Creation of supply chain management methods based on multi-agent systems and metaheuristics

Eugene Fedorov[1,*,†], Olga Nechyporenko[1,†], Oleg Grygor[1,†] and Maryna Leshchenko[1,†]

[1] *Cherkasy State Technological University, Shevchenko blvd., 460, Cherkasy, 18006, Ukraine*

Abstract

Currently, the interaction between its participants plays an important role in the supply chain management process. The article proposes a multi-agent method for selecting product suppliers, which automates supplier selection and speeds up the decision-making process of the seller. The developed method includes determining the characteristics of product suppliers, creating a protocol for interaction between the seller and suppliers, developing methods for forming requirements for suppliers, analyzing the characteristics of suppliers and evaluating suppliers based on the results of their work. The proposed metaheuristic method for assigning orders to product suppliers is based on a genetic algorithm and, through the use of dynamic parameters, improves the accuracy and speed of the method and ensures its convergence. Prospects for further research include studying the proposed methods for a wide class of artificial intelligence problems.

Keywords

supply chain management, multi-agent system, protocol for interaction between seller and suppliers, metaheuristics, genetic algorithm

1. Introduction

Currently, one of the areas of logistics is supply chain management. Within this direction, artificial intelligence methods are actively used for forecasting, classification, and clustering [1]. Also for the supply chain management process, the interaction between its participants plays an important role [2-3], which can be modeled using a multi-agent system [4-6].

The goal of the work is to increase the efficiency of selecting product suppliers and assigning orders to them based on multi-agent systems and metaheuristics.

To achieve the goal, the following tasks were set and solved:

1. Conduct an analysis of existing multi-agent systems and optimization methods aimed at optimizing the assignment of orders within the framework of supply chain management.

2. Create a multi-agent method for selecting product suppliers.
3. Create a multi-agent metaheuristic method based on a genetic algorithm to solve the assignment problem.
4. Perform a numerical study.

2. Literature review

Interactive sites of sellers (for example, online stores) are usually not intended for interactions with suppliers, and, in particular, for holding a competition among them. Therefore, multi-agent technologies are currently becoming widespread, allowing for the selection of suppliers.

The introduction of a multi-agent environment into the sales force structure is promising due to the following aspects:

1. Security of transactions on the Internet, which is achieved due to the fact that any information about the consumer and supplier is transmitted using a special agent message language, Agent Communication Language (ACL). This eliminates the possibility of using intercepted information flows.
2. The ability to provide up-to-date information about supplier agents in your directory, thanks to the use of a directory service (DF).
3. The possibility of creating new types of agents to work with consumers and suppliers.

Currently, the most preferred protocol for interaction between the seller agent and supplier agents is the Contract net protocol, which is implemented in multi-agent environments. However, the standard Contract Net is abstract in nature, is not focused on a specific subject area, and does not cover all stages of interaction. Thus, there is a need to modify this protocol.

To improve efficiency, the seller's assignment of orders to selected suppliers should rely on optimization techniques. Optimization methods that find an exact solution have high computational complexity. Optimization methods that find an approximate solution through a directed search have a high probability of hitting a local extremum. Random search methods do not guarantee convergence. In this regard, the problem of insufficient efficiency of optimization methods arises, which needs to be solved.

To quickly find a quasi-optimal solution to optimization problems and reduce the probability of hitting a local extremum, metaheuristics (or modern heuristics) are used [7-8]. Metaheuristics extend the capabilities of heuristics by combining heuristic methods based on a high-level strategy [8-10]. Metaheuristics are used for discrete and continuous optimization [11-12].

Existing metaheuristics have one or more of the following disadvantages:

- there is only an abstract description of the method or the description of the method is focused on solving only a specific problem [13];
- the influence of the iteration number on the process of finding a solution is not taken into account [14];
- the convergence of the method is not guaranteed [15];
- there is no possibility to use non-binary potential solutions [16];

- the procedure for determining parameter values is not automated [17];
- there is no possibility to solve conditional optimization problems [18-19];
- insufficient accuracy of the method [20-21].

Thereby, the problem arises of constructing effective metaheuristic optimization methods [22-23].

One of the popular metaheuristics is evolutionary metaheuristics [24-25]. Among them, we can highlight a genetic algorithm that allows solving combinatorial optimization problems (for example, the assignment problem) [26-27].

3. Multi-agent method for selecting product suppliers

To intellectualize the interactive selection of product suppliers to a seller (for example, an online store), it is necessary to:

- determine the characteristics of product suppliers;
- propose a protocol for interaction between the seller and suppliers;
- develop methods to formulate requirements for suppliers, analyse supplier characteristics and evaluate suppliers based on the results of their work.

3.1   Product supplier characteristics

This paper highlights the following characteristics of product suppliers:

x_1 – cost of purchased products,

x_2 – products' grade (varies from high-quality to low-quality products, it is possible to indicate a list of manufacturing companies),

x_3 – the distance between the seller and the supplier's products warehouse,

x_4 – conditions for the delivery of products (pickup by the seller, delivery to the seller's warehouse, direct delivery to consumers) and its guarantee (if the delivery is carried out by the supplier),

x_5 – lead times for orders,

x_6 – the minimum batch size of products purchased from the supplier and the period for which this batch should be purchased,

x_7 – terms of payment (no advance payment, work on credit, provision of installments, or lack thereof) and the need to make a security deposit,

x_8 – the availability and organization of the supplier's quality management system,

x_9 – the ability of the supplier to service the supplied equipment with spare parts throughout its entire service life,

x_{10} – the supplier's connections with respectable organizations,

x_{11} – the number of years of the supplier's existence in the market of products and services,

x_{12} – the availability of product certificates,

x_{13} – the availability of discounts and conditions for their granting,

x_{14} – the frequency of products' delivery to the supplier,

x_{15} – providing notifications about product updates,

x_{16} – providing notifications about changes in products' prices,

x_{17} – providing notice of termination of supply of certain products,

x_{18} – terms of return and exchange of defective products,

x_{19} – terms for the return or exchange of illiquid products,

x_{20} – the range of products (wide or narrow).

These characteristics are learned from product suppliers in the process of interactive interaction with them.

3.2   Interaction protocol between the seller and suppliers

Today, the interaction of the seller with suppliers is intellectualized through a multi-agent system (MAS). In MAS, the online store and suppliers are represented by their respective agents. The most effective protocol for interaction between them is the Contract Net. However, the standard Contract Net is abstract in nature, is not focused on a specific subject area, and does not provide for all stages of interaction.

Therefore, in order to select product suppliers based on an interactive competition, a modified Contract Net protocol is proposed, which includes the following stages:

- problem identification (the need for the supply of products);
- problem announcement;
- proposal;
- contract conclusion;
- problem solving;
- declaring further cooperation.

The first four stages are shown in Figure 1.

At the *problem identification stage*, the sales agent determines, based on the product sales forecast and inventory control, that he needs products to sell. The selling agent understands that he needs products of a certain quality, in certain terms, etc.

At the *problem announcement stage*, the selling agent informs the supplying agents about his problem and also makes demands on them. Three situations are possible:

1. If the selling agent does not know about certain characteristics of the supplying agents, then he is forced to send a message to all the supplying agents.
2. If the selling agent knows which supplying agents are possible candidates, then he can send a limited message to only those candidates.
3. If the selling agent knows exactly which supplying agent is the appropriate one, then he can send a message to the specific supplying agent.

At the *proposal stage*, the supplying agents, aware of the problem, evaluate it relative to their own characteristics, i.e. whether they can solve it quickly and whether they can satisfy the requirements of the selling agent. If the problem can be solved by the supplying agent,

then the supplying agent makes a proposal to the selling agent. The proposal specifies the characteristics of the provider agent that are associated with solving this problem. A selling agent may receive many such proposals in response to a single problem announcement.
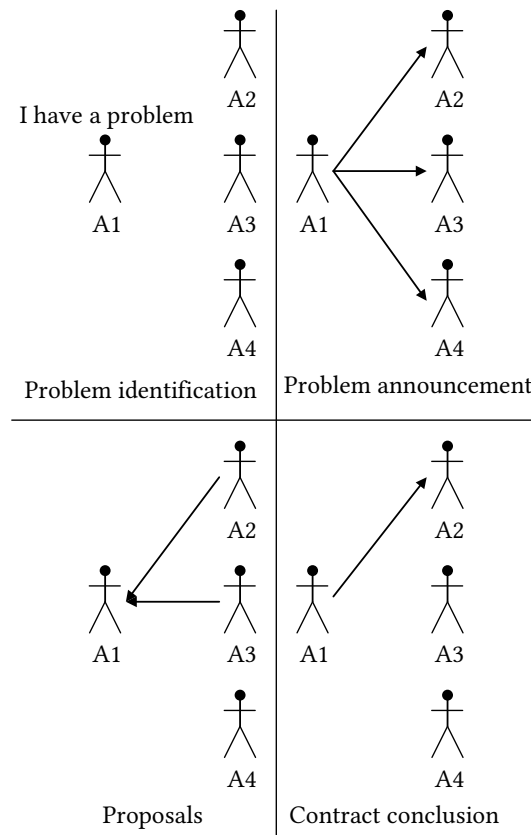

Figure 1: Contract Net Protocol.

At the *contract conclusion stage*, based on the characteristics of the supplying agents contained in their proposals, the selling agent selects the most suitable supplying agents to solve the problem. The selling agent announces his choice to the selected supplying agents through a message about the conclusion of the contract. These selected supplier agents take responsibility for solving the problem.

At the *problem solving stage*, supplier agents implement the solution to the problem. Once the problem is resolved, the supplier agents send messages to the selling agent.

At the *stage of declaring further cooperation*, the selling agent informs about the prospects for further cooperation with the supplying agents based on the results of their solution to the problem.

This negotiation process can be simplified to make the protocol more efficient. If the selling agent knows exactly which supplying agent is suitable for solving the problem, then he can enter into a direct contract with that supplying agent.

This contract differs from the announced contract in that there is no announcement and no proposals, but the conclusion of the contract is made directly. In such cases, supplier agents

with whom contracts have been entered into must acknowledge receipt of the contract notification and have the opportunity to refuse.

For tasks that make simple requests for information, the contract may not be used. In such cases, the request-response sequence can be used without further improvements. Such messages are implemented as request and inform messages. The request message is used to encode direct requests for information when entering into a contract is unnecessary. The inform message is used to respond to a request message and to transmit data.

For the Contract Net interaction protocol, there is its FIPA specification SC00029, which in this work was modified by adding a stage for declaring further cooperation and adding methods for the stages of offer, contract conclusion and declaring further cooperation. The modified FIPA-ContractNet interaction protocol (represented as a UML diagram in Figure 2) allows the selling agent (Initiator) to declare its problem to the supplying agents (Particiapnt).

Supplier agents process the statement of problem (cfp) of the selling agent and decide whether to send their proposals (propose) or refuse to solve the problem (refuse) based on logical-formal rules.
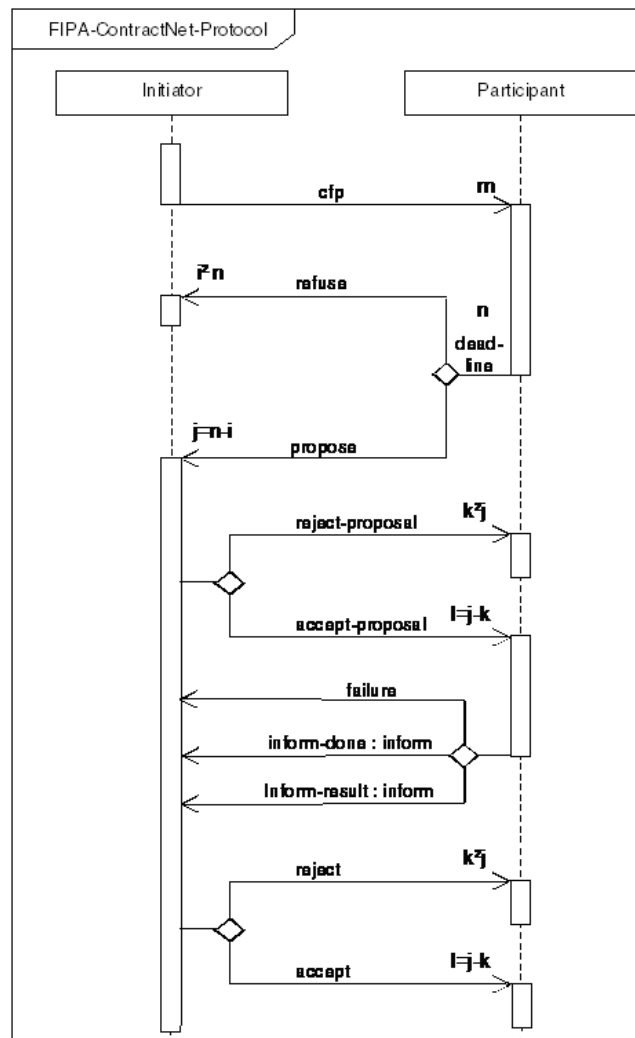

Figure 2: UML diagram of the modified FIPA-ContractNet interaction protocol.

The selling agent processes proposals and makes a decision to accept (accept-proposal) or reject (reject-proposal) proposals based on logical-formal rules. If the proposals are accepted, the relevant supplying agents notify the selling agent of the outcome of the problem resolution. Such a notification can be failure (if the solution to the problem is unsuccessful), inform-done (if the problem is successfully resolved and the sales agent is notified only about this) or inform-result (if the problem is successfully resolved and the sales agent is notified about this and the result).

The selling agent processes the result of solving the problem and makes a decision to accept or reject further cooperation based on logical-formal rules. In addition, during the execution of this protocol, the recipient can inform the sender that he did not understand the information sent to him. This is done through a non-understood message.

Appropriate methods must be developed for the stages of proposal, contract conclusion and declaring further cooperation.

3.3 Method of forming requirements for suppliers

In this work, for the proposal stage, a method has been created for generating requirements for suppliers based on the following logical-formal rules

$$x_1 \in A_1 \wedge x_2 \in A_2 \wedge x_3 \in A_3 \wedge x_4 \in A_4 \wedge x_5 \in A_5 \rightarrow propose,$$

$$x_1 \notin A_1 \vee x_2 \notin A_2 \vee x_3 \notin A_3 \vee x_4 \notin A_4 \vee x_5 \notin A_5 \rightarrow refuse,$$

where A_i – is the set of values of the ith characteristic of the supplier, declared by the seller at the stage of the problem announcement.

3.4 Method for analyzing supplier characteristics

In this work, for the contract conclusion stage, a method has been created for analyzing the characteristics of suppliers based on the following logical-formal rules

$$\sum_{i=6}^{20} P(B_i) < \varepsilon \rightarrow reject\_proposal,$$

$$\sum_{i=6}^{20} P(B_i) \geq \varepsilon \rightarrow accept\_proposal,$$

where B_i – is the set of values of the ith characteristic, declared by the supplier at the proposal stage,

$P(B_i)$ – predicate which means that B_i satisfies the selling agent,

ε – threshold value, is specified.

3.5 Method to evaluate suppliers based on their performance

In this work, for the stage of declaring further cooperation, a method has been created for evaluating suppliers based on the results of their work based on the following logical-formal rules

$$y_1 \in B_1 \wedge y_5 \in B_5 \rightarrow accept,$$

$$y_1 \notin B_1 \vee y_5 \notin B_5 \rightarrow reject,$$

where y_i – the real value of the ith characteristic obtained by the supplier at the problem solving stage.

4. Metaheuristic method for assigning orders to product suppliers with an unequal number of suppliers and orders

The fitness function (target function) in the case of a generalized assignment problem with equal weights (equal to 1) and equal budgets (equal to M) is defined as

$$F(x) = \sum_{i=1}^{L} \sum_{j=1}^{M} w_{ij}[x_j = i] \to \max_x, \ [x_j = i] = \begin{cases} 1, & x_j = i \\ 0, & x_j \neq i \end{cases},$$

$$\forall i \in \overline{1, L} \ \sum_{j=1}^{M} v_{ij}[x_j = i] \leq t_i, \ v_{ij} = 1, \ t_i = M, \tag{1}$$

$$\forall j \in \overline{1, M} \ \sum_{i=1}^{L} [x_j = i] = 1,$$

where w_{ij} – income from fulfillment of the the i th supplier of the j th order,

v_{ij} – weight of fulfillment of the the i th supplier of the j th order,

t_i – i th supplier's budget,

x_j – supplier for j th order,

M – number of orders,

L – number of suppliers.

The genetic algorithm with dynamic parameters and natural genes consists of the following steps:

1. Initialization.

1.1. Setting the maximum number of iterations N, population size K, chromosome length (number of orders) M, number of possible chromosome gene values (number of suppliers) L, number of crossover iterations N^c, number of mutation iterations N^m, minimum and maximum values for probability $p^{\min}, p^{\max}$.

1.2. Creation of the initial population P.

1.2.1. Chromosome number $k = 1$, $P = \varnothing$.

1.2.2. Randomly creating a chromosome

$$x_k = (x_{k1}, ..., x_{kM}), \ x_k = round(1 + (L-1)U(0,1)), \tag{2}$$

where $U(0,1)$ – function returning a standard uniformly distributed random number,

$round()$ – function that rounds a number to the nearest integer.

1.2.3. If $x_k \notin P$, then $P = P \bigcup \{x_k\}$, $k = k+1$.

1.2.4. If $k \leq K$, then go to step 1.2.2.

1.3. Determination of the global best chromosome

$$k^* = \arg \max_{k \in \overline{1,K}} F(x_k), \ x^* = x_{k^*}. \tag{3}$$

1.4. The initial probability is set

$$p(1) = p^{\max}. \tag{4}$$

2. Iteration number $n = 1$.

3. Reproduction to create an intermediate population $\tilde{P}$.

3.1. Arrange P by fitness function, i.e. $F(x_k) > F(x_{k+1})$.

3.2. The probability of choosing each kth chromosome is defined as

$$p(x_k) = \frac{1}{K} p(n) + \frac{F(x_k)}{\sum\limits_{s=1}^{K} F(x_s)} (1 - p(n)), \ i \in \overline{1, K}. \tag{5}$$

3.3. Reproduction iteration number $k = 1$.

3.4. If $\exists m \in \overline{1, K-1}: \sum\limits_{s=1}^{m} p(x_s) < U(0,1) < \sum\limits_{s=1}^{m+1} p(x_s)$, then $\tilde{x}_k = x_m$.

3.5. If $k < K$, then $k = k+1$, go to step 3.2.

4. Crossing over natural genes to create an intermediate population $\breve{P}$.

4.1. Crossing over iteration number $m = 1$.

4.2. The first parent is randomly selected

$$k = round(1 + (K-1)U(0,1)). \tag{6}$$

4.3. The second parent is selected based on proximity to the first parent and depending on the iteration number.

$$\text{If } U(0,1) < p(n), \text{ then } l = \arg \min_{s \in \{1, \ldots, K\} \setminus \{k\}} \left\{ \sum_{j=1}^{M} [x_{kj} = x_{sj}] \right\},$$

$$\text{otherwise } l = \arg \max_{s \in \{1, \ldots, K\} \setminus \{k\}} \left\{ \sum_{j=1}^{M} [x_{kj} = x_{sj}] \right\}. \tag{7}$$

4.4. Two descendants are created

$$\lambda_j = U(0,1), \ \breve{x}_{mj} = \begin{cases} x_{kj} & \lambda_j < 0.5 \\ x_{lj} & \lambda_j \geq 0.5 \end{cases}, \ \breve{x}_{2m,j} = \begin{cases} x_{lj} & \lambda_j < 0.5 \\ x_{kj} & \lambda_j \geq 0.5 \end{cases}, \ j \in \overline{1, M}. \tag{8}$$

4.5. If $m < N^c$, then $m = m+1$, go to step 4.2.

5. Mutation over natural genes to create an intermediate population $\widehat{P}$.

5.1. Mutation iteration number $s = 1$, descendant number $m = 0$.

5.2. If $U(0,1) < p(n)$, then go to step 5.5, otherwise $m = m+1$.

5.3. Chromosome is randomly selected

$$k = round(1 + (K-1)U(0,1)). \tag{9}$$

5.4. The chromosome gene is randomly selected

$$s = round(1 + (M-1)U(0,1)). \tag{10}$$

5.5. Chromosome gene mutation

$$\lambda_s = round(1 + (L-1)U(0,1)), \ \hat{x}_{mj} = \begin{cases} x_{kj} & j \neq s \\ \lambda_s & j = s \end{cases}, \ j \in \overline{1,M}.$$

(11)

5.6. If $s < N^m$, then $s = s+1$, go to step 5.2.

6. Combining the current and intermediate populations into one population

$$H = P \bigcup \breve{P} \bigcup \hat{P}.$$

(12)

7. Reduction of the combined population.

7.1. Arrange H by fitness function, i.e. $F(h_k) > F(h_{k+1})$.

7.2. Update current population

$$x_k = h_k, \ k \in \overline{1,K}.$$

(13)

8. Determine the best chromosome based on the target function

$$k^* = \arg\max_{k \in 1,K} F(x_k).$$

(14)

9. Determination of the global best chromosome. If $F(x_{k^*}) > F(x^*)$, then $x^* = x_{k^*}$.

10. Update probability

$$p(n+1) = p^{\min} + (p^{\max} - p^{\min})\left(1 - \frac{n}{N}\right).$$

(15)

10. If $n < N$, then $n = n+1$, go to step 3, otherwise stop.

The result is x^*.

5. Experiments and results

A numerical study of the proposed optimization method was carried out using the Python package in the Google Colab environment.

In the work, the population size is $K = 100$, the maximum number of iterations is $N = 100$, the number of crossing over iterations is $N^c = 100$, the number of mutation iterations is $N^m = 100$, the minimum and maximum probability values are $p^{\min} = 0.01, p^{\max} = 1$.

For the assignment problem, the search for a solution was carried out on the standard tai50a database (traditionally used for testing methods for solving the assignment problem).

The decreasing probability function is determined by the formula $p(n+1) = p^{\min} + (p^{\max} - p^{\min})\left(1 - \frac{n}{N}\right)$ and is presented in Figure 3.

The dependence (Figure 3) of the decrease in probability on the iteration number shows that the probability decreases with increase iteration number.

For the assignment problem, the search for a solution was carried out on the standard tai50a database. The results of comparing the proposed method with the traditional genetic algorithm method are presented in Table 1.
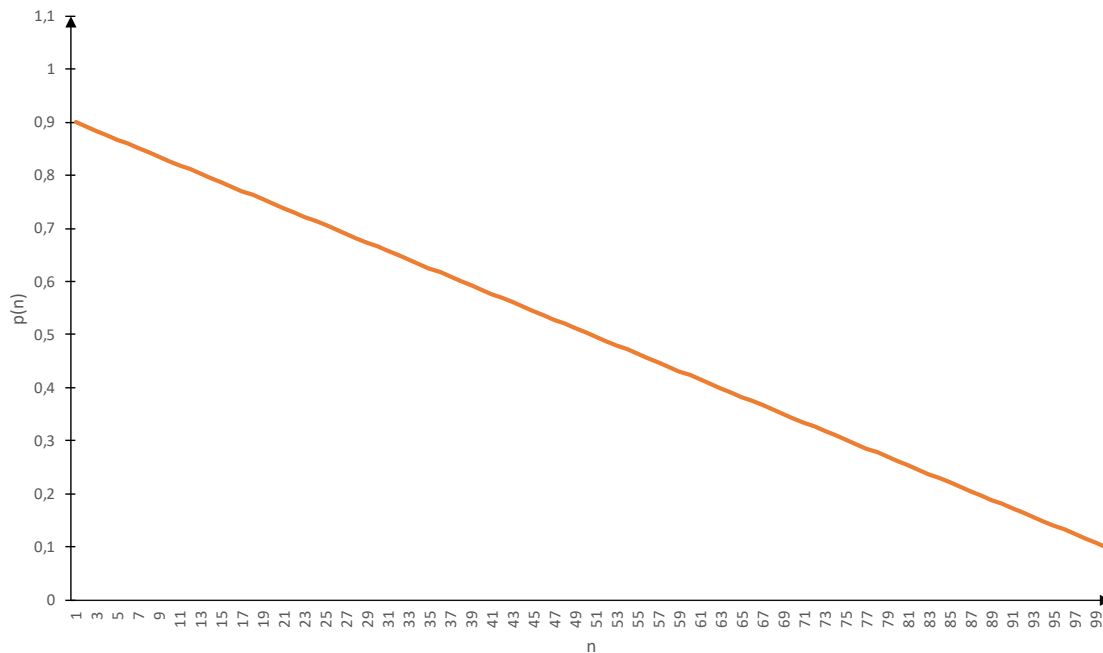
Figure 3: Decreasing probability function.

Table 1

Comparison of the proposed and traditional ant system method

Number of iterations		Mean squared error of the method	
proposed	existing	proposed	existing
100	1000	0.03	0.09

6. Discussion

Advantages of the proposed method:

1. The proposed reproduction operator can be used at both the initial and the final stages of the genetic algorithm, since at the early stages of the genetic algorithm, equiprobable selection is used, ensuring the exploration of the entire search space (random selection of chromosomes), and at the final stages proportional selection is used, making the search directed (the current best chromosomes are saved).

2. The proposed crossover operator can be used at both the initial and the final stages of the genetic algorithm, since outbreeding (crossing the most distant chromosomes from each other) is used at the early stages of the genetic algorithm, which ensures the exploration of the entire search space (random selection of chromosomes), and in the final stages, inbreeding is used (crossing the chromosomes that are closest to each other), making the search directed (the current best chromosomes are retained).

3. The proposed mutation operator can be used at both the initial and the final stages of the genetic algorithm, since at the early stages of the genetic algorithm the probability of mutation is high, which ensures exploration of the entire search space (random

selection of chromosomes), and at the final stages the probability of mutation is low, which makes the search directed (the current best chromosomes are retained).

4. The proposed method is superior to the classical method in terms of accuracy and speed (Table 1).

7. Conclusions

1. A multi-agent method for selecting product suppliers was developed, which automates the selection of suppliers and speeds up the decision-making process for the seller. The developed method includes determining the characteristics of product suppliers, creating a protocol for interaction between the seller and suppliers, developing methods for forming requirements for suppliers, analyzing the characteristics of suppliers and evaluating suppliers based on the results of their work.
2. The proposed metaheuristic method for assigning work to product suppliers through the use of dynamic parameters makes it possible to increase the accuracy and speed of this method and ensure its convergence.
3. Prospects for further research are the study of the proposed methods for a wide class of artificial intelligence problems.

References

[1] N. Boyko, R. Kovalchuk, Data Update Algorithms in the Machine Learning System, Computer Systems and Information Technologies, volume 1 (2023) 6-13. doi: 10.31891/csit-2023-1-1.
[2] G. Jezic, J. Chen-Burger, M. Kusek, R. Sperka, R. J. Howlett, L. C. Jain (Eds.), Agents and multi-agent systems: technologies and applications, volume 186 of Smart innovation, systems and technologies, 2020. doi.org/10.1007/978-981-15-5764-4.
[3] S. Russell, P. Norvig, Artificial Intelligence: a Modem Approach, Englewood Cliffs, NJ: Prentice Hall PTR, 2020.
[4] E. Fedorov, O. Nechyporenko, Modeling of Negotiations Between Supply Chain Participants Based on a Multi-Agent System, in: CEUR Workshop Proceedings, 2022, volume 3156, pp. 33-43.
[5] T. T. Mezgebe, H. B.-El Haouzi, G. Demesure, R. Pannequin, A. Thomas, Multi-agent systems negotiation to deal with dynamic scheduling in disturbed industrial context, Journal of Intelligent Manufacturing 31(2020) 1367-1382. doi: 10.1007/s10845-019-01515-7.
[6] J. Rocha, Multi-agent Systems, IntechOpen, London, 2017.
[7] X.-S. Yang, Nature-inspired Algorithms and Applied Optimization, Charm: Springer, 2018. doi: 10.1007/978-3-642-29694-9.
[8] A. Nakib, El-G. Talbi, Metaheuristics for Medicine and Biology, Berlin: Springer-Verlag, 2017. doi: 10.1007/978-3-662-54428-0.
[9] S. Subbotin, A. Oliinyk, V. Levashenko, E. Zaitseva, Diagnostic rule mining based on artificial immune system for a case of uneven distribution of classes in sample, Communications, volume 3 (2016) 3-11.
[10] X.-S. Yang, Optimization Techniques and Applications with Examples, Hoboken, New Jersey: Wiley & Sons, 2018. doi: 10.1002/9781119490616.

[11] T. Neskorodieva, E. Fedorov, Method for automatic analysis of compliance of settlements with suppliers and settlements with customers by neural network model of forecast, in: Advances in Intelligent Systems and Computing, volume 1265, 2021, pp. 156-165.

[12] A. Vasuki, Nature Inspired Optimization Algorithms, 1st ed., Boca Raton, FL: Chapman and Hall/CRC, New York, 2020. doi: 10.1201/9780429289071.

[13] H. Wang, M. Huang, J. Wang, An effective metaheuristic algorithm for flowshop scheduling with deteriorating jobs, Journal of Intelligent Manufacturing 30(2019) 2733–2742. doi: 10.1007/s10845-018-1425-8.

[14] C. Blum, G. R. Raidl, Hybrid Metaheuristics. Powerful Tools for Optimization, Charm: Springer, 2016. doi: 10.1007/978-3-319-30883-8.

[15] R. Martí, P. M. Pardalos, M. G. C. Resende, Handbook of Heuristics, Charm: Springer, 2018. doi: 10.1007/978-3-319-07124-4.

[16] B. Chopard, M. Tomassini, An Introduction to Metaheuristics for Optimization, Springer, New York, 2018. doi: 10.1007/978-3-319-93073-2.

[17] J. Radosavljević, Metaheuristic Optimization in Power Engineering, New York: Institution of Engineering and Technology, 2018. doi: 10.1049/PBPO131E.

[18] N. S. Jaddi, J. Alvankarian, S. Abdullah, Kidney-inspired algorithm for optimization problems, Communications in Nonlinear Science and Numerical Simulation 42(2017) 358–369. doi: 10.1016/j.cnsns.2016.06.006.

[19] Y. Tian, R. Cheng, X. Zhang, Y. Jin, PlatEMO: A MATLAB platform for evolutionary multi-objective optimization, IEEE Computational Intelligence Magazine 12(2017) 73–87. doi: 10.1109/MCI.2017.2742868.

[20] E. Fedorov, O. Nechyporenko, Methods for Solving the Traveling Salesman Problem Based on Reinforcement Learning and Metaheuristics, in: CEUR Workshop Proceedings, 2022, volume 3309, pp. 94–103.

[21] T. Neskorodieva, E. Fedorov, M. Chychuzhko, V. Chychuzhko, Metaheuristic Method for Searching Quasi-Optimal Route Based On the Ant Algorithm and Annealing Simulation, Radioelectronic and Computer Systems 1(2022) 92–102. doi: 10.32620/reks.2022.1.07.

[22] A. Shukla, R. Tiwari, Discrete Problems in Nature Inspired Algorithms, 1st ed., Boca Raton, FL: Chapman and Hall/CRC, New York, 2019. doi: 10.1201/9781351260886

[23] A. Slowik, Swarm Intelligence Algorithms, A Tutorial, 1st ed., Boca Raton, FL: Chapman and Hall/CRC, New York, 2021. doi: 10.1201/9780429422614

[24] O. Bozorg Haddad, M. Solgi, H. Loaiciga, Meta-heuristic and Evolutionary Algorithms for Engineering Optimization, Hoboken, New Jersey: Wiley & Sons, 2017. doi: 10.1002/9781119387053.

[25] K.-L. Du, M. N. S. Swamy, Search and Optimization by Metaheuristics. Techniques and Algorithms Inspired by Nature, Charm: Springer, 2016. doi: 10.1007/978-3-319-41192-7.

[26] A. Kaveh, T. Bakhshpoori, Metaheuristics Outlines, MATLAB Codes and Examples, Cham: Springer, 2019. doi: 10.1007/978-3-030-04067-3.

[27] A. Nayyar, D.-N. Le, N. G. Nguyen, Advances in Swarm Intelligence for Optimizing Problems in Computer Science, 1st ed., Boca Raton, FL: Chapman and Hall/CRC, New York, 2018. doi: 10.1201/9780429445927.