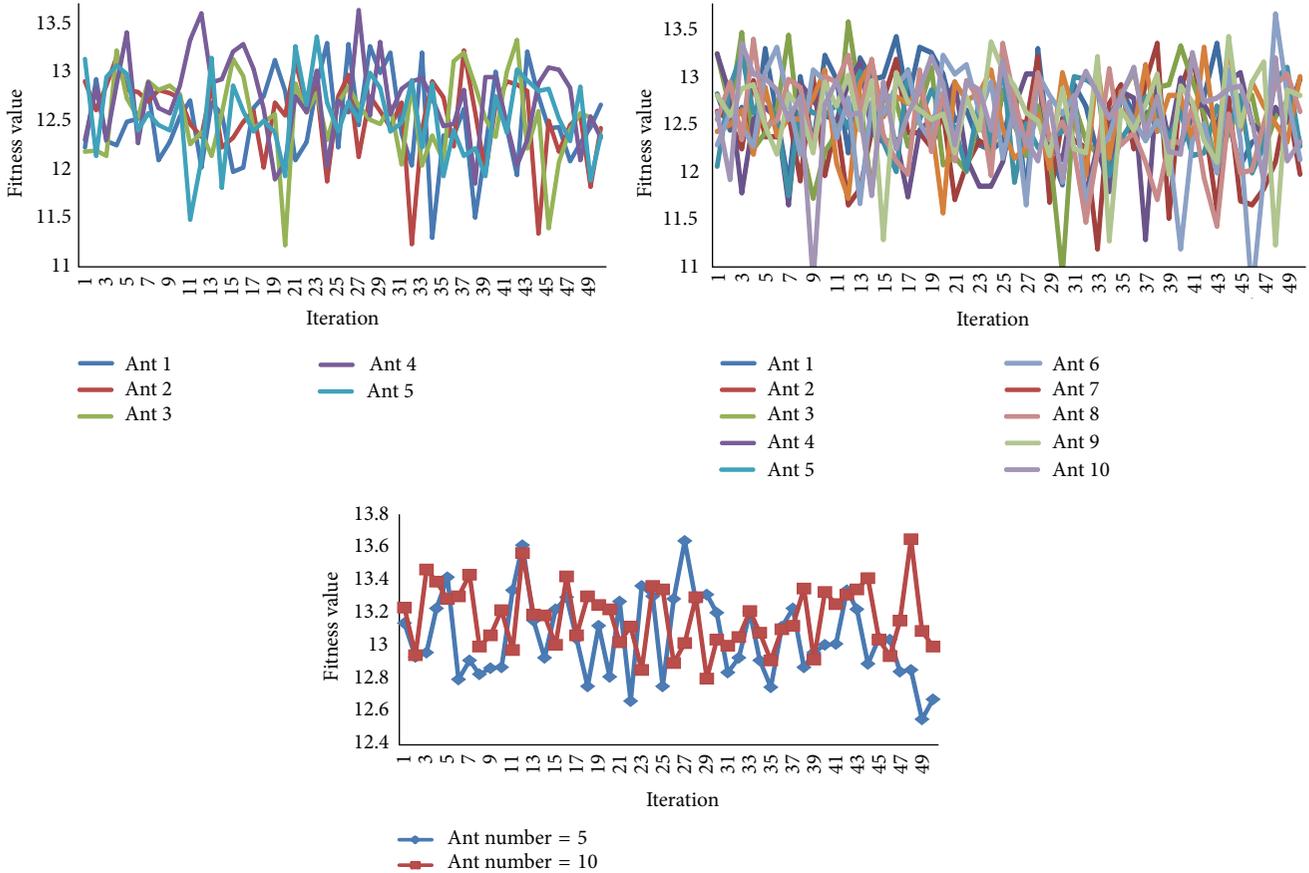


FIGURE 4: Performance of hybrid algorithm of GA and ACO with different numbers of ants.

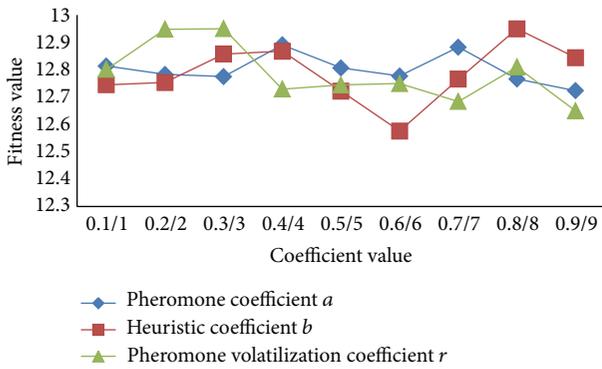


FIGURE 5: Parameter optimization by fused algorithm of GA and ACO.

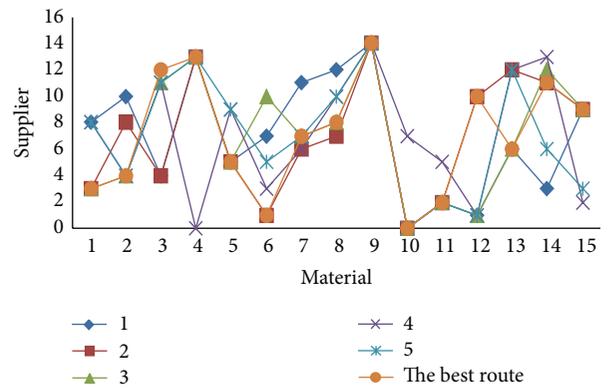


FIGURE 6: The result of supplier selection.

TABLE 1: Binary-coding and converting scheme of parameters.

Parameter	Binary-coding scheme							Decimalization
a	0	1	1	1	0	1	1	0.5
b	1	1	1	0	0	0	1	9
r	0	0	0	1	0	0	1	0.1

results, and the settings of the initial parameters are shown in Appendix Table 3.

Figure 6 displays the search results (six routes) of the ants of the hybrid algorithm, where the orange line shows the optimum. It verifies the feasibility of our new hybrid algorithm and the effectiveness of the optimal parameters and demonstrates that the hybrid algorithm can retain the superior solution and increase the diversity of solutions.

Figure 7 displays comparisons of GA-ACO, the GA, and ACO during 100 iterations, including the variations of fitness value and evolutionary rate. Figure 7(a1) plots the

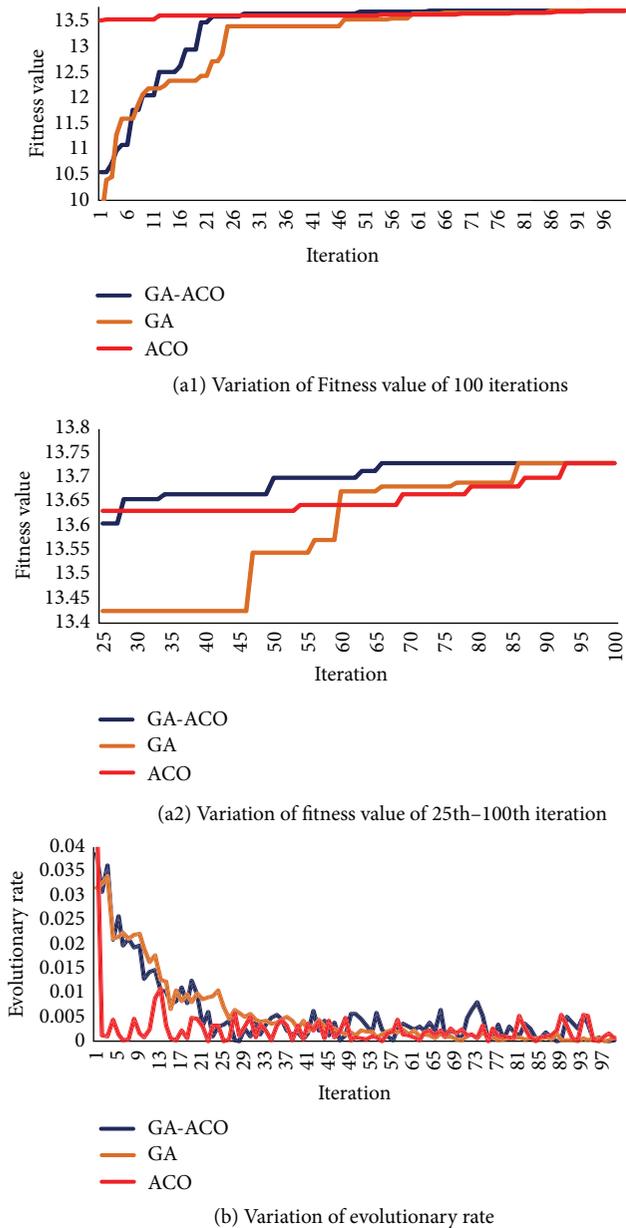


FIGURE 7: A comparison of operation process with GA-ACO, GA, and ACO.

fitness variations of GA-ACO, the GA, and ACO against 100 iterations. Figure 7(a2) shows the variations of fitness values for GA-ACO, the GA, and ACO from the 25th to the 100th iteration, which are easily observed. In detail, the GA (orange line) was stable at 13.729 after 86 iterations. ACO (red line) required 93 iterations to be stable at 13.729, whereas for the integrated algorithm (blue line), the function value reached the optimum, 13.729 after 66 iterations. Figure 7(b) shows that, at the early stage of searching, the fused algorithm has a higher convergence rate than ACO has, and at the later stage, the fused algorithm has a faster evolutionary rate than GA has. This demonstrates the main improvement and contribution of our novel fused algorithm compared with

traditional single algorithms and demonstrates its advantages of shorter time expenditure and higher efficiency.

Details of the comparison of these three algorithms are as follows.

4.3.1. Genetic Algorithm. As the orange curves in Figures 7(a1) and 7(a2) show, after 24 iterations, the variance of fitness values was dramatic, from 9.5679 to 12.8549. From the 25th iteration, the convergence rate gradually slowed and the fitness value changed from 13.424 to 13.545. From the 47th to the 65th iteration, the fitness value varied from 13.545 to 13.671. From the 66th to the 76th iteration, the evolutionary rate declined continuously, and after 11 iterations, the fitness value was 13.689. From the 77th to the 85th iteration, the searching process was smooth with a low changing ratio. At the 86th iteration, the algorithm reached its optimal value, 13.729. Until then, the searching algorithm had been stable. When considering the orange curve in Figure 7(b), although the evolutionary rate declined substantially, at the initial searching stage, the GA clearly had an excellent convergence rate and high efficiency. However, from the 25th iteration, the algorithm required excessive time to seek a better solution; that is, as the iterations increased, the convergence rate dropped even though it obtained the optimum, 13.729, after the 86th iteration. This verifies that at the later stage of the GA, its search efficiency was relatively low, and redundant iterations occurred frequently.

4.3.2. Ant Colony Optimization. Consider the red curve in Figure 7; at the early stages of the search, the overall change of the ACO fitness value was lower than that of the GA, up to the 39th iteration. However, at the later stages, from the 66th to the 92nd iteration, the solving process of ACO was relatively shorter than that of the GA, and the convergence rate was faster until the 96th iteration. At the 93rd iteration, ACO reached stability at the optimum, 13.729. This illustrates that ACO had the capacity to converge quickly to a local optimum. However, this also exposes a flaw of ACO, namely, that our search stagnated at a local optimum from the 12th to the 53rd iteration. From the red line, ACO clearly had a higher initial value than the GA and the new fused algorithm (ACO had 13.527, the GA had 9.5679, and our fused algorithm had 10.555). This is because the GA is a random algorithm, and its original populations are generated at random; however, in ACO, each transfer of ants is determined by probability. Therefore, ACO is N -level decision-making problem and ACO can likely obtain a better value than the GA can obtain. Additionally, because of the randomness, for the GA, the orange line fluctuates more frequently as does the fitness value, but for ACO, the red line is flatter and the fitness value changes only a few times.

4.3.3. Fusing Algorithm. The blue curve in Figure 7 shows the process of the integrated algorithm; the first 19 iterations used the GA, and ACO began from the 20th iteration. The optimal value varied quickly from 13.495 to 13.655 between the 20th and the 28th iteration. Most of the solving process was shorter than those of the GA and ACO. This clearly demonstrates the

merits of the GA, namely, a high convergence rate at early search stages, and also illustrates the advantages of ACO, namely, the ability to converge quickly to a local optimum. Although ACO is often limited by a low improvement rate in its early iterations because of the lack of pheromones, the proposed method overcame that obstacle. Moreover, the proposed method efficiently avoided the redundant late stage iterations that are typical of a GA.

5. Conclusions and Future Research

In this paper, we described a novel fused algorithm that employs a GA and ACO for the supplier selection problem. It provides the advantages of a GA and ACO and effectively avoids their defects. Each part of the fused algorithm is improved, and in the context of Xiong et al. [29], the rational integration of these two algorithms is carefully observed and designed. To test the feasibility and effectiveness of the new fused algorithm, three separate instances of a supplier selection problem were implemented for the GA, ACO, and our new fused algorithm. The results show that our new fused algorithm delivered a better time than the times of its competitors, and the new fused algorithm delivered the optimal known value as the solution of its objective function.

The present study has some limitations. The proposed ideas deserve to be improved and explored. For example, the scale of the simulation case applied in this paper is relatively small, and some large-scale studies should test our fused algorithm. Therefore, further research can focus on verifying our fused algorithm in terms of other typical combinatorial optimization problems, such as the TSP. Additionally, the universality of our new fused algorithm must be tested and numerous previously unresolved challenges can be further investigated with our new fused method. Furthermore, parameters and their influence on optimization performance should be studied in greater detail; identifying the optimal time to cease the GA and engage ACO would be warranted.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This work has been supported by the Natural Science Foundation of China (Projects nos. 71271012, 71671011, and 71332003).

References

- [1] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Oxford, UK, 1975.
- [2] P. F. Peng, "Improvement and simulation of ant colony algorithm based on genetic gene," *Computer Engineering & Applications*, vol. 46, no. 4, pp. 43–45, 2010.
- [3] Q. Zhu and S. Chen, "A new ant evolution algorithm to resolve TSP problem," in *Proceedings of the 6th International Conference on Machine Learning and Applications (ICMLA '07)*, pp. 62–66, Cincinnati, Ohio, USA, December 2007.
- [4] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *BioSystems*, vol. 43, no. 2, pp. 73–81, 1997.
- [5] Y. L. Tsai, Y. J. Yang, and C.-H. Lin, "A dynamic decision approach for supplier selection using ant colony system," *Expert Systems with Applications*, vol. 37, no. 12, pp. 8313–8321, 2010.
- [6] F. Abbattista, N. Abbattista, and L. Caponetti, "An evolutionary and cooperative agents model for optimization," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 668–671, IEEE, Perth, Australia, 1995.
- [7] A. Acan, "GAACO: A GA+ACO hybrid for faster and better search capability," in *Proceedings of the 3rd International Workshop on Ant Algorithms*, vol. 2483 of *Lecture Notes in Computer Science*, pp. 300–301, ANTS, Brussels, Belgium, 2002.
- [8] D. X. Gong and X. G. Ruan, "A hybrid approach of GA and ACO for TSP," in *Proceedings of the 5th World Congress on Intelligent Control and Automation*, pp. 2068–2072, IEEE, Hangzhou, China, July 2004.
- [9] S. Zhu, W. Dong, and W. Liu, "Logistics distribution route optimization based on genetic ant colony algorithm," *Journal of Chemical & Pharmaceutical Research*, vol. 6, no. 6, pp. 2264–2267, 2014.
- [10] W. G. Zhang and T. Y. Lu, "The research of genetic ant colony algorithm and its application," *Procedia Engineering*, vol. 37, no. 2012, pp. 101–106, 2012.
- [11] Y. H. Zhang, L. Feng, and Z. Yang, "Optimization of cloud database route scheduling based on combination of genetic algorithm and ant colony algorithm," *Procedia Engineering*, vol. 15, pp. 3341–3345, 2011.
- [12] Z. Yao, J. Liu, and Y.-G. Wang, "Fusing genetic algorithm and ant colony algorithm to optimize virtual enterprise partner selection problem," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 3614–3620, IEEE, Hong Kong, June 2008.
- [13] Z. Yao, R. Pan, and F. Lai, "Improvement of the fusing genetic algorithm and ant colony algorithm in virtual enterprise partner selection problem," in *Proceedings of the World Congress on Computer Science and Information Engineering (CSIE '09)*, pp. 242–246, Los Angeles, Calif, USA, April 2009.
- [14] H. F. Xiao and G. Z. Tan, "Study improvement of the fusing genetic algorithm and ant colony algorithm in virtual enterprise partner selection problem on fusing genetic algorithm into ant colony algorithm," *Journal of Chinese Computer System*, vol. 30, no. 3, pp. 512–517, 2009.
- [15] X. M. Li, Z. Mao, and E. Qi, "Research on multi-supplier performance measurement based on genetic ant colony algorithm," in *Proceedings of the 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation (GEC '09)*, pp. 867–870, 2009.
- [16] S. Gao, Z. Zhang, and C. Cao, "A novel ant colony genetic hybrid algorithm," *Journal of Software*, vol. 5, no. 11, pp. 1179–1186, 2010.
- [17] Y. D. Zhang and L. N. Wu, "A novel genetic ant colony algorithm," *Journal of Convergence Information Technology*, vol. 7, no. 1, pp. 268–274, 2012.
- [18] M. Bessedik, F. B.-S. Tayeb, H. Cheurfi, and A. Blizak, "An immunity-based hybrid genetic algorithms for permutation flowshop scheduling problems," *International Journal of Advanced Manufacturing Technology*, vol. 85, no. 9, pp. 2459–2469, 2016.

- [19] Z. H. Ahmed, "Experimental analysis of crossover and mutation operators on the quadratic assignment problem," *Annals of Operations Research*, 2015.
- [20] X. M. Wang, X. Liu, and G. Liu, "Performance comparison of several kinds of improved genetic algorithm," *Journal of Chemical and Pharmaceutical Research*, vol. 6, no. 9, pp. 463–468, 2014.
- [21] M. Lopez-Ibanez, T. Stutzle, and M. Dorigo, "Ant colony optimization: a component-wise overview," IRIDIA-Technical Report Series TR/IRIDIA/2015-006, 2015.
- [22] S. H. Niu, S. K. Ong, and A. Y. C. Nee, "An enhanced ant colony optimiser for multi-attribute partner selection in virtual enterprises," *International Journal of Production Research*, vol. 50, no. 8, pp. 2286–2303, 2012.
- [23] D. E. Aliabadi, A. Kaazemi, and B. Pourghannad, "A two-level GA to solve an integrated multi-item supplier selection model," *Applied Mathematics and Computation*, vol. 219, no. 14, pp. 7600–7615, 2013.
- [24] D. Simić, V. Svirčević, and S. Simić, "A hybrid evolutionary model for supplier assessment and selection in inbound logistics," *Journal of Applied Logic*, vol. 13, no. 2, pp. 138–147, 2015.
- [25] P. C. Yang, H. M. Wee, S. Pai, and Y. F. Tseng, "Solving a stochastic demand multi-product supplier selection model with service level and budget constraints using genetic algorithm," *Expert Systems with Applications*, vol. 38, no. 12, pp. 14773–14777, 2011.
- [26] A. Mazidi, M. Fakhrahmad, and M. Sadreddini, "A meta-heuristic approach to CVRP problem: local search optimization based on GA and ant colony," *Journal of Advance in Computer Research*, vol. 7, no. 1, pp. 1–22, 2016.
- [27] G. F. Dong, W. W. Guo, and K. Tickle, "Solving the traveling salesman problem using cooperative genetic ant systems," *Expert Systems with Applications*, vol. 39, no. 5, pp. 5006–5011, 2012.
- [28] M. J. Liu, *Research on integration and performance of ant colony algorithm and genetic algorithm [Ph.D. thesis]*, School of Science, China University of Geosciences, Beijing, China, 2013.
- [29] Z.-H. Xiong, S.-K. Li, and J.-H. Chen, "Hardware/software partitioning based on dynamic combination of genetic algorithm and ant algorithm," *Journal of Software*, vol. 16, no. 4, pp. 503–512, 2005.
- [30] Z. J. Ma, "Partner selection of supply chain alliance based on genetic algorithm," *Academic Journal of System Engineering Theory and Practice (Chinese Journal)*, vol. 9, pp. 81–84, 2003.
- [31] T. Stützle and H. H. Hoos, "MAX-MIN ant system," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 889–914, 2000.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

