Proceedings of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 693–697 DOI: 10.15439/2025F4025 ISSN 2300-5963 ACSIS, Vol. 43

Knitwear Production Scheduling

Stefka Fidanova
Ruse University "Angel Kanchev",
Ruse, Bulgaria
Centre of Excellence in Informatics and
Information and Communication Technologies
Sofia, Bulgaria
E-mail: stefka.fidanova@iict.bas.bg

Jelenko Stanchov
Institute of Information and
Communication Technology
Bulgarian Academy of Sciences
Sofia, Bulgaria
E-mail: elenko.stanchov@iict.bas.bg

Maria Ganzha
System Research Institute
Polish Academy of Sciences
Warsaw, Poland
E-mail: maria.ganzha@ibspan.waw.pl

***Abstract*—Clothing production is an important sector within manufacturing. It includes the sewing, knitting and leather industries. It is important for a manufacturer to organize the production process well. This organization includes personnel allocation, machine loading, and material allocation. The goal is to complete the given order in the shortest time and, if possible, at the lowest cost. This paper addresses the task of producing knitted shirts. An ant colony optimization algorithm has been proposed to solve the problem. The objective is to complete the order in the shortest possible time.**

***Index Terms*—Feshion production, Optimization, Scheduling**

I. Introduction

OPTIMIZING knitted garment production is a special part of job shop scheduling problem. Job shop scheduling problem is an important problem arising in manufacturing and industry. The production consists of producing various elements/parts on different machines. Sometimes it matters in what order the individual elements are produced. Individual machines may have some specialization, as well as different productivity. Therefore, it is important to find the optimal distribution of the production across machines so that the final product can be produced in minimal time or within a certain period of time and, if possible, at a lower cost. There is a wide variety of job shop scheduling problems, depending on the specific production processes, requirements, and constraints. A survey was conducted in about the various job shop problems and algorithms solving them [1]. This is a typical machine scheduling problem. For the first time, an algorithm for solving a variant of this problem, with machine setup time included, was made in 1954 in [2]. Next years other variants of the problem called flow shop scheduling problem is solved in [3], [4]. These publications solve small-sized problems. There are 1 to 4 operations that are performed on 2 to 4 machines. Exact methods are used to solve the problem. The term job shop was first used by Sisson in 1959 [5].

In general, the job shop scheduling problem can be defined as follows:

There is a set of m machines $M = \{M_1, M_2, \ldots, M_m\}$.

There is a set of n jobs $J = \{J_1, J_2, \ldots, J_n\}$.

The job J_i has a set of operations $O_i = \{O_1^1, O_2^i, \ldots, O_{n_i}^i\}$. The operations have predefined technological sequence.

The operation O_i^j has an operation time p_{ij} and is assigned on machine M_k from the set of machines M.

The solution is a scheduling of the jobs on the machines. The aim is to minimize the completing time of all the jobs.

When the problem is large, it is difficult to solve it in a reasonable time using exact methods. For this reason, metaheuristic methods are applied, which find an approximate solution in a short time.

In this article, we solve a variant of the job shop scheduling problem, related to the production of knitted clothes. To solve the problem, we propose an algorithm based on the ant colony optimization.

The organization of the rest of the paper is as follows. Section 2 gives literature review of the variants of job shop scheduling problems and methods to solve them. Section 3 provides the definition of the considered variant of the job shop scheduling problem. Section 4 describes the algorithm for solving the problem. Computational examples are given in Section 5. Section 6 gives some concluding remarks and direction for future work.

II. Literature Review

When manufacturing a product, we have various measures and constraints. There may be a requirement that jobs be completed in a minimum time or within a given interval. Another requirement may be minimizing the cost of the production, and others. The problem may relate to optimizing one of the measures or to meeting a maximum number of criteria.

In [6] scheduling considers flexible manufacturing system. They solve the problem by applying an adaptive genetic algorithm. Machine flexibility is the focus of the problem solved in [7], [8]. They consider energy efficiency too. Job shop scheduling problem is applied on semiconductor production [9], manufacturing of machines [10], automobile industry [11], metallurgical industry [12].

Wang [13] solves robust job shop problem with a specific structure. Dynamic flexible job shop problem is solved in [14]. Biogeography-based optimization algorithm for solving flexible multi-objective job shop problem has been proposed in [15].

Anghinolfi [16] apply a split greedy heuristic for parallel machine scheduling. In [17] has been proposed a hybrid genetic algorithm for energy adaptive production. Scalia [18] proposed mixed integer programming model for solving job shop scheduling problem. In [19] is applied implementation in Python using *pyomo* library for solving the problem. A review of flexible job shop scheduling problem is done in [20].

III. Knitwear Production Problem

In this article, we consider the problem of optimizing the production of knitted garments. This is a variant of job shop scheduling problem. Suppose there is an order to produce knitted shirts. The shirts come in $s = 3$ sizes: M, L and XL. $N_{i,j}$ units of type i and size j must be produced, where $i = 1, \dots, d$ and $j = 1, \dots, s$. Each shirt consists of $d = 4$ types of parts: front, back, collar and two sleeves. t_{ij} is the time for manufacturing part i of size j, $i = 1, \dots, d$ and $j = 1, \dots, s$. There is a set M of m machines available. $M_{i,j,k}$ is the number of parts of type i and size j that are manufactured on machine k, $i = 1, \dots, d$, $j = 1, \dots, s$ and $k = 1, \dots, m$. All machines are identical. The material used to make the shirts is the same and therefore does not require recharging.

The time during which machine k operates is:

$$T_k = \sum_{i=1}^{d} \sum_{j=1}^{s} M_{i,j,k} * t_{i,j} \tag{1}$$

We assume that all machines start working on the order at the same time. Thus the order completion time is:

$$T = \max_{k=1}^{m} T_k \tag{2}$$

or this is the time by which all the machines will have finished working.

The objective is to minimize T, or to find the minimal order completion time, $\min T$. The solution is described by the matrix M with elements $M_{i,j,k}$, indicating which parts will be manufactured on which machine.

The mathematical description of the problem is as follows: We look for:

$$min \left(max_{k=1}^{m} \left(\sum_{i=1}^{d} \sum_{j=1}^{s} M_{i,j,k} * t_{ij} \right) \right) \tag{3}$$

fulfilling the conditions:

$$\textstyle\sum_{k=1}^{m} M_{i,j,k} = N_{i,j} \quad i = 1, \dots d \quad j = 1 \dots s \tag{4}$$

The aim is to finish the order for the minimal time.

IV. Production Scheduling

The job shop scheduling problem is NP-hard combinatorial optimization problem. This type of problem requires a large amount of computer memory and an exponential number of calculations. For this reason, when the size of the problem is large, it could not be solved by an exact method or traditional numerical method. In such cases, metaheuristic algorithms are applied. Metaheuristic methods are iterative stochastic methods, based on random searches or searches with probability and statistical feedback from the results of previous iterations. Prior knowledge of the task and estimates, if any, are also used to improve and guide the search.

One of the most successful method for solving combinatorial optimization problems is Ant Colony Optimization (ACO) [21], [22]. In the early 1990s, Marco Dorigo proposed a stochastic method for solving combinatorial optimization problems, based on observations of the behavior of ants in nature [23]. The main thing is how ants manage to find the shortest path to a food source, marking the paths with pheromone and following the one with the highest concentration of pheromone. In order for ants to be imitated, the problem to be solved must be reduced to finding the shortest path. This is done by representing the problem in terms of a graph. A shortest path is searched in the graph under certain conditions, using numerical information (weight) on the edges or nodes of the graph, imitating the pheromone in nature. Ants use a combination of pheromone and random search to improve their path.

Artificial ants move to a new node in the task graph using a transition probability. This consists of heuristic information and the pheromone corresponding to that transition.

$$P_{i,j} = \frac{\tau_{i,j}^{a} \cdot \eta_{i,j}^{b}}{\sum\limits_{k \in Unused} \tau_{i,k}^{a} \cdot \eta_{i,k}^{b}}, \tag{5}$$

where P_{ij} is the transition probability to go from node i to node j, η_{ij} is the heuristic information related to the problem and τ_{ij} is the quantity of the pheromone, $Unused$ is the set of nodes which are not used yet in the solution.

It starts with a small positive pheromone value, the same for all elements of the graph τ_0, $0 < \tau_0 < 1$. After each iteration, the pheromone is updated depending on whether the corresponding vertex belongs to a solution and how good this solution is compared to the other solutions found. The main pheromone update rule is:

$$\tau_{i,j} \leftarrow \rho \cdot \tau_{i,j} + \Delta\tau_{i,j}, \tag{6}$$

ρ mimics evaporation in nature, which reduces the amount of pheromone over time. Then, a new pheromone is added $\Delta\tau_{i,j}$, which depends on the quality of the solution and is inversely proportional to the value of the objective function.

The iteration ends when all ants have built their solutions, i.e. the transition probability is 0. The algorithm ends its work when: it has performed its predetermined number of iterations; when there is no improvement in the result for a fixed number of iterations; when the value of the objective function is sufficiently close to a predetermined lower bound.

One of the important elements in applying the ant method is the description of the problem with a graph. The graph we use to describe the problem we are solving is three-dimensional, and node (i, j, k) corresponds to element i of size j manufactured on machine k, $i = 1, \dots, d$, $j = 1, \dots, s$, $k = 1, \dots, m$. This node is assigned a value $M_{i,j,k}$, which indicates how many units of element i of size j to be manufactured on machine k. So if on machine k produces parts of type i, size j, then $M_{i,j,k} > 0$. If on machine k

does not produced parts of type i, size j, then $M_{i,j,k} = 0$. This means that a single part and size can be manufactured on several machines, and different parts with different sizes can be manufactured on one and the same machine.

Graph of the problem is completely connected. An ant begins building the solution from a random node in the graph. At the begining the value of $M_{i,j,k} = 0$, $i = 1, \dots, d$, $j = 1, \dots, s$, $k = 1, \dots, m$, because the distribution of the parts along the machine is not fixed. When the ant passes through node (i, j, k), the value of $M_{i,j,k}$ increases by 1 or $M_{i,j,k} \leftarrow M_{i,j,k} + 1$. The ant can go through the same vertex (i, j, k) multiple times adding 1 to $M_{i,j,k}$. The important thing is not to violate the constraints of the problem.

The pheromone will be associated with the nodes of the graph. It indicates the importance of the node to the solution. Thus, pheromone $\tau_{i,j,k}$ corresponds to node (i, j, k) and its value indicates how important it was for the solutions of the previous iteration. So, the quantity of the pheromone $\tau_{i,j,k}$ indicates how significant the node was for the decisions of previous iterations.

At the beginning, we randomly select one of the elements to be manufactured and assign it to a randomly selected machine. Every time an element i of size j is assigned to machine k, we increase the value of $M_{i,.j,k}$ by 1.

To assign the next elements for fabrication, we select element i of size j at random and we calculate the transition probability to assign it to one of the machines. The transition probability is a product of the amount of pheromone and the heuristic function uses prior knowledge of the problem to guide the search for a good solution. We propose the following heuristic function:

$$\eta_{i,j,k} = \frac{1}{T_k} \tag{7}$$

where the value of T_k is calculated for the current values of the elements of the matrices M by increasing the value of $M_{i,j,k}$ by 1. We calculate the transition probability for the selected element i of size j for all possible machines k. We choose the node (machine) with maximum probability. If there are two or more nodes $(i, j, k_1), \dots, (i, j, k_s)$, $1 \leq s \leq m$, with the same transition probability value, then we choose one of them randomly.

After an element i of size j is selected to be manufactured on machine k, we increase the value of $M_{i,j,k}$ by 1, $M_{i,j,k} \leftarrow M_{i,j,k} + 1$. At the same time we decrease the value of $N_{i,j}$ by 1 $N_{i,j} \leftarrow N_{i,j} - 1$. The current value of $N_{i,j}$ shows how many items of type i and size j have not yet been assigned to the machines for the production.

The algorithm repeats until $N_{i,j} = 0$, for all $i = 1, \dots, d$ and $j = 1, \dots, s$. This means that all the parts have already been assigned for the production on the machines, i.e. the ant has built the solution. The matrix M shows for each type and size of parts on which machine and how many are produced.

The order of the production of the parts is not important, so for each machine k it is possible to produce first all the parts of type 1 and size 1, then all the parts of type 1 size 2 and so on, and at the end all the parts of type d and size s.

When all the ants have built their solutions, we calculate the value of the objective function, i.e. the time to complete the order. Then we update the pheromone. After reducing/evaporating the pheromone with the evaporation parameter ρ, we add a new pheromone.

$$\begin{array}{lll} \Delta\tau_{i,j,k} = & (1-\rho)\frac{1}{T} & \text{if} \quad M_{i,j,k} \neq 0 \\ \Delta\tau_{i,j,k} = & 0 & \text{if} \quad M_{i,j,k} = 0 \end{array} \tag{8}$$

This concludes the current iteration. The algorithm performs a predetermined number of iterations or there is a fixed number of iterations without updating the results. In our problem, updating the results occurs when the time for filling the document is reduced.

V. Computational examples

The organization of the production is very important for its success. Given a variety of elements and a fixed number of the production machines, the elements must be distributed so that the order can be completed in the shortest possible time. In the task of producing knitted garments, there are knitting machines that can produce each of the elements. The material used to make the garments is the same for the entire order. Therefore, there is no delay in moving from one item to another because there is no need to load new material. The garments produced come in three sizes: *M*, *L*, *XL*. The machines available for the production are identical and the same item is produced on each of the machines in the same amount of time. Each garment consists of four parts/elements, a front, a back, a collar and two sleeves. The individual elements can be manufactured on different machines. It is important that for each size the number of fronts is equal to the number of backs, is equal to the number of collar and the number of sleeves is twice as large as the number of other elements. This is a necessary requirement so that the garments can then be assembled. Let the time for making the individual elements be as indicated in Table I.

TABLE I: Knitted shirt production time

element	size M	size L	size XL
front	17 min	18 min	19 min
back	13 min	14 min	15 min
sleeve	10.5 min	11 min	12.5 min
collar	3 min	3 min	3 min

The data on the production time for each element was taken from a Bulgarian company producing knitted garments.

Let the company have 3 machines. The machines are identical and each of them can knit each of the elements. Let an order be received for the production of knitted shirts as noted in Table II.

TABLE II: Knitted shirt order

shirts	size M	size L	size XL
	10 pieces	20 pieces	10 pieces

Each size could be produced on a separate machine. This is one possible solution, but not optimal. If the production of the parts is distributed in this way, each of the machines will complete its assigned work in the following time Table III.

TABLE III: Knitted shirt production time solution 1

shirts	machine 1	machine 2	machine 3
shirt quantity	10M	20L	10 XL
front	10M	20L	10XL
back	10M	20L	10XL
sleeve	20M	40L	20XL
collar	10M	20L	10XL
production time	540 min	1140 min	620 min

With this distribution of work, the order will be completed in 1140 minutes, which is 19 hours. This is the time by which all machines will have finished working. The machine 2 finished last and it determines the order execution time.

Another valid solution is to have machine 1 produces all 10 risques of size M and only 4 of size L. Machine 2 produces 13 risques of size L, and machine 3 produces all 10 risques of size XL and 3 risques of size L. This way, the distribution of work across machines will be almost even. The machines will complete its assigned work in the following time Table IV.

TABLE IV: Knitted shirt production time solution 2

shirts	machine 1	machine 2	machine 3
shirt quantity	10 M + 4 L	13 L	3 L + 10 XL
front	10M, 4L	13L	10XL
back	10M, 4L	13L	10XL
sleeve	20M, 8L	20L	10XL
collar	10M, 4L	13L	10XL
production time	768 min	741 min	791 min

With this distribution of work, the order will be completed in 791 minutes, which is 13 hours and 11 minutes, which is shorter than previous solution. The machine 3 finished last and it determines the order execution time.

Let's apply the ant colony optimization to solving the problem of making knitted shirts. We use 2 ants. Every ant constructs his own solution and after we compare them and choose the better and it will be the iteration best solution. We compare it with the current global best solution and if it is better it is the new global best solution. The ant chose randomly an element from the order and assigns it to randomly chosen machine. After the ant chose randomly other element from the order and calculates the transition probability to be assigned to every one of the existing machines. The element is assigned for the production to the machine with a higher probability. This procedure is repeated till all elements from the order are assigned. After several iterations our ant algorithm achieves following solutions, see Table V and Table VI.

TABLE V: Ant's solution 1

element	machine 1	machine 2	machine 3
front	10M, 4L	14L, 1XL	2L, 9XL
back	10M, 4L	14L, 1XL	2L, 9XL
sleeve	20M, 8L	28L, 2XL	4L, 18XL
collar	10M, 4L	14L, 1XL	2L, 9XL
time	768 min	760 min	770 min

With this distribution of work, the order will be completed in 770 minutes, which is 12 hours and 50 minutes, which is shorter than previous solution. The machine 3 finished last and it determines the order execution time.

TABLE VI: Ant's solution 2

element	machine 1	machine 2	machine 3
front	10M, 4L	14L, 1XL	2L, 9XL
back	10M, 4L	14L, 1XL	2L, 9XL
sleeve	20M, 8L	28L, 2XL	4L, 18XL
collar	10M, 3L	16L, 1XL	1L, 9XL
time	765 min	766 min	767 min

With this distribution of work, the order will be completed in 767 minutes, which is 12 hours and 47 minutes, which is shorter than previous solution. The machine 3 finished last and it determines the order execution time. We observe that the time during which the machines work to fulfill the order is almost the same. The difference in time is within 1 to 2 minutes, which is less than the time for producing any of the elements. Therefore, if we move a part from one machine to another, the total order execution time will not be reduced any further. This gives us reason to assume that the proposed algorithm, based on the ant colony optimization, has found the optimal solution for this example.

VI. Conclusion

This paper proposes an ant colony optimization algorithm for solving the knitwear production scheduling problem. This is a problem coming from the industry. The simplest variant of the task is solved, where we have a single type of machine and the manufactured parts are of the same type and differ only in size. In our future work, we will test the algorithm with large examples with a range of several hundred numbers of elements. We will include the creation of clothes in different colors, which will require a recharge of the machine, which takes time , but it will not require a reconfiguration. Another variant of this problem involves producing multiple types of garments, which would require machine reconfiguration time.

Acknowledgment

The work is supported by the project "Russe ressearch university" financed by European Union-NextGenerationEU, through the National Recovery and Resilience Plan of the Republic of Bulgaria, project number BG-RRP-2.013-0001 and by the Polish-Bulgarian collaborative grant "Practical aspects for scientific computing". The work was partially supported by the Centre of Excellence in Informatics and ICT under the Grant No BG16RFPR002-1.014-0018, financed by the Research, Innovation and Digitalization for Smart Transformation Programme 2021-2027 and co-financed by the European Union.

References

[1] Xiong H., Shi S., Ren D., Hu J.,*A survey of job shop scheduling problem: The types and models*, Computers & Operations Research, Vol. 142,2022, 105731.

[2] Johnson S.M., *Optimal Two- and Three-stage Production Schedules With Setup Times Included*, Naval Res. Logist. Quart., Vol. 1(1), 1954, 61-68.

[3] Akers S.B., Friedman J., *A Non-Numerical Approach to Production Scheduling Problems*, J. Oper. Res. Soc. Am., Vol. 3(4), 1955, 429-442.
[4] Akers S.B., *A Graphical Approach to Production Scheduling Problems*, Oper. Res., Vol.4(2), 1956, 244-245.
[5] Sisson R.L., *Methods of Sequencing in Job Shops- A Review*, Oper. Res., Vol.7(1), 1959, 10-29.
[6] Jerald J., Asokan P., Saravanan R., Rani A.D.C., *Simultaneous scheduling of parts and automated guided vehicles in an FMS environment using adaptive genetic algorithm*, Int. J. Adv. Manuf. Technol., Vol.29(5–6), 2006, 584-589.
[7] Yin L., Li X., Gao L., Lu C., Zhang Z., *A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem*, Sustainable Comput. Inf. Syst., Vol.13, 2017, 15-30.
[8] Meng L., Zhang C., Shao X., Ren Y., *MILP models for energy-aware flexible job shop scheduling problem*, J. Cleaner Prod., Vol. 210, 2019, 710-723.
[9] Ham A., *Flexible job shop scheduling problem with parallel batch processing machine*, Proceedings of the 2016 Winter Simulation Conference, 2016, 2740-2749.
[10] Gao K.Z., Suganthan P.N., Pan Q.K., Tasgetiren M.P., A. Sadollah A., *Artificial bee colony algorithm for scheduling and rescheduling fuzzy flexible job shop problem with new job insertion*, Knowl.-Based Syst., Vol. 109, 2016, 1-16.
[11] Liu Q., Zhan M., Chekem F.O., Shao X., Ying B., Sutherland J.M., *A hybrid fruit fly algorithm for solving flexible job-shop scheduling to reduce manufacturing carbon footprint*, J. Cleaner Prod., Vol. 168, 2017, 668-678.
[12] Yang Y.Z., Gu X.S., *Pareto-based complete local search and combined timetabling for multi-objective job shop scheduling problem with no-wait constraint*, J. Donghua Univ. (English Ed.), Vol. 33(4), 2016, 601-624.
[13] Wang X., Wang B., Zhang X., Xia X., Q. Pan Q., *Two-objective robust job-shop scheduling with two problem-specific neighborhood structures*, Swarm Evol. Comput., Vol. 61, 2021, 100805.
[14] Li K., Deng Q., Zhang L., Fan Q., Gong G., Ding S.,*An effective MCTS-based algorithm for minimizing makespan in dynamic flexible job shop scheduling problem*, Comput. Ind. Eng., Vol. 155(2), 2021, 107211.
[15] An Y., Chen X., Li Y., Han Y., Zhang J., Shi H., *An improved non-dominated sorting biogeography-based optimization algorithm for the (hybrid) multi-objective flexible job-shop scheduling problem*, Appl. Soft Comput.,Vol. 99, 2021, 106869.
[16] Anghinolfi D., Paolucci M., Ronco R., *A bi-objective heuristic approach for green identical parallel machine scheduling*, Eur. J. Oper. Res., Vol. 289(2), 2021, 416-434.
[17] Grosch B., Kohne T., Weigold M,, *Multi-objective hybrid genetic algorithm for energy adaptive production scheduling in job shops*, Procedia CIRP, Vol. 98, 2021, 294-299.
[18] Scalia B., Leite C.F., The Job-Shop Scheduling Problem: Mixed-Integer Programming Models, TCS Archive, 2023.
[19] Bynum, M. L. et al., *Pyomo-optimization modeling in python*, Springer, 2021.
[20] Dauzère-Pérès S., Ding J., Shen L., Tamssaouet K., *The flexible job shop scheduling problem: A review*, European Journal of Operational Research, Vol. 314(2), 2024, 409-432.
[21] Dorigo M, Stutzle T., *Ant Colony Optimization*, MIT Press, 2004.
[22] Fidanova S. Hybrid Ant Colony Optimization Algorithm for Multiple Knapsack Problem. 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), IEEE, 2021, DOI:10.1109/ICRAIE51050.2020.9358351, 1-5.
[23] Bonabeau E., Dorigo M. and Theraulaz G., *Swarm Intelligence: From Natural to Artificial Systems*, New York,Oxford University Press, 1999.