TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics, Mechanics, and Engineering
Vol. 67, Issue 1II, September, 2024

IMPROVING EFFICIENCY IN MASS PRODUCTION: APPLYING GENETIC ALGORITHMS IN CONJUNCTION WITH OBJECTIVE FUNCTIONS

Mircea TEODORESCU, Florin LUNGU, Radu VLAD

Abstract: *In the dynamic context of the manufacturing industry, process optimization becomes essential for maintaining competitiveness. This article introduces an innovative methodology using genetic algorithms along with the objective function to address the specific challenges of mass production. By integrating the principles of genetic algorithms, we examine their capacity to model and optimize complex production processes, highlighting their applicability in the efficient design of assembly lines. Genetic algorithms are the ones that solve the objective function, offering a detailed perspective on how they can improve the accuracy of the final solutions. The case study presented demonstrates the practical application of this theoretical framework in a real production scenario, emphasizing significant improvements in efficiency and cost reduction. The results obtained validate the effectiveness of combining genetic algorithms with the Guinet objective function, proposing a promising direction for future research in the field of production optimization.*

Key words: *objective function, lot sizing, genetic algorithms*

1. INTRODUCTION

Genetic algorithms represent an efficient methodology for addressing complex optimization problems, thanks to their ability to explore large search spaces and identify solutions close to the global optimum, even in the case of complex and multidimensional objective functions. These algorithms use principles inspired by natural selection, such as selection, crossover, and mutation, to generate robust and adaptable solutions.

Following recent developments, several authors have attempted to complement the initial vision regarding genetic algorithms. Synthesizing the literature, we can conclude that a genetic algorithm is defined as an optimization process inspired by natural selection and genetic principles, which uses a population of individuals to explore and exploit the search space of a problem, with the aim of finding optimal or satisfactory solutions.

2. OBJECTIVE FUNCTION AND THE IMPORTANCE OF GENETIC ALGORITHMS

2.1. Defining the objective function

Genetic algorithms simulate the evolution of a population of candidate solutions (chromosomes) to optimize an objective function. These algorithms are effective in: scenarios where no polynomial-time algorithms exist; large search spaces, such as combinatorial problems; multi-objective problems, especially if multimodal or deceptive. [1]

Highlighting the importance of genetic algorithms and the objective function is crucial for adapting manufacturing systems to meet customer demands and maintain competitiveness. Genetic algorithms help navigate the challenges of increasing product variety, which often reduces productivity. [2]

Optimizing the objective function is vital for enhancing production processes, identifying optimal solutions, and assessing the performance of various candidates. In

Received: 30.04/03.07.24; **Similarities**: 20.08.24: **Reviewed**: 30.05./10.06.24: **Accepted**:22.08.24.

processing parameter optimization, it directly impacts production costs, productivity, and quality. The objective function must accurately reflect optimization goals and differentiate between solution qualities efficiently. The complexity and evaluation cost of the objective function are significant factors affecting the algorithm's efficiency. [3]

Various methods for modeling and optimizing the objective function are discussed in the literature. Gavrus et al. (2022) propose an efficient method based on the linearization of the objective function, eliminating the need for additional software. This approach utilizes linear mathematical programming despite the nonlinear nature of optimization models, particularly when integrating tool life into the model. [4] [5]

To address this challenge, the authors present an original mathematical procedure for analytically integrating tool life into the objective function. This method has been translated into a software tool, validating its effectiveness through comparisons with other optimization techniques. The results show that the proposed method is both simple and effective.

2.2. Introduction into genetic algorithms

Beginning with John Holland's pioneering work in the 1960s, genetic algorithms have evolved to tackle increasingly complex real-world challenges. By mimicking natural selection, these algorithms use operators such as selection, crossover, and mutation to generate optimized solutions, showcasing their flexibility and efficiency in solving difficult problems.

In "Adaptation in Natural and Artificial Systems" (1975), Holland defined genetic algorithms as a search and optimization process inspired by biological evolution. This concept emphasizes using natural selection and genetic mechanisms to evolve solutions to complex problems. [6]

David E. Goldberg, in "Genetic Algorithms in Search, Optimization, and Machine Learning" (1989), highlights the effectiveness of genetic algorithms in exploring large and complex search spaces. He explains how these algorithms simulate selection, crossover, and mutation processes to optimize and identify high-quality solutions efficiently. [7]

Melanie Mitchell's "An Introduction to Genetic Algorithms" (1996) offers an accessible definition, explaining genetic algorithms as inspired by biological evolution to solve problems by imitating selection, reproduction, crossover, and mutation. Mitchell emphasizes their adaptability and flexibility in addressing varied problems, including optimization, automation, and artificial intelligence. [8]

In the paper "A hybrid evolutionary algorithm for the job shop scheduling problem," an initial population of the genetic algorithm (GA) is generated with random solutions, followed by a local search process using a different heuristic function. [9]

Werner's 2011 study, "Genetic Algorithms for Shop Scheduling Problems: A Survey," explores the application of GAs in flexible manufacturing systems, focusing on minimizing total execution time and superficially analyzing works optimizing multiple criteria. [10]

In "A local search genetic algorithm for the job shop scheduling problem with intelligent agents," a genetic algorithm based on a local search strategy for optimal problem resolution is described. [11]

3. BASIC OPERATING PRINCIPLES OF GENETIC ALGORITHMS AND THEIR IMPORTANCE IN THE CONTEXT OF LOT SIZING

3.1. Basic operating principles

The description of the stages of a genetic algorithm follow the current path:

a) initialization. This stage involves creating an initial population of solutions. Each solution, or "individual," is typically represented as a string of characters, called a chromosome, which can be composed of bits, integers, or real values, depending on the specific problem. The initial population can be generated randomly or through more sophisticated methods that take into account preliminary knowledge about the problem space.

b) selection. Selection is the process by which individuals are chosen from the current population to generate the next generation.

Selection is based on the "fitness" of each individual, which indicates how well the solution represented by the individual solves the problem. Common methods of selection include roulette wheel selection, tournament selection, and rank-based selection.

c) crossover. In this stage, pairs of selected individuals are "crossed" to produce offspring, which inherit characteristics from both parents. Crossover can be simple (one crossover point), multiple (several crossover points), or uniform (choosing individual characteristics from each parent with equal probability). This process introduces variety into the solution population.

d) mutation. Mutation introduces random variations in the chromosomes of individuals, changing one or more genes. This mechanism is essential to maintain genetic diversity in the population and to help avoid local optima by exploring new regions of the search space.

e) replacement. In the final stage, it is decided which individuals will be kept to form the new generation. There are several replacement strategies, such as complete generation replacement (where all parents are replaced by descendants), elitist replacement (where the best individuals from previous generations are kept), or strategies based on age or fitness.

3.2. Lot sizing

In the context of production and manufacturing management, lot sizing refers to the decision-making process regarding the quantity of a product to be produced, ordered, or transported at once. This concept is crucial for balancing between inventory holding costs and setup or ordering costs. It's particularly relevant when discussing optimization techniques, such as those involving genetic algorithms, where the goal is to find the most cost-effective and efficient production plans. [12]

The challenge in lot sizing comes from the need to find an optimal balance. ***Holding costs*** are costs associated with storing inventory over a period, including warehousing, insurance, and depreciation. Larger lot sizes can lead to higher holding costs because more goods are stored before being sold or used in production.

Setup costs (in manufacturing) or ordering costs (in procurement) are costs incurred every time a production process is started or an order is placed. They can include expenses related to labor, machine setup, or administrative tasks for processing orders. Smaller lot sizes may reduce holding costs but can increase setup or ordering costs due to the more frequent switches in production or more frequent orders. [13]

Lot sizing techniques and models aim to determine the optimal lot size that minimizes the total cost, considering both holding and setup/ordering costs. Genetic algorithms can be particularly useful for solving complex, dynamic lot sizing problems that are difficult to solve with traditional optimization methods. They can efficiently handle multiple, conflicting objectives, such as minimizing costs while maximizing service level, and can adapt to various constraints and production environments. [14]

4. CASE STUDY

Identifying an optimization model for the production planning process necessarily involves collaboration between the researcher and a company where the proposed model can be implemented, so that the desired outcome of this research can be achieved.

High-performing enterprises are those whose decision-makers can easily and immediately adapt to new, changing conditions and identify viable opportunities and methods for improving the applied vectors, with perspective being one of the fundamental criteria for this purpose.

Foresight involves developing several probable scenarios of the future, taking into account past trends, with consideration for the hypothesis of crises or breaks in the equilibrium of the production activity.

Our partner in this research is a major manufacturer of metal subassemblies for the domestic electrical appliance industry (25 years of experience and a number of 75 qualified personnel) located in the municipality of Cluj Napoca, a company that has shown a total openness to the perspective of removing the causes of delivery time impacts to the client of the finished product.

Within the technological processes carried out within the company, several production centers need an improvement process for production scheduling, as described in the following table. At this moment, both the cutting stage (101L-Laser or 101P-Mechanical cutting) and the 102F-Bending stage are experiencing delays, therefore not all products are timely processed by the allocated center, leading to delays, blockages, and major inconveniences in delivering the final product to the company's clients.

Currently, there are 3 production centers that generate major delays in the production flow. Since these production centers are among the first ones a product goes through, these delays cause cascading delays in the production centers of the subsequent stages.

Table No. 1 shows the situation in week 40 of the year 2021. In column No. 3, which describes the situation in week 40 of 2021, some delays that have already occurred are highlighted. For example, the production center 102F - Bending has a deficit of 218 production hours (marked in red) theoretically needed to process all products requiring this operation, products introduced on the production line. As observed, this "bottleneck" generates chain delays, and the following production centers 102LS – Deburring, 102S-Metal Inserts, and 102A-Case Making will, in turn, generate delays. Any delay in the production flow can increase the risk that the entire batch of products being manufactured will be delivered to the client past the deadline. According to the delivery contracts, in this situation, the client would have the right to request delay penalties.

As a result of discussions with the company representatives, we analyzed the existing situation and concluded that it is necessary to address the problem systematically and in stages.

In the first stage, we will define an objective function aimed at minimizing production times.

In the second stage, it is necessary to analyze the results obtained, comparing them with the current situation in the company. The differences identified will lead to conclusions that will form the basis for changing the order of entry into production of various batches, an order that, at this moment, is determined by the unit manager, based on his experience. We have thus concluded that the most important decision is related to optimizing the splitting of orders that contain a large number of parts. The purpose of this algorithm is to find the optimal way of splitting them to achieve reductions in total production times.

As a result, we identified 4 different products whose technological processes differ, as can be seen in Table no 1.

Table no.1

index	1	2	3	4
	EFR 01	EFR 01	EFR 01	EFR 01
	Case 6 - Mini Case Eurotherm	Printed Board	Assembled Board	Assembled Board
101 P	1.8	0.9	1.3	1
102 F	1.5	1.2	2.6	1.1
102 LS	1.7	2.5	2.4	0
102 S	4.7	0.8	2	0
107 S	0	1.3	0	0
445 D	0.8	0.5	0.6	0.3
105 P	1.2	0.7	0.3	0.4
109 P	0	3	0	0

Upon analyzing the existing situation, we concluded that Guinet's mathematical model could be applied. [15] The customization was made based on the information available. As such, we arrived at the following objective function:

$$f = min \sum_{i=1}^{I} T_{i,M_p} \tag{1}$$

The constraints related to this objective function are:

$$T_{i,m_p} - T_{j,m_p} + KZ_{i,j,m_p} \geq d_{j,m_p} \quad \forall i,j \in [1,I] \;\; i \neq j \;\; \forall\, m_p \in [1,M_P] \tag{2}$$

$$T_{j,M_p} - T_{i,M_p} + K(1 - Z_{i,j,m_p}) \geq d_{i,m_p} \quad \forall i,j \in [1,I] \;\; i \neq j \;\; \forall\, m_p \in [1,M_P] \tag{3}$$

$$T_{i,m_p+1} - T_{i,m_p} \geq d_{i,m_p} \quad \forall i, \in [1,I] \quad \forall\, m_p \in [1,M_P - 1] \tag{4}$$

$$T_{i,1} \geq dint_i \qquad \forall i \in [1, I] \tag{5}$$

$$T_{i,M_P} \leq dl_i - d_{i,M_P} \; \forall \, i \in [1, I] \tag{6}$$

where:

$Z_{i,j,m}$

- is the variable with value 1 when operation i is scheduled before operation j on center m

$Z_{i,j,m}$

- is the variable with value 0 when operation j is scheduled before operation i on center m

$d_{i,Mp}$

- is the processing time of operation i on production center m and machine p

$d_{j,Mp}$

- is the processing time of operation j on production center m and machine p

$dint_i$

- is the moment from which order i can be scheduled

dl_i

- is the moment by which order i must be delivered

K - the constant that has a large value
M - The number of production centers
I,J - the number of orders
P - the number of machines in each production center
With value limits:
i,j ∈ [1 , 1500]
m ∈ [1 , 4]
p ∈ [1 , 8]

Constraints (2) and (3) prevent the overlapping in time of two orders i and j scheduled in the same production center m. Constraints (4) play a similar role, except that in this case, the overlaps of processing order i on machines m+1 and m are avoided. Constraints (5) and (6) ensure the assignment of values for the variables Ti,mp, so that the processing of orders does not start earlier than the entry moment into the company's records, nor later than the delivery deadline.

5. RESULTS AND FINDINGS

The algorithm is designed to address lot-sizing problems in the context of production planning and scheduling. These problems are crucial in manufacturing and production processes, focusing on determining the optimal production quantity for each product over a given time horizon to minimize costs while meeting demand and adhering to production constraints. The algorithm, through its various functions and optimization techniques like genetic algorithms, aims to optimize the scheduling of production plans by considering various constraints (e.g., machine availability, production capacity, and delivery deadlines) and objectives (e.g., minimizing production time). This approach is highly beneficial for efficiently managing resources and meeting customer demand in a timely manner.

The algorithm tackles lot-sizing problems in production planning by optimizing both the quantity of products to be produced and the scheduling of these production activities. It employs genetic algorithms and nonlinear programming to find solutions that minimize costs (setup costs) while adhering to constraints such as production capacity, demand fulfilment, and delivery deadlines. The detailed functions within the code prove a capability to handle complex constraints, including those nonlinear in nature, which is critical for accurately modelling real-world production scenarios.

The optimization algorithm for the objective function was executed using Matlab and was checked using specific options to ensure feasibility.

objective_.m

```
clear
clc
tic
global K I M dli dim dinti Anl;
M=11;
I=10;
K=10^6;
objective = @(x) sum(x(M:M:M*I));
ProductsTimeProduction=[0 1.8 1.5 1.7 4.7 0 0 0
0.8 1.2 0]';
dim=repmat(ProductsTimeProduction,[I,1]);
dli=2*max(max(dim(:,:)))*ones(I,1);
dinti=zeros(I,1);
for i=1:I
   v=0:dli(i)/M:dli(i);
x0((i-1)*M+1:(i)*M)=v(1:M);
end
x0=x0';
```

```
x0_=0:2*max(dim(1:end)):(I*M-
1)*2*max(dim(1:end));
x0=x0_';
lb = 0.01 * ones(I*M,1);
ub = 100.0 * ones(I*M,1);
disp(['Initial Objective: ' num2str(objective(x0))])
[A2,A3,A4,A5,b2,b3,b4,b5] =
AA(I,M,dim,dli,dinti);
A = full([A2;A3;0*A4;A5]);
b = [b2;b3;0*b4;b5];
Aeq = [];
beq = [];
nonlincon = @nlcts;
toc
x=ga(objective,I*M,A,b,[],[],lb,ub,[]);
disp(['Final Objective: ' num2str(objective(x))])
ProductsTimeProduction=[0 1.8 1.5 1.7 4.7 0 0 0
0.8 1.2 0]';
dim=repmat(ProductsTimeProduction,[I,1]);
dim_=repmat(ProductsTimeProduction',[I,1]);
disp('Solution')
x
xsmat=reshape(x,I,M);
GanttChart( xsmat',dim_' );
toc
```

This script sets up the objective function for the optimization problem, initializes various parameters, and use a genetic algorithm (ga) for optimization. The nlcts function is designed to compute nonlinear constraints for an optimization problem, incorporating a specific set of conditions that involve decision variables x, global variables (K, I, M, dim, Anl), and some complex indexing and matrix operations. The description of its operations and logic can be found in the initial setup:

· Global Variables: Declares global variables such as K, I, M, dli, dim, dinti, and Anl for use across different functions or scripts.

· Parameter Initialization: Sets the number of machines (M), items or projects (I), and a large constant K for use in constraint formulations. An objective function is defined to minimize the sum of certain variables within x, targeting efficient resource allocation or scheduling.

Initial Configurations

· ProductsTimeProduction and dim: Specifies initial production times for products and replicates these times across I entities, forming the dim matrix that represents task durations or requirements.

· dli and dinti Initialization: Sets up vectors representing specific limits (dli) and distances/intervals (dinti) between tasks, initialized based on dim.

Initial Guess

· Constructs an initial guess x0 for the optimization variables, forming a starting point for the GA to begin searching for an optimal solution. Two approaches for defining x0 are shown, with the latter being chosen for use.

Variable Bounds

· Defines lower (lb) and upper (ub) bounds for the optimization variables, constraining the solution space to realistic or feasible values.

Linear Constraints

· Calls the AA function to generate matrices (A2, A3, A4, A5) and vectors (b2, b3, b4, b5) representing linear constraints for the problem. These are assembled into a single matrix A and vector b for use with the GA, with some adjustments (e.g., 0*A4 and 0*b4) suggesting modifications or deactivations of certain constraints.

Nonlinear Constraints

· Sets nonlincon to a function handle @nlcts, pointing to a function that defines nonlinear constraints for the optimization problem.

Optimization

· Uses MATLAB's Genetic Algorithm function ga to search for an optimal solution x to the objective function under the given linear and (implicitly) nonlinear constraints, within the specified variable bounds.

Final Objective and Solution Reporting

· Displays the final objective value achieved with the optimized solution x.

· Replicates initial configurations for dim for further use or analysis.

· Prints the optimized solution x and reshapes it into a matrix xsmat for visualization or further processing.

Gantt Chart Visualization

· Calls a custom GanttChart function with the transposed xsmat and dim_ (a transposed version of ProductsTimeProduction replicated across I entities) to visualize the optimized schedule or resource allocation as a Gantt chart.

Code general details:

Structure Analysis:

- The code is structured into multiple functions, each handling a specific part of a larger optimization task. Functions like **A_** and **AA** sett up matrices and vectors for constraints, **cmbformat** we have use for combinatorial formatting, **GanttChart** for visualizing schedules, and **nlcon** and **nlcts** for handling nonlinear constraints. The **objective_.m** file define the objective function for optimization.

Optimization Goal:

- The ultimate goal is to optimizing a schedule or allocation, subject to various constraints (linear and nonlinear).

Key Operations:

- Matrix and vector operations for setting up the problem.
- Definition of constraint functions.
- Genetic algorithm (**ga**) for solving the optimization problem.

Algorithm Flow:

- Initialize matrices and constraints.
- Define objective function.
- Solve the optimization problem using a genetic algorithm.
- Visualize the results with a Gantt chart.

GanttChart function with the transposed xsmat and dim_ (a transposed version of ProductsTimeProduction replicated across I entities) to visualize the optimized schedule or resource allocation as a Gantt chart. This algorithm led to the acquisition of the objective function values for each individual product and Gantt charts for the four considered products. The products were labeled: 1 EFR 01, 2 EFR 01, 3 EFR 01, 4 EFR 01.

The value of the objective function for product 3 EFR 01

```
Initial Objective: 395.2
Elapsed time is 0.065904 seconds.
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.
Final Objective: 59.697
```

The value of the objective function for product 4 EFR 01

```
Initial Objective: 167.2
Elapsed time is 0.048123 seconds.
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.
Final Objective: 31.7457
```

Gantt Charts

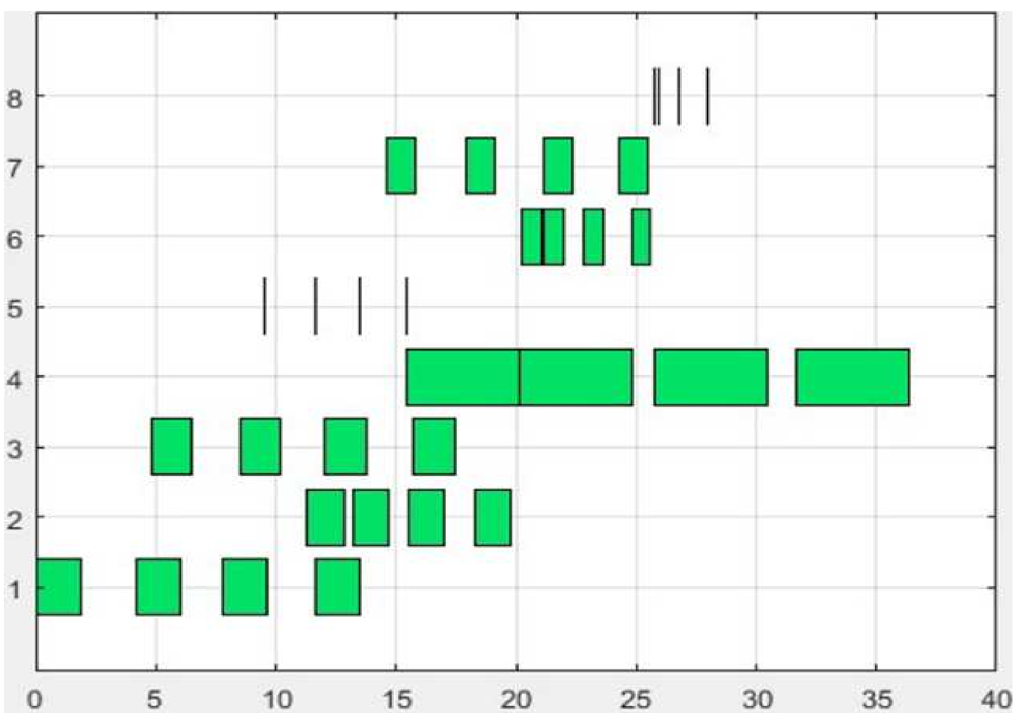

Fig. 1. Gantt Diagram 1 EFR 01

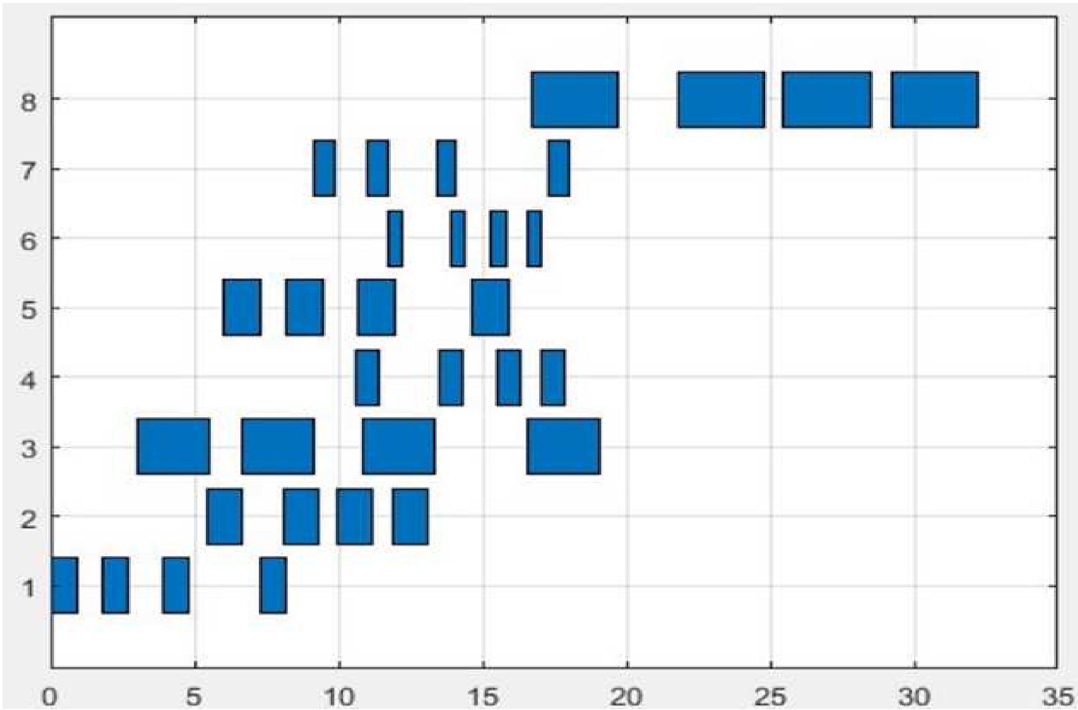

Fig. 2. Gantt Diagram 2 EFR 01

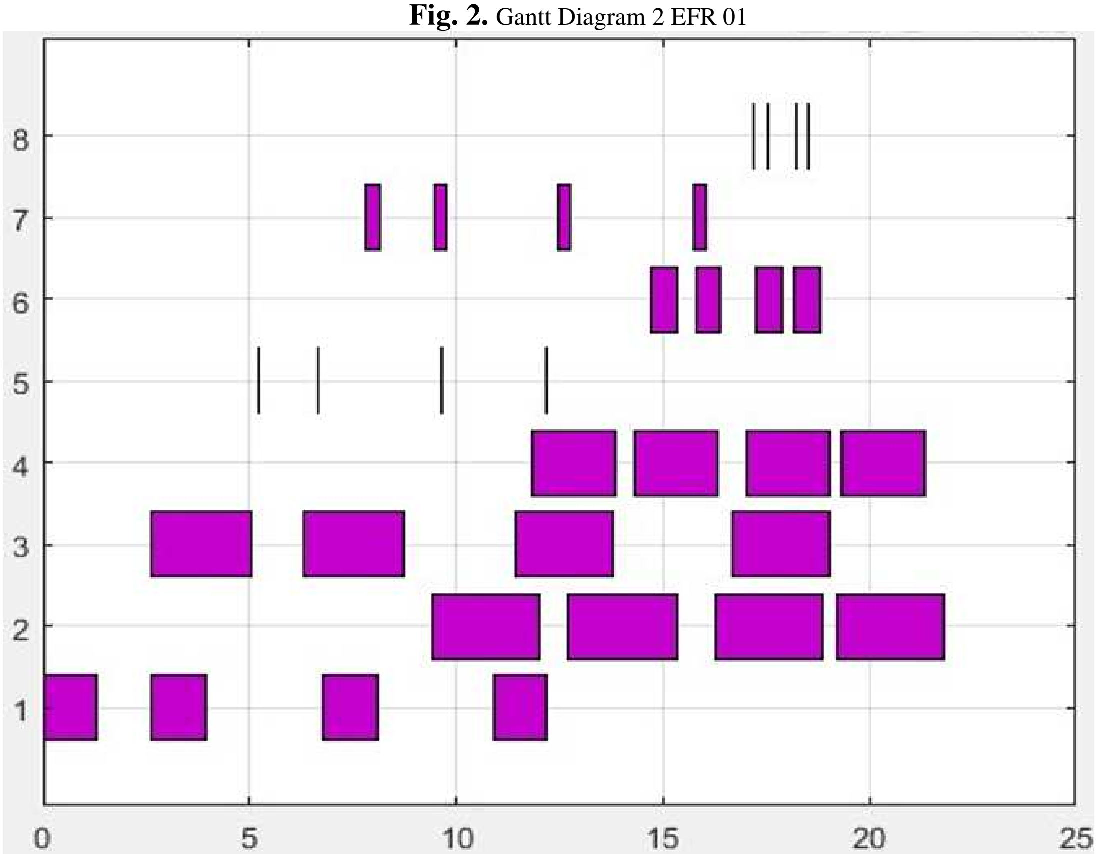

Fig. 3. Gantt Diagram 3 EFR 01

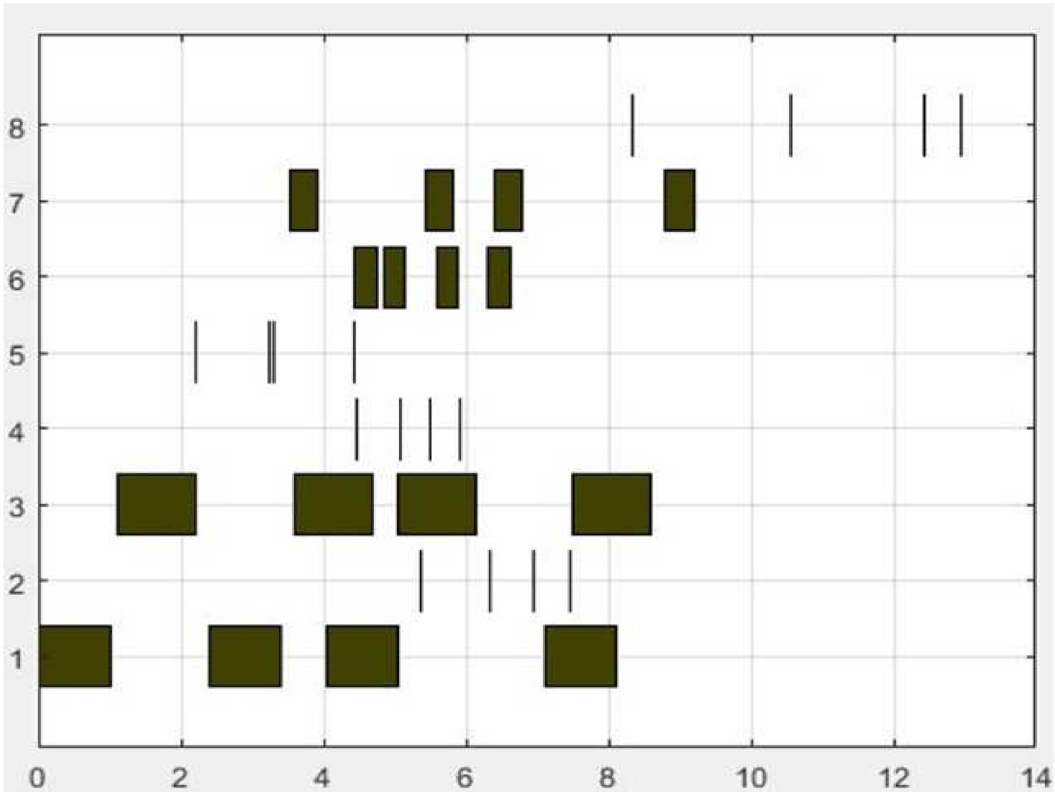

Fig. 4. Gantt Diagram 3 EFR 01

6. CONCLUSIONS

Correctly defining the objective function is fundamental to the successful application of genetic algorithms in production process optimization.

Genetic algorithms, inspired by natural processes of selection and genetic evolution, are metaheuristic search and optimization methods used to solve complex problems across a variety of fields, from engineering to artificial intelligence.

Incorporating genetic algorithms into lot sizing offers a flexible and powerful approach to navigating the complexities of production and inventory management, allowing for the exploration of a wide range of potential solutions and the identification of optimal or near-optimal lot sizes under varying conditions and constraints.

Through this approach, the algorithm can efficiently determine the optimal production lot sizing that meets demand in a cost-effective manner, crucial for effective production planning and operational efficiency.

7. REFERENCES

[1] Gen, M., Gheng, R., *Genetic Algorithms and engeneering optimization*, Wiley Series in Engeneering Design and Automation, Series Editor, 2000

[2] Oprea, M., Nicoara, S., *Inteligența arfiticială*, Universitatea Petrol-Gaze din Ploiesti, 2005

[3] Cristea C., *Scheduling optimization of flexible manufacturing system using genetic algorithms,* ACTA TECHNICA NAPOCENSIS Series: Applied Mathematics and Mechanics,Vol. 56, Issue I, March, pp.95-98, 2013

[4] Cristea C., *The allocation of customers to warehouses optimization using genetic algorithms,* ACTA TECHNICA NAPOCENSIS, Series: Applied Mathematics and Mechanics, Vol. 56, Issue I, March, pp. 89-93, 2013

[5] Gavrus C, Ivan N-V., Oancea G., *Machining Parameters Optimization Based on*

Objective, Function Linearization, Mathematics, 10, 803, 2022, available online at: https://doi.org/10.3390/math10050803
[6] Holland J.H., *Adaptation in Natural and Artificial Intelligence,* The MIT Press, Cambridge, Massachusetts, ISBN: 9780262581110, pp.8-12, 1992
[7] Goldberg D.E., *Genetic Algorithms – in Search, Optimization & Machine Learning*, Addison-Wesley Publishing Company, Reading (Massachusetts), ISBN 0-201-15767-5, p.2-3, 1989
[8] Mitchell M., *An Introduction to Genetic Algorithms,* The MIT Press, Cambridge, Massachusetts, ISBN: 9780262631853, p.2, 1996
[9] Zobolas, C.I., Trantilis, C.D., Ioannou, G.*, A hybrid evolutionary algorithm for the job shop scheduling problem*, Journal of the Operational Research Society, Vol. 60, No. 2, p. 221-235, 2009
[10] Werner, F. *Genetic Algorithms for Shop Scheduling Problems: A Survey*, Preprint, Vol. 11, p. 1-66, 2011.
[11] Asadzadeh L., *A local search genetic algorithm for the job shop scheduling problem with intelligent agents*, Computers & Industrial Engineering, ISSN 0360-8352, Vol. 85, pp. 376-383, July, 2015
[12] Fleischmann B., *The discrete lot-sizing and scheduling problem,* European Journal of Operational Research 44, 337-348 337 North-Holland, pp.336-348, 1990
[13] Clark A., Lobo B.A., Almeder C., *Lot sizing and scheduling: industrial extensions and research opportunities,* International Journal of Production Research 49:9, pp. 2457-2461, DOI: 10.1080/00207543.2010.532908 , 2011
[14] Degraeve Z., Jans R., *An industrial extension of the discrete lot-sizing and scheduling problem,* IIE Transactions, 36:1, 47-58, DOI: 10.1080/07408170490247296, 2010
[15] Vlad R.C., Marton A., Jucan D.C., Bogdan M., *Applying linear programming to flow shop scheduling,* Management and Economic Engineering, Vol. 14, No.4, pp.880-892, 2015

ÎMBUNĂTĂȚIREA EFICIENȚEI ÎN PRODUCȚIA DE MASĂ: APLICAREA ALGORITMILOR GENETICI ÎMPREUNĂ CU FUNCȚIILE OBIECTIV

Rezumat: *În contextul dinamic al industriei prelucrătoare, optimizarea proceselor devine esențială în vederea menținerii competitivității. Acest articol prezintă o metodologie inovatoare care utilizează algoritmi genetici împreună cu funcția obiectivă pentru a aborda provocările specifice producției. Prin integrarea principiilor algoritmilor genetici, a fost examinată capacitatea acestora de a modela și optimiza procese complexe de producție, evidențiind aplicabilitatea lor în proiectarea eficientă a liniilor de asamblare. Algoritmii genetici sunt cei care rezolvă funcția obiectiv, oferind o perspectivă detaliată asupra modului în care pot îmbunătăți acuratețea soluțiilor finale. Studiul de caz prezentat demonstrează aplicarea practică a acestui cadru teoretic într-un scenariu real de producție, punând accent pe obținerea de îmbunătățiri semnificative în eficiență și reducerea costurilor. Rezultatele obținute validează eficacitatea combinării algoritmilor genetici cu funcția obiectiv Guinet, propunând o direcție promițătoare pentru cercetările viitoare în domeniul optimizării producției.*

Mircea TEODORESCU, PhD. student, Eng**.,** Technical University of Cluj-Napoca**,** Technical University of Cluj-Napoca, Economical Engineering and Management Department, Blvd. Muncii, No.103-105, 400641, E-mail: mteodorescu@mail.utcluj.ro, phone: +40 728 85 89 84

Florin LUNGU, PhD., Eng., Professor, Technical University of Cluj-Napoca**,** Technical University of Cluj-Napoca, Economical Engineering and Management Department, Blvd. Muncii, No.103-105, 400641, E-mail: florin.lungu@mis.utcluj.ro

Radu VLAD, PhD., Eng., Associate Professor, Technical University of Cluj-Napoca**,** Technical University of Cluj-Napoca, Economical Engineering and Management Department, Blvd. Muncii, No.103-105, 400641, E-mail: florin.lungu@mis.utcluj.ro