VOLUME 16, N° 1 2022
Journal of Automation, Mobile Robotics and Intelligent Systems

DEVELOPMENT OF A MODIFIED ANT COLONY ALGORITHM FOR ORDER SCHEDULING IN FOOD PROCESSING PLANTS

Submitted: 14th September 2021, Accepted 24th February 2022

Igor Korobiichuk, Serhii Hrybkov, Olga Seidykh, Volodymyr Ovcharuk, Andrii Ovcharuk

DOI: 10.14313/JAMRIS/1-2022/6

Abstract:
This developed modified ant colony algorithm includes an additional improvement with local optimization methods, which reduces the time required to find a solution to the problem of optimization of combinatorial order sequence planning in a food enterprise. The planning problem requires consideration of a number of partial criteria, constraints, and an evaluation function to determine the effectiveness of the established version of the order fulfillment plan. The partial criteria used are: terms of storage of raw materials and finished products, possibilities of occurrence and processing of substandard products, terms of manufacturing orders, peculiarities of fulfillment of each individual order, peculiarities of use of technological equipment, expenses for storage and transportation of manufactured products to the end consumer, etc. The solution of such a problem is impossible using traditional methods. The proposed algorithm allows users to build and reconfigure plans, while reducing the time to find the optimum by almost 20% compared to other versions of algorithms.

Keywords: *order fulfillment planning, modified ant colony algorithm, efficiency of the algorithms, optimization, food industry.*

1. Introduction

The food industry is one of the most strategic industries in any country, because it ensures the nation's food independence. The food industry market is characterized by very high competition, in particular between domestic and foreign producers. In the face of fierce competition, food industry enterprises, which are trying to work on long-term contracts, face the acute task of keeping consumers, which is possible only by making quality products at a reasonable price and providing services for a short period of time. This requires the purchase and storage of raw materials, which have a limited period of use, to make products in compliance with all necessary technological requirements, and to ensure storage of manufactured products. Under these conditions, enterprise management must ensure flexible and responsive production with maximum consideration of external and internal factors and constraints in production planning and operational decision-making [1].

Most technological and production processes in the food industry are difficult to describe quantitatively, which, together with the multi-criteria nature of planning and control tasks, complicates the use of traditional deterministic mathematical methods to make appropriate decisions [2, 3]. This necessitates the use of new intelligent methods for solving management problems based on operational information from specialist experts regarding the prioritization of any alternatives in the decision-making process.

It is important to note that the industry problem of improving the efficiency of food enterprises requires a comprehensive solution, which includes a set of problem-oriented models, methods and algorithms, management technologies and appropriate information and software [4-6].

It should also be noted that very often, the process of planning the fulfillment of orders for the manufacture of products in food enterprises takes place under conditions of uncertainty and risk.

Estimation of efficiency of the established version of the plan requires consideration of external and internal influences.

For the criteria, which are estimated in monetary units, the convolution with the prioritization of each of them for each individual case is applied [7].

The total estimated function will be represented by the additive convolution of all criteria (1):

$$F_0' = \lambda_1 F_1 - \sum_{\gamma=2}^{7} \lambda_\gamma F_\gamma \to \max, \tag{1}$$

where λ_γ is the criterion importance ratio $\lambda_\gamma \in (0,1)$, $\Sigma\lambda_\gamma = 1$;

F_1 is the total maximum profit from fulfillment of orders for production for a given planning period;

F_2 is the total amount of penalties for late fulfillment of the order for a given planning period;

F_3 is the total amount of costs when fulfilling the order for a given planning period;

F_4 is the total costs of maintenance and service of equipment not used in the given planning period;

F_5 is the total costs for processing and disposal of substandard products or production waste in the fulfillment of all orders for a given planning period;

F_6 is the total costs of storage and delivery of the end product to the final consumer for a given planning period;

F_7 is the total storage costs of raw materials and supplies in the enterprise for a given planning period.

In addition, it is necessary to ensure the fulfillment of each order to a given moment of shipment, and to minimize downtime of the main equipment.

Most often, a manager should form an order fulfillment plan, taking into account all the operations of the technological process in the manufacture of products, from the supply of raw materials to the delivery of finished products to the end consumer. Depending on the situation and the type of enterprise, the manager can adjust the necessary set of partial criteria, as well as take into account the necessary features of the parameters of order fulfillment. The planning task can also arise in case of emergency situations, which require reconfiguration of the existing order fulfillment plan in order to minimize losses in case of downtime or repair of certain technological equipment. Such a problem is a complex combinatorial multi-criteria optimization problem. To solve such problems, it is reasonable to use modified heuristic and metaheuristic algorithms [7-9].

Thus, the actual scientific and applied problem is the development and use of modified algorithms and methods to ensure reduction of time for decision-making, reducing the impact of the human factor and reducing costs and time to fulfill orders for the manufacture of products by the food enterprise.

2. Materials and Methods of Research

In a 2020 paper by Hrybkov et al., the information technology for solving the problem of planning the fulfillment of orders for manufacturing products in food enterprises in conditions of uncertainty and risk is proposed [7]. The information technology for solving the problem of order fulfillment planning is based on the method of data mining, modified multi-agent and genetic algorithms. A combined algorithm for solving the planning problem was developed to take into account the peculiarities of the subject domain. The paper does not consider other modifications of multi-agent algorithms, nor does it investigate modifications of the ant colony algorithm.

Another study by Hrybkov et al. and Kharkianen et al., the authors proposed a mathematical model of the problem of planning the fulfillment of contracts, and suggested the use of a modified ant colony algorithm [8, 9]. However, the mathematical model and the solution were directed to non-food industry enterprises.

In a paper by Senthilkumar et al., the authors proposed a hybrid planning algorithm based on the particle swarm algorithm and the ant colony algorithm. The proposed algorithm does not take into account the specifics of the food enterprise [10].

A 2013 paper aimed to solve the planning problem using the bee colony algorithm, and the local search is based on a greedy constructive-destructive procedure [11]. This approach cannot be applied when it is necessary to take into account the priorities of partial criteria.

The mathematical model given in a paper by LIn does not take into account all of the partial efficiency criteria for planning the fulfillment of orders, and does not take into account the supply of the final product and its storage [12].

Another study considered the application of the ant colony algorithm to solve the problem of production planning in metallurgical plants, but this approach requires adaptation for food enterprises [13].

Another paper consolidated the approaches to production planning of various products [14]. The paper highlights the factors affecting the solution of the planning problem using classical optimization methods, but is not devoted to planning in the enterprises of the food industry.

A 2014 paper by Pintea is devoted to the application of solving planning problems using ant colony algorithms, but does not consider modifications of the ant colony algorithm [15].

Having analyzed the literature reviewed above, we may argue for the feasibility of creating and using a modified ant colony algorithm to solve the problem of planning the fulfillment of orders for the manufacture and delivery of products in food enterprises in a given time to minimize costs.

The main management task, which requires considering the task and constraints of all levels, is the operational adjustment of the production plan to fulfill orders. As a rule, the solution of such a task was reduced by decomposition to the required level. In this case, only the necessary criteria and functions with constraints were taken into account, but the connection of all impact criteria was lost.

The production plan is optimal if its fulfillment provides the maximum profit for a given period of time. The optimal plan of order fulfillment does not violate the overall strategic plan of the enterprise, minimizes variable costs, and allows the maximum use of production and technological equipment.

This problem belongs to the class of hierarchical multi-criteria optimization problems, including various partial criteria, which together influence the choice of the optimal solution and the use of which is determined by specific conditions [7-9].

Depending on the social and economic situation, as well as the characteristics of the manager responsible for drawing up an operational and calendar plan for order fulfillment, the problem can be solved in different ways [7, 8]:

- all the criteria are considered and ranked in the evaluation of the effectiveness of the operational plan of order fulfillment, or
- the task is simplified to the selection of certain partial criteria.

The complexity of solving such a problem increases with the number of orders, as well as with the stages of different execution options at different technological links of production.

The task can be simplified only at enterprises where the technological process of product manufacturing takes place at one automated technological complex having a continuous conveyor production cycle. An example would be pasta production, where a certain type of product is manufactured on a single automated production line.

It should be noted that some of the equipment used in domestic food enterprises requires additional adjustment efforts when introducing new types of products. In that case, the cost of adjusting and setting up the process equipment according to each product manufacturing process is added.

3. The Results of Research

We propose a modified "ant colony" algorithm to solve the problem of order fulfillment planning in a food enterprise.

The "ant colony" algorithm is based on the principle of collective intelligence as exemplified by their behavior in finding optimal routes for finding food. In the task of making an operational plan for order fulfillment, the best plan, which is optimal according to the given criteria and constraints, acts as a food. Each version of the schedule corresponds to one ant, performing during the algorithm the construction of its own route, which corresponds to the sequence of order fulfillment. Among the obtained variants of alternative solutions, the best options are selected according to the value of the target function or given criteria. The information obtained is accumulated and used by the ants in the following iterations. Each ant performs its actions according to the rules of a probabilistic algorithm, and uses statistical information when choosing the next step, which reflects the previous history of the collective search, not only based on the target function [9, 13].

First, all orders are divided according to the possibility of their fulfillment on certain equipment, and operational planning for certain equipment and orders begins. A multilayer graph is taken as the basis, where each of the layers corresponds to a certain level of task decomposition.

The first level has representations of the order level, where the node denotes the order number; the edges have directions reflecting the sequence of order fulfillment, namely, the transition from one order to another. The start time of the next order is determined based on the completion of the first stage of the current order according to the technological path of production [9, 13].

The transition from the a and b vertices of the graph determines the time that must be spent to pass the technological operations of manufacturing products by order a without downtime and overlapping operations in the fulfillment of order b [8, 9, 13].

The second level has detailing, which displays all possible options for the fulfillment of orders received from the customer in the form of a graph, where we denote an edge — the department or technological line l, on which a certain technological operation j can be performed.

The node of the graph is the intermediate state in which the semi-finished product is at the transition between technological operations. The transition between the nodes of the graph determines the time Δt_{ijl}, required to pass the entire batch of semi-finished product for the implementation of the corresponding i technological operation of the j step on the l equipment [9, 13].

The third level of decomposition reflects the ordering sequence graph for each l equipment.

The application of the modified "ant colony" algorithm determines, depending on the production and conditions that are determined by the decision maker, the search at different levels. In the simplest variant only the first level is used, which implies the use of only a complete automated line that performs all the technological operations for production [9, 13].

Due to the repeated iterative search, at each iteration, the best option among the current ones is selected and compared with the global value. There is also a control of the best sequence of order fulfillment for a given number of ants. In this case, each ant corresponds to one of the variants of the schedule of order fulfillment, which means that, when performing one iteration of the algorithm, it is this ant that forms it. Positive feedback is implemented as an imitation of the ant behavior, that is, "leave traces – move by traces." The more traces are left on the path, the more ants will move along it [8, 9, 13, 17].

New traces appear on the path (corresponding to the order fulfillment sequence), attracting additional ants. The positive feedback is realized by the following stochastic rule: the probability of including an edge of the graph in the ant's path is proportional to the number of pheromones on it. This rule ensures the randomness of the formation of new variants. The number of pheromones deposited by an ant on an edge of the graph is inversely proportional to the efficiency of the schedule. The more efficient the schedule is, the more pheromones will be deposited on the corresponding edges of the graph, and more ants will use them when synthesizing their routes. The pheromones deposited on the edges allow for efficient routes (execution sequences) to be stored in the ant's global memory. Such routes can be improved in the subsequent iterations of the algorithm [8, 9, 18-20].

Using only positive feedback leads to prematurely finding a convergent solution when all ants move along the same suboptimal route. To avoid this, negative feedback (pheromone evaporation) is used. The evaporation time should not be too long, because the previous situation with finding a suboptimal route may occur. A very short evaporation time will lead to a loss of colony memory, thus not ensuring cooperative behavior of the ants.

Cooperativity is important for the algorithm: multiple identical ants simultaneously explore different points in the solution space and communicate their experiences through changes in the cells of the ant's global memory. For each ant, the transition from node a to node b depends on three components: ant memory, visibility, and virtual pheromone trace. For the second level, a list of operations to be performed emerges [2, 3, 7, 10-14].

An ant's memory is a list of orders already included in execution that should not be considered when scheduling. Using this list, an ant is guaranteed not to choose the same order. At each step, the memory of the ant grows, and at the beginning of each new iteration of the algorithm the memory becomes empty. Let us designate through the $Lt_{a,i}$ the list of orders already included in the schedule of i ant ($Lt_{a,i} \in n$, where n is all orders) after the inclusion of the a order, and the list to be included in the schedule is $Ln_{a,i}$ ($Ln_{a,i} \in n$, where n is all orders). That is $Lt_{a,i} \cup Ln_{a,i} = n$, herewith $Lt_{a,i} \cap Ln_{a,i} = 0$. For the second and third levels of graph decomposition, the list of j stage number from the set of stages is taken into account $j \in w_i$ for i order, w_i is the number of steps required for i order, as well as the l number of equipment from the set of equipment ($j \in \sigma_i$) for i order, σ_i is the amount of equipment involved to perform all stages in the manufacture of the i order [10, 13, 16].

Visibility is the value inverse of the evaluation of edges of graph a and b, namely $\eta_{a,b}(\iota) = 1/D_{a,b}(\iota)$, where $D_{a,b}(\iota)t$ is the value of the graph edge estimation (depending on the level of management, decomposition can be represented in cost or time form) between vertices a and b (the more optimal the value, the more preferable it is to choose this transition) at iteration [8, 9, 13, 16-20].

The virtual pheromone trail on the edge (a, b) reflects the desire to visit vertex b from vertex a based on ant experience. Unlike visibility, the pheromone trail is a more global and dynamic information. It changes after each iteration of the algorithm, reflecting the experience accumulated by ants [16]. The amount of virtual pheromone on edge (a, b) at iteration, which leaves i insect, is denoted by $\xi_{a,b}(\iota,i)$. In the first iteration $\iota = 0$ of the algorithm operation, the amount of pheromone for each i ant is taken equal to a small positive number $0 < \xi_{0,b}(0,i) < 1$.

The transition probability of i ant to vertex b from vertex a at the ι iteration is determined by Formula (2) [10, 12-14].

$$P_{a,b}(\iota,i) = \begin{cases} \dfrac{\left[\xi_{a,b}(\iota,i)\right]^{\alpha}\left[\eta_{a,b}(\iota,i)\right]^{\beta}}{\sum_{b=0}^{Ln_{a,i}}\left(\left[\xi_{a,b}(\iota,i)\right]^{\alpha}\left[\eta_{a,b}(\iota,i)\right]^{\beta}\right)}, \text{ if } b \in Ln_{a,i} \\ 0, \quad if \quad b \notin Ln_{a,i} 0 \end{cases}, \tag{2}$$

where α is pheromone weight ratio, $0 \le \alpha \le 1$, which determines the relative importance of the mark intensity influence on the choice of the path. At $\alpha = 0$ the shortest edge (according to the evaluation of the problem – time, cost) will be selected for the transition that corresponds to the greedy algorithm. At $\alpha = 1$ the edge with the highest level of marks will be selected.

β is the visibility coefficient in route selection $0 \le \beta \le 1$, which determines the relative importance of the influence of visibility on the choice of the path, when $\beta = 0$ only pheromone amplification will work, which will lead to finding a local suboptimal solution: $\alpha + \beta = 1$, with the coefficient α determining the greediness of the algorithm, and β being hardiness.

Formula (2) defines only the probability of selection of the next order to include it in the execution schedule, which will correspond to the graph of selection of the next vertex. The value obtained by Formula (2) does not change over the course of the algorithm iteration, but in two different ants, the value of transition probability will differ, because they have different list of visited vertices [16-19].

After completing the route, each ant deposits on each edge the number of pheromones that are included in its route and meet the schedule, which are calculated by Formula (3).

$$\Delta\xi_{a,b}(\iota,i) = \begin{cases} \dfrac{1}{F_0'} * \dfrac{Q}{S_i(\iota)}, if\ (a,b) \in Lt_i(\iota) \\ 0, if\ (a,b) \notin Lt_i(\iota) \end{cases}, \tag{3}$$

where a and b are the indices of the pair of nodes uniting the edge the agent passed; $Lt_i(\iota)$ is the formed route at the ι iteration of the i ant; $S_i(\iota)$ is route length $Lt_i(\iota)$, expressed in time or cost; Q is the adjustable parameter approximating the optimal route, given or calculated by the previous iteration; and $F_{?}'$ is the evaluation of the route according to the selected partial criteria or evaluation function.

In the case that the number of pheromones deposited was not excessive, they are updated according to Formula (4), taking into account the pheromone level update coefficient $0 \le \rho \le 1$, which determines its relative decrease over time.

$$\Delta\xi_{a,b}(\iota+1) = (1-\rho) * \Delta\xi_{a,b}(\iota) + \sum_{i=1}^{n}\Delta\xi_{a,b}(\iota,i), \tag{4}$$

where n is the number of ants, which corresponds to the number of orders, and $\rho \in [0,1]$ is the pheromone evaporation coefficient.

To protect against prematurely finding a suboptimal solution, it is proposed to introduce restrictions on the concentration of pheromones on the edges $\Delta\xi_{\min} \le \Delta\xi_{a,b} \le \Delta\xi_{\max}$.

The formula for updating the number of pheromones on graph edges turns from expression (3) to Formula (5) [2, 11-14].

$$\begin{aligned}\Delta\xi_{a,b}(\iota+1) &= (1-\rho) * \xi_{a,b}(\iota) + \Delta\xi_{a,b,best} = \\ &= (1-\rho) * \xi_{a,b}(\iota) + \frac{1}{Lt_{best}(\iota)},\end{aligned} \tag{5}$$

where $Lt_{best}(\iota)$ is the best formed route on the ι iteration.

After each iteration of the algorithm, only one ant leaves a trail to choose from: with the best at the current iteration being $Lt_{best}(\iota) = Lt_{\iota\_best}(\iota)$, and the best for all the time the algorithm works being $Lt_{best}(\iota) = Lt_{gl\_best}(\iota)$. It is reasonable in the first iterations to use $Lt_{\iota\_best}(\iota)$, and in the last iterations to use $Lt_{gl\_best}(\iota)$.

At each iteration, the amount of pheromone on each edge of the graph is adjusted according to Formula (6).

$$\xi_{a,b}(\iota) = \begin{cases} \xi_{\min}, \text{ if } \xi_{a,b}(\iota) < \xi_{\min} \\ \xi_{a,b}(\iota), \text{ if } \xi_{\min} \le \xi_{a,b}(\iota) \le \xi_{\max} \\ \xi_{\max}, \text{ if } \xi_{\max} < \xi_{a,b}(\iota) \end{cases}. \tag{6}$$

Instead of the empirical rule of choosing the best ant, it is proposed to use Formula (7) based on the

Cauchy probability distribution law, which allows for a smooth transition from the technology of global search, at the initial steps, to the technology of selecting the best global search for the whole time of the algorithm at the final steps.

$$P_{term}(\iota)=\frac{1}{\pi}arctg\left(\frac{\iota-\frac{\iota_{\max}}{2}}{\iota-\frac{\iota_{\max}}{10}}\right)+0.5, \qquad (7)$$

where $\iota_{\max}$ is the selected number of generations of colony life.

Based on a comparison of their values, either the best ant for the entire time of the algorithm $Lt_{gl\_best}(\iota)$ or the best at the current iteration $Lt_{\iota\_best}(\iota)$ is selected. The diversification operations are used if no improvement in the global solution has occurred for every $\iota_{\max}/2$ or $\iota_{\max}/10$ iteration.

During the diversification operation, the amount of pheromone on all edges is re-initialized. The maximum pheromone value is determined using Formula (8) by assigning them a value inverse to the evaporation coefficient and multiplied by an estimate of the global optimum path [16-20].

$$\xi_{\max}=\frac{1}{\rho * Lt_{gl\_best}(\iota)}. \qquad (8)$$

The minimum value of the pheromone concentration on the rib is calculated by Formula (9).

$$_{\min}=\frac{_{\max}}{2*n}, \qquad (9)$$

where n is the number of ants, which corresponds to the number of orders.

After each finding of the best solution the values of $\xi_{\max}$ and $\xi_{\min}$ should be listed by formulas (8) and (9), respectively [16].

When constructing a route, the selection of the transition from the a node is carried out on the basis of the rule (8), but for the selection of the next node, the entire list of nodes $Ln_{a,i}$ ($Ln_{a,i} \in n$, where n is all orders), that i ant still has to visit is not used; only a list of the closest peaks, *nlist,* is used. The list *nlist* is a matrix of size $n{\times}n'$, where n is the number of vertices in the graph, and n' is the number of closest vertices.

Each i row of this matrix contains numbers of the nearest vertices ordered by the distance from a vertex. Since *nlist* is calculated and formed by the known distance matrix D at the beginning of the algorithm, the construction of the route at the stage of selecting the next vertex is significantly accelerated. If all vertices of the *nlist* set are exhausted, the selection of vertex b will be determined by taking into account the distance and level of edge pheromones according to Formula (10) [16-20].

$$b=\max_{b\in Ln_{a,i}}\left\{\left[\xi_{a,b}(\iota,i)\right]^{\alpha}\left[\eta_{a,b}(\iota)\right]^{\beta}\right\}. \qquad (10)$$

So, considering all the modifications, the solution algorithm will appear as follows [7, 8].

1. Selection of the planning period. According to a given period, orders are selected, which should be performed during this period on certain equipment.

2. Selection of the evaluation function and partial selection criteria for solving the problem.

3. Formation of a multilayer graph, describing each of the layers responsible for a particular level of decomposition of the problem.

4. Initialization of algorithm parameters α, β, ρ and Q.

5. Selecting rules for calculating visibility parameters $\eta_{a,b}$ and pheromone concentration $\xi_{a,b}$.

6. All orders are sorted by time parameter dt_i, for which the products of the i order must be produced. As a result we get a sorted order list $Ln_{a,i}$ ($Ln_{a,i} \in n$, where N is all orders).

7. We perform the construction of the initial routes simultaneously in the forward and reverse directions. Construction of the forward direction implies the sequence obtained after sorting. In fact, we take as such a sequence the sorted list of orders $Ln_{a,i}$ ($Ln_{a,i} \in n$, where N is all orders). And the construction of the reverse direction implies formation from the last to the first order. When building in reverse order, an additional condition for selecting the next order is the time dt_i, for which it is necessary to produce products under the i order. After that we carry out the reversal. Thus, we obtain the value of $Lt_{gl\_best}(\iota)$ global and $Lt_{\iota\_best}(\iota)$ local (at the current iteration) optimum.

8. We form a population of ants, each of which on the first vertex corresponds to a certain order, which is formed using a random number generator. In fact, each ant at the beginning of its path must take a path on the graph corresponding to a certain order.

9. Execute the cycle on the life time of the colony $\iota_{\max} \Leftarrow 1..\iota$.

9.1. Performing the cycle on all ants $i \Leftarrow 1..n$.

9.1.1. Construct a route for each new ant using formula (2), and calculate the length of $Lt_\iota(\iota)$.

9.1.2. Apply a local 2-opt and/or 3-opt search to the route.

9.1.3. Carry out isolation of pheromones according to (5).

9.2. We evaluate each of the routes and compare them to the local and global optimal value to update the others.

9.3. Perform pheromone updates on all edges of graph (4).

10. Deriving the best local and global optimum.

11. Completion of the algorithm.

After completing the algorithm, we obtain the optimal plan for order fulfillment in the form of local and global optimum route.

As can be seen from the above algorithm, after the route has been constructed, an additional improvement is carried out over the obtained solution by one of the local optimization methods, namely by 2-opt and 3-opt methods [21, 22].

The modified "ant colony" algorithm in the form of a block diagram is shown in Fig. 1.

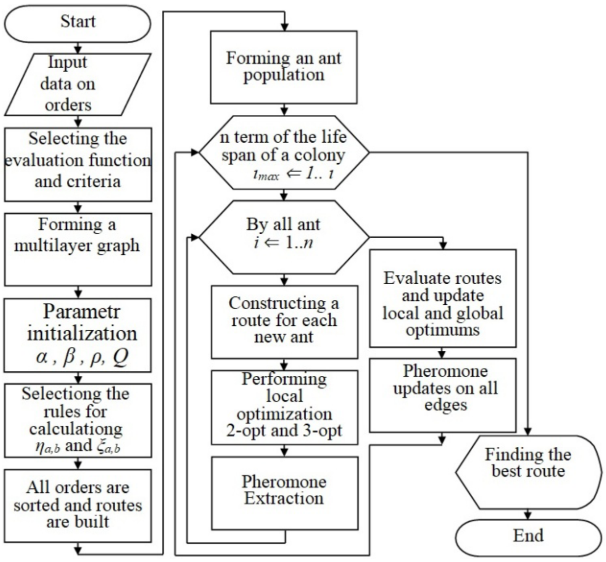

Fig. 1. Modified "ant colony" algorithm

At the beginning of the "ant colony" algorithm, it is necessary to set whether one of the 2-opt or 3-opt methods will be automatically selected. One must use both methods and choose the best.

Local optimization by the 2-opt method (Fig. 2) consists of selecting two pairs of non-adjacent schedule elements and swapping them with each other, then evaluating and leaving the best option. If after swapping we get the best option, we leave it [22], which is shown in Fig. 2.

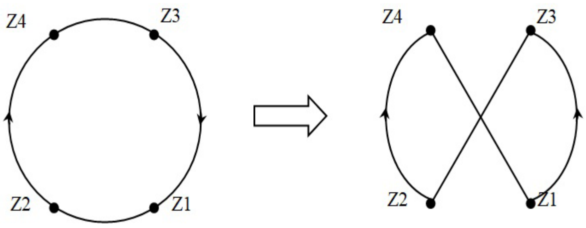

Fig. 2. Schematic use of the 2-opt method

The application of local optimization by the 2-opt method should be carried out for each variant of the plan, which in the ant algorithm corresponds to the route of the ant.
The plan variant searches for two non-contiguous sequences, the exchange of which will provide the maximum effect according to the evaluation function or set of criteria. An example application of the 2-opt method is shown in Fig. 3.

Option plan for the 2-opt method	1	2	5	6	4	3
Variant of the plan after applying the 2-opt method	1	3	5	6	2	4

Fig. 3. Schematic use of the 2-opt method

The 3-opt method (Fig. 4) uses the approach to the 2-opt method, but selects 3 pairs of non-adjacent schedule elements to replace them with each other [22].

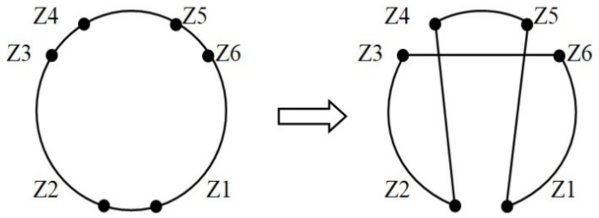

Fig. 4. An example of using the 3-opt method

This improves the found route at each iteration, which will reduce the total number of iterations of the algorithm, as well as the total running time of the algorithm.

4. Discussion of the results of approbation of the improved mathematical model and the modified ant colony algorithm

The modified ant colony algorithm was tested at various enterprises of the food industry, and its performance was compared to the following methods:
The "bee colony" algorithm [21];
the "chaotic bat" algorithm [23-26];
the "bat algorithm based on the Levy flight search strategy" [24-26];
the "bat algorithm based on reduction factor" [25-26];
the "genetic algorithm" [27-28];
the "pack of wolves" algorithm [29-32];
the "ant colony algorithm using elite ants" [8];
the modified "ant colony algorithm."

To test the modified algorithm, the corresponding software modules were created implementing the modified algorithm in the Java language. Most tests were performed on the Intel Core i5-4690K (3.5GHz) / RAM 16GB / NDD 1TB.

In order to obtain reliable information when applying the considered algorithms, data on order fulfillment plans for different previous periods of enterprises activity were selected. 25, 50, and 100 orders were randomly selected for different periods of time. At the beginning of the algorithms, the orders were placed in the same order as they came into the enterprise.

To compare the algorithms, each algorithm was first matched to the selected sample, and then the results of the algorithms were compared.
The efficiency of the algorithms was evaluated on the basis of the following indicators:
- time to find the optimal schedule;
- the effectiveness of the found plan in monetary terms as the difference between the estimates by the target function of the calculated plan and the actual plan;
- the efficiency of order fulfillment time reduction (calculated as the difference between the calculated plan and the actual plan in terms of execution time).

The efficiency of the found plan is calculated by Formula (11), and the efficiency percentage is calculated by Formula (12).

$$f_E = f_{opt} - f_{fact}\ , \tag{11}$$

$$\varepsilon = \frac{f_{opt} - f_{fact}}{f_{opt}} 100\%\ , \tag{12}$$

where f_{fact} is the evaluation of the actual plan by the target function; f_{opt} is the evaluation of the established plan by the target function.

Tables 1 shows a comparison of the use of algorithms for 100 orders.

Tab. 1. Comparison of algorithms at 100 orders

No.	Algorithm name	Plan search time, min	The effectiveness of the plan found		Reduced turnaround time	
			standard units	%	Hour	%
1	The Bee Colony Algorithm	7.8	2156	4	3	2
2	The chaotic bat algorithm	7.7	2364	4.39	3	2
3	A bat algorithm based on the Levy flight search strategy	7.55	2345	4.35	3	2
4	A bat algorithm based on the reduction factor	7.6	2163	4.01	3	2
5	Genetic algorithm	7	2420	4.49	6	4
6	Wolf pack algorithm	6.8	3523	6.99	6	4
8	Ant colony algorithm using elite ants	6.8	3523	6.99	6	4
9	Modified ant colony algorithm	6.51	3523	6.99	6	4

According to the results of calculations for 25 orders, it was found that all algorithms found the same plan, which is 2% more efficient than the actual plan, and its total execution time was 2 hours less than the actual plan. The search speed of our algorithm compared to the other algorithms was 2.56% faster.

According to the results of calculations for 50 orders, the search results for the algorithms were found to be different. The plan efficiencies for the "bee colony," "chaotic bat," "bat based on Levy flight search strategy," and "bat based on reduction factor" algorithms were almost the same and minimal compared to other algorithms. The best result was shown by our modified "ant colony" algorithm. Its search speed compared to the first 4 methods was 9.25% faster, and compared to its unmodified versions, it was 1.8% faster.

From the data shown in Table 1, we can see that the search results for the algorithms are different. The best results were shown by the modified "ant colony" algorithm. Its search speed is almost 20% faster, and in comparison with its unmodified versions, is faster by 4.45%.

The proposed modified algorithm is based on the combination of a metaheuristic algorithm with a heuristic one. Most other modifications try to avoid this type of modification to prevent an increase in algorithm complexity [33]. The main advantage of the proposed modified algorithm lies in reducing the time to find a solution to the problem, which is very relevant in real production conditions when critical situations arise.

The main disadvantages of the proposed modification of the algorithm include that its application is possible using large amounts of data, but without the use of special means of storage, such as a database management system, it is impossible, because it is necessary to constantly save a multitude of additional calculations.

Further advancement of research and development is aimed at the inclusion and use of the proposed algorithm in the decision support system for the management of the food enterprise, as well as in the information technology presented in the paper by Hrybkov et al [7].

5. Conclusion

The developed modified ant colony algorithm allows for the reconfiguration of plans, while reducing the time to find the optimum by almost 20% compared to other versions of the algorithms. When unmodified versions are used, the time to find the optimum increases by 4.5%. Modification of the "ant colony" algorithm reduces the time by using local optimization approaches.

The greatest effect of the use of the proposed modified algorithm is achieved in the decision support systems for planning the sequence of order fullfillment, which will allow for: quick formation of operational and calendar plan of order fulfillment with cost minimization and profit maximization; operational adjustment of the existing calendar plan of orders, allowing to react to the order in real time and ensure optimal use of technological equipment; efficiency in the use of raw materials and supplies, as well as minimizing storage costs; rapid response to negative and contingency situations by making appropriate changes to the current order fulfillment plan; clear distribution of all tasks for each order between production departments, which allows to take into account the sequence of fulfillment and necessary resources with time constraints; and optimal use of production capacities.

The use of the proposed modified algorithm is possible in various tasks of work sequences and related planning.

AUTHORS

Igor Korobiichuk* – ŁUKASIEWICZ Research Network – Industrial Research Institute for Automation and Measurements PIAP, Jerozolimskie 202, 02-486 Warsaw, Poland, e-mail: igor.korobiichuk@piap.lukasiewicz.gov.pl.

Hrybkov Serhii – Ukrainian State University of Food Technologies, 68 Volodymyrska Street, 01033, Kyiv, Ukraine, e-mail: sergio_nuft@nuft.edu.ua.

Olga Seidykh – Ukrainian State University of Food Technologies, 68 Volodymyrska Street, 01033, Kyiv, Ukraine, e-mail: olgased@ukr.net.

Volodymyr Ovcharuk – Ukrainian State University of Food Technologies, 68 Volodymyrska Street, 01033, Kyiv, Ukraine, e-mail: ovcharuk2004@ukr.net.

Andrii Ovcharuk – Ukrainian State University of Food Technologies, 68 Volodymyrska Street, 01033, Kyiv, Ukraine, e-mail: ovch2011@gmail.com.

* Corresponding author

REFERENCES

[1] A. Oliinyk, S. Skrupsky, S. Subbotin, I. Korobiichuk, "Parallel Method of Production Rules Extraction Based on Computational Intelligence", *Automatic Control and Computer Sciences*, 2017, Vol. 51, No. 4, 2017, pp. 215–223. 10.3103/S0146411617040058

[2] I. Korobiichuk, A. Ladaniuk, V. Ivashchuk, "Features of Control for Multi-assortment Technological Process", n: Szewczyk R., Krejsa J., Nowicki M., Ostaszewska-Liżewska A. (eds) *Mechatronics 2019: Recent Advances Towards Industry 4.0. MECHATRONICS 2019. Advances in Intelligent Systems and Computing*, vol 1044. Springer, Cham, 2020, pp. 214-221. 10.1007/978-3-030-29993-4_27

[3] I. Korobiichuk, A. Ladanyuk, N. Zaiets, L. Vlasenko, "Modern development technologies and investigation of food production technological complex automated systems". In: *ACM International Conference Proceeding Series. 2nd International Conference on Mechatronics Systems and Control Engineering ICMSCE 2018*, February 21-23, 2018, Amsterdam, Netherlands. 52-57, 10.1145/3185066.3185075

[4] V. Tregub, I. Korobiichuk, O. Klymenko, A. Byrchenko, K. Rzeplińska-Rykała, "Neural Network Control Systems for Objects of Periodic Action with Non--linear Time Programs". In: Szewczyk R., Zieliński C., Kaliczyńska M. (eds) *Automation 2019. AUTOMATION 2019. Advances in Intelligent Systems and Computing*, vol 920. Springer, Cham, pp. 155-164 (2020), 10.1007/978-3-030-13273-6_16

[5] I. Korobiichuk, A. Lobok, B. Goncharenko, N. Savitska, M. Sych, L. Vihrova, "The Problem of the Optimal Strategy of Minimax Control by Objects with Distributed Parameters". In: Szewczyk R., Zieliński C., Kaliczyńska M. (eds) *Automation 2019. AUTOMATION 2019. Advances in Intelligent Systems and Computing*, vol 920. Springer, Cham, 2020, pp. 77-85. 10.1007/978-3-030-13273-68

[6] I. Korobiichuk, N. Lutskaya, A. Ladanyuk, S. Naku, M. Kachniarz, M. Nowicki, R. Szewczyk, "Synthesis of Optimal Robust Regulator for Food Processing Facilities". In: *Advances in Intelligent Systems and Computing*, Vol. 550, ICA 2017: Automation 2017, pp. 58-66. 10.1007/978-3-319-54042-9_5

[7] S. Hrybkov, O. Kharkianen, V. Ovcharuk, I. Ovcharuk, "Development of information technology for planning order fulfillment at a food enterprise", *Eastern-European Journal of Enterprise Technologies*, vol. 1/3, no. 103, 2020, pp. 62–73. 10.15587/1729-4061.2020.195455

[8] S. Hrybkov, V. Lytvynov, H. Oliinyk, "Web-oriented decision support system for planning agreements execution", *Eastern-European Journal of Enterprise Technologies*, vol. 3/2, no. 99, 2018, pp. 13–24. 10.15587/1729-4061.2018.132604

[9] O. Kharkianen, O. Myakshylo, S. Hrybkov, M. Kostikov, "Development of information technology for supporting the process of adjustment of the food enterprise assortment", *Eastern-European Journal of Enterprise Technologies*, vol. 1/3, no. 91, 2018, pp. 77–87. 10.15587/1729-4061.2018.123383

[10] K. Senthilkumar, V. Selladurai, K. Raja, V. Thirunavukkarasu, "A Hybrid Algorithm Based on PSO and ACO Approach for Solving Combinatorial Fuzzy Unrelated Parallel Machine Scheduling Problem", *European Journal of Scientific Research*, vol. 2, no. 64, 2011, pp. 293-313.

[11] F. Rodriguez, M. Lozano, C. García-Martínez, J. González-Barrera, "An artificial bee colony algorithm for the maximally diverse grouping problem", *Information Sciences*, no. 230, 2013, pp. 183–196. 10.1016/j.ins.2012.12.020

[12] Yang-Kuei Lin, "Scheduling efficiency on correlated parallel machine scheduling problems", *Operational Research,* vol. 18, 2018, pp. 603–624. 10.1007/s12351-017-0355-0

[13] T. Zheldak, "Application of the method of modeling the ant colony to the solution of combinatorial problems scheduling the execution of order by metallurgical plants". *Mathematical Machines and Systems*, vol. 4, 2013, pp. 95–106.

[14] G. Georgiadis, A. Elekidis, M. Georgiadis, "Optimization-Based Scheduling for the Process Industries: From Theory to Real-Life Industrial Applications", *Processes,* vol. 7, 2019, pp. 1–35. 10.3390/pr7070438.

[15] Pintea C-M., *Advances in Bio-inspired Computing for Combinatorial Optimization Problem*, Springer: Berlin, 2014, p. 188. 10.1007/978-3-642-40179-4.

[16] S. Talatahari, "Optimum Performance-Based Seismic Design of Frames Using Metaheuristic Optimization Algorithms", *Metaheuristic Applications in Structures and Infrastructures,* 2013, pp. 419–437. 10.1016/B978-0-12-398364-0.00017-6

[17]. K. Jha. Manoj, "Metaheuristic Applications in Highway and Rail Infrastructure Planning and Design: Implications to Energy and Environmental Sustainability", *Metaheuristics in Water, Geotechnical and Transport Engineering*, 2013, pp. 365-384. 10.1016/B978-0-12-398296-4.00016-7

[18] V. Kureichik, A. Kazharov, "Using Fuzzy Logic Controller in Ant Colony Optimization", *Artificial Intelligence Perspectives and Applications,* 2015, pp. 151–158. 10.1007/978-3-319-18476-0_16

[19] V. Kureichik, A. Kazharov, "The Development of the Ant Algorithm for Solving the Vehicle Routing Problems", *World Applied Sciences Journal*, vol. 26, no. 1, 2013, pp. 114–121. 10.5829/idosi.wasj.2013.26.01.13468

[20] T. Stutzle, H. Hoos, "Max-min ant system", *Future Generation Computer Systems,* vol. 8, no. 16, 2000, pp. 889–914.

[21] K. Rocki, R. Suda, "Accelerating 2-opt and 3-opt Local Search Using GPU in the Travelling Salesman Problem", *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing,* 2012, pp. 705–706. 10.1109/CCGrid.2012.133

[22] F. Rodriguez, M. Lozano, C. García-Martínez, J. González-Barrera, "An artificial bee colony algorithm for the maximally diverse grouping problem", *Information Sciences*, 230, 2013, pp. 183–196. 10.1016/j.ins.2012.12.020

[23] I. Fister, D. Fister, X.-S. Yang, "A hybrid bat algorithm", *Electrotechnical Review*, vol. 1–2, no. 80, 2013, pp. 1–7.

[24] G. Wang, L. Guo, "A Novel hybrid bat algorithm with harmony search for global numerical optimization". *Hindawi Publishing Corporation*, 2013, pp. 1–21. 10.1155/2013/696491

[25] X.-S. Yang, X. He, "Bat algorithm: literature review and applications", *International Journal of Bio-Inspired Computation*, vol. 5, no. 3, 2013, pp. 141–149. 10.1504/IJBIC.2013.055093

[26] X.-S. Yang, H. Gandomi, "Chaotic bat algorithm", *Journal of Computational Science*, vol. 5, no. 2, 2014, pp. 224–232. 10.1016/j.jocs.2013.10.002

[27] Y. Zhang, S. Balochian, P. Agarwal, V. Bhatnagar, O. Housheya, "Artificial Intelligence and Its Applications", *Mathematical Problems in Engineering*, vol. 2014, pp. 1–10. 10.1155/2014/840491

[28] Y. Virginia, A. Analía, "Deterministic crowding evolutionary algorithm to form learning teams in a collaborative learning context", *Expert System Appl.*, vol. 10, no. 39, 2012, pp. 8584-8592. 10.1016/j.eswa.2012.01.195

[29] Y. Jie, K. Nawwaf, P. Grogono, "Bi-objective multi population genetic algorithm for multimodal function optimization", *IEEE Trans. Evol. Comput.*, vol. 1, no. 14, 2010, pp. 80–102. 10.1109/TEVC.2009.2017517

[30] W. Hu-Sheng, Z. Feng-Ming, "Wolf Pack Algorithm for Unconstrained Global Optimization", *Mathematical Problems in Engineering*, vol. 2014, pp. 1–17. 10.1155/2014/465082

[31] S. Mirjalili, A. Lewis, "Grey Wolf Optimizer", *Advances in Engineering Software*, vol. 69, 2014, pp. 46–61. DOI: 10.1016/j.advengsoft.2013.12.007

[32] A. Madadi, M. Motlagh, "Optimal Control of DC motor using Grey Wolf Optimizer Algorithm", Technical Journal of Engineering and Applied Science, vol. 4, no. 4, 2014, pp. 373–379.

[33] M. Gendereau, J. I. Potvin, "Handbook of Metaheuristic", 3rd ed., Springer International Publishing: Cham, 2019, p. 604. 10.1007/978-3-319-91086-4