

Article

Application Research of Soft Computing Based on Machine Learning Production Scheduling

Melinda Timea Fülöp ^{1,*} , Miklós Gubán ² , Ákos Gubán ² and Mihály Avornicului ²¹ Faculty of Economics and Business Administration, Babeş-Bolyai University, 400591 Cluj-Napoca, Romania² Faculty of Finance and Accountancy, Budapest Business School, 1149 Budapest, Hungary;

guban.miklos@uni-bge.hu (M.G.); guban.akos@uni-bge.hu (Á.G.);

avornicului.mihalyzilard@uni-bge.hu (M.A.)

* Correspondence: melinda.fulop@econ.ubbcluj.ro

Abstract: An efficient and flexible production system can contribute to production solutions. These advantages of flexibility and efficiency are a benefit for small series productions or for individual articles. The aim of this research was to produce a genetic production system schedule similar to the sustainable production scheduling problem of a discrete product assembly plant, with more heterogeneous production lines, and controlled by one-time orders. First, we present a detailed mathematical model of the system under investigation. Then, we present the IT for a solution based on a soft calculation method. In connection with this model, a computer application was created that analyzed various versions of the model with several practical problems. The applicability of the method was analyzed with software specifically developed for this algorithm and was demonstrated on a practical example. The model handles the different products within an order, as well as their different versions. These were also considered in the solution. The solution of this model is applicable in practice, and offers solutions to better optimize production and reduce the costs of production and logistics. The developed software can not only be used for flexible production lines, but also for other problems in the supply chain that can be employed more widely (such as the problem of delivery scheduling) to which the elements of this model can be applied.

Keywords: soft computing; genetic algorithms; product scheduling; heuristic methods



Citation: Fülöp, M.T.; Gubán, M.; Gubán, Á.; Avornicului, M.

Application Research of Soft Computing Based on Machine Learning Production Scheduling. *Processes* **2022**, *10*, 520. <https://doi.org/10.3390/pr10030520>

Academic Editor: Anna Trubetskaya

Received: 19 February 2022

Accepted: 3 March 2022

Published: 5 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In view of demographic change and high-cost pressure, ever greater efforts are being made to automate processes. On the one hand, there is an enormous combinatorial variety in process design. This includes dividing the tasks between available robots, determining the processing sequence, and adjusting the process settings. On the other hand, due to narrow and dynamic robot workspaces, motion planning proves to be extremely computationally intensive. The aim of this work was therefore to use heuristic algorithms to make processes more efficient to save costs and time. In this work, a new implementation of the genetic algorithm is used for process optimization.

Industrial production has been going through a major change for several years, also known as the fourth industrial revolution [1–3]. An essential part of this change is the complete penetration of processes with the help of virtual methods. At the same time, due to increasing complexity and decreasing batch sizes in production, there is an ever-increasing need for software for planning, automation, and optimization. To meet these increasing requirements, new methods and technologies have to be developed that enable optimized planning and control under different target values [4–7].

The sustainability-oriented production scheduling problem has been researched by many authors in the field, and there is ample literature on the topic [8–14]. However, there are circumstances which have led to changes in production processes, such as changes

in customers' habits in the past decade, especially in post-crisis times. In this article, we deal with the sustainable scheduling of generalizable systems with a structure similar to the production scheduling problem of a discrete product assembly plant with more heterogeneous production lines, and controlled by one-time orders. Some examples of such systems are digital server channels and parallel transport routes, multichannel IT, and so on. Herein, for simplicity, we will be dealing with sustainable production scheduling, and rely on the concepts used therein when modeling our system.

Several solutions have been applied to the scheduling problem. In addition to linearizing the task, linear programming methods and heuristic methods for the problem have been developed at the University of Miskolc. In practice, other queuing solutions have been used as well.

Several studies [8–12,14] have already shown that one of the best results of most heuristic methods of soft computing is the GA (genetic algorithm) for logistics systems. Following these studies, we chose this method. The advantages of this solution are that it is fast and accurate, especially in comparison to heuristic solutions. By setting the parameters of the objective function to the current production target, production can be quickly redesigned according to the current targets. The algorithm is very simple, and the runtime is better than for LP (linear programming) solutions. In addition, many aspects can be easily considered by the developed solution method. The model is more adaptable. In the daily, weekly, and monthly breakdown of the production organization, the schedule must be available during production. Thus, there is no need for real-time results. Experience has shown that, in the event of a production line failure, the production schedule can be easily rescheduled in a short time.

International competition is intensifying, and many companies are feeling the pressure to shorten their innovation cycles while at the same time further individualizing their customer approach. In order to continue to operate successfully on the market, innovative and intelligent solutions for comprehensive process optimization are required. Intelligent automation, also known as hyperautomation, offers a very good solution here [15–17].

Production planning and control are often characterized by laborious problem solving. A plan often has to be changed at short notice because of a lack of material or employees, or because a machine has broken down. As a rule, there is a lack of transparency about the impact of these adjustments on relevant key figures such as delivery reliability and productivity. Heuristic methods are used for the approximate solution of complex decision and optimization problems or associated optimization models. Opening methods construct a (first) feasible solution, while improvement methods lead to improved (locally optimal) solutions through successive solution transformation. Metastrategies drive improvement processes with a view to investigating promising solution areas and overcoming local optimality. The main contribution of this study is the development of a mathematical model for production scheduling, for which a new, modified GA solution is given. Software has been developed and tested in a practical environment. This can be integrated into a production management system to make the scheduling production processes on flexible production lines more efficient. During the solution, in addition to the appropriate optimal composition of the services, the shorter lead time also ensures fewer shifts, so the use of harmful substances used during the changeover is also less. In addition, less energy consumption helps sustainability, as the optimal solution leaves a smaller footprint.

In this article, we present a detailed mathematical model of the system under investigation. Then, we compile the IT for a solution based on a soft calculation method. We have created a computer application for this model, which we have run multiple times. Runtime results are also presented in this study.

2. Review of the Literature

In the relevant literature, there are many studies on how to improve the efficiency of production schedules. An optimum or close-to-optimum solution results from the use of the genetic algorithm. The genetic algorithm produces new individuals by crossing individual

pairs, whose positive properties may improve [18]. During selection, the individuals with the better properties remain. Thus, the solution gradually improves and approaches the optimum [12]. The efficiency of the genetic algorithm is greatly influenced by the applied crossover operator. Today, several crossovers have been developed. Crossbreeding simulates the analogy of genetic crossbreeding in nature. This is a procedure that generates a child entity from two selected parent entities [9]. They differ in various ways, but each is similar in that they either change just the order or just the position [19]. Mutation can do a lot to help eliminate the stagnation of the procedure, making mutations just as important as the crossover operator.

Other heuristic solutions to this problem are suggested in the literature, and the proposed method is also unique in that it can be effectively applied to rescheduling—even in the case of dynamic production scheduling—because it provides a very fast run. No such approach was found with GA [10,20].

Due to rapid technological development, automation has improved significantly in many areas of production. This leads to a complicated situation where decisions need to be made within a short timeframe and from a number of possible cases. Wadhwa and his colleagues suggest a flexible system, where n independent workplace products need to be processed on m machines, and each workplace has the same processing order on the machines [20]. It is important that the system finds jobs on the machines that minimize the make-span. The objective is achieved by the evolutionary heuristics of the genetic algorithm on the sustainable scheduling problem of the flow line. The advantage of the genetic algorithm is that it manages constraints and goals easily, thus facilitating the adaptation of the GA scheduler to a wide range of possible scheduling problems. The results of the study show that the implementation of the genetic algorithm is very effective compared to standard sequencing rules, such as a shorter processing time, the total processing time, and so on.

He and Hui [10] presented a heuristic genetic algorithm for the parallel one-step multipurpose scheduling (SMSP) of large units. First, they proposed a random search based on heuristic rules. By crawling through a set of random solutions, they obtained more feasible solutions. To improve the quality of the solutions, a genetic algorithm under heuristic rule was proposed. Because the run time of the genetic algorithm drastically increased due to certain limitations, a penal procedure was introduced. Thus, the proposed algorithm became effective, and can be used to deal with highly limited, large scheduling problems.

The authors studied the reactive sustainable scheduling method in a previous study [21]. In this previous study, a genetic algorithm-based reactive scheduling method was proposed. When dealing with aggregate production schedules, it is difficult to change the initial schedule due to unknown factors in the manufacturing system. In their research, they modified only a portion of the initial scheduling that sets the appropriate scheduling range.

The studies [10–12] on the field of the genetic algorithm also describe the network scheduling of a typical multipurpose batch plant. Multipurpose process scheduling is more difficult to handle than one-step or multistep process scheduling [14,19,20]. Most authors apply mathematical programming (MP) methods to solve this problem. However, these adjustments result in a very long calculation time. The genetic algorithm proposed by the authors selected a small part of the binary variables to encode into binary chromosomes, which is key to identifying the tasks. The genetic algorithm was first developed with a separate crossover to minimize the makespan and maximize the production.

The researchers of [8] developed a module based on genetic algorithm scheduling based on the priority rules (PRGA-Sched) module that provides shorter completion times in the production scheduling. The module was integrated into the Faborg-Sim simulation tool. Using production data from the Faborg-Sim PRGA Module, a heating boiler production system was analyzed and simulated using six products and orders from customers. Their results showed that a better finishing time and starting position can be achieved through the PRGA-Schedule module.

Dao and Marian [22] presented a genetic algorithm for the integrated optimization of precedent-limited production sequences and scheduling in several production line environments. This group of problems is an NP-hard combinatorial problem, requiring triple optimization: allocation of resources to each production line, production line layout, and production line scheduling. Due to the nature of the constraints, the length of the problem varies. To overcome this variability and the global optimum search, new resource allocation strategies, chromosome, crossover, and mutation were encoded.

Zhang [14] proposed a solution to reduce the pollutants generated through fabric production. First, a three-way model was drawn that included both the traditional delays and the environmental aspects. Then, he presented an innovative solution for the sustainable scheduling problem, namely a multi-objective genetic algorithm with a taboo-enhanced, iterated greedy local search strategy.

We noticed that the issue related to production programs has been approached from several perspectives, as is a complex optimization problem that has been analyzed from different perspectives by different researchers. For example, models such as backpack modelling [23–25], the PSO-GA hybrid algorithm [26–28], the tabu search algorithm [29–32], the improved cuckoo search (ICS) [33–35], Lagrangian heuristic algorithm and other heuristic algorithms [36–40], the mixed model [36–43], and so on, are other types of models used in the literature.

Based on the characteristics of these genetic algorithms, the scope is expanding, and research on the optimization and improvement of genetic algorithms is becoming more sophisticated. In recent years, production planning based on genetic algorithms has been analyzed in the literature using various software [44–54]. Our article contributes to the existing literature in that we did not use existing software. Instead, we developed software based on the genetic algorithm to solve this problem. We consider that this model is easy to adopt for different production problems, and can be customized by the software that we developed in comparison to the models using standard software.

3. Research Methodology

When investigating the problem, we concluded that, after the elaboration of the detailed mathematical model, the evolutionary algorithm was the most efficient solution to solving the problem associated with the model.

An exact solution may be given for the problem (as for all mathematical problems with a finite number of elements), but the number of steps is of a magnitude (appearing as an NP-problem) that precludes practical application, especially for the solution of real-time problems [55,56]. This is already a deterrent in terms of exact solutions. Our practical tests have already shown significant runtimes for small tasks, and the use of the trunk function during the solution presented a further problem (which we simplified in our sample tasks). To solve the problems emerging in practice, an efficient algorithm must be developed.

During our initial attempts, the exact solutions were discarded, and the heuristic methods were unreliable. The evolutionary algorithms were successfully applied to other problems. As such, for the next step in finding a solution, the choice of method seemed to be obvious. Implementing evolutionary algorithms is simple, and we can easily create applications for unique problems.

Evolutionary algorithms have already been used efficiently in practice for solving several problems [57]. Genetic and evolutionary algorithms gained significant popularity in the 1990s, and several studies have analyzed these methods [58–60]. In addition to its simplicity and ease of implementation, adequate modifications could provide even more efficient solutions.

4. Modelling

Investigating the problem reveals an optimization problem. In such cases, the mathematical modelling of the problem is the most convenient way of creating a computational model of it. When preparing the model, one should take care that the level of detail of

the model is optimally suited to the problem. All notations of the model are presented in Appendix A. The model does not have to be too detailed, because then it would be unmanageable, while a too broad model would produce inaccurate results [61].

4.1. The Tested System

To demonstrate the operation of the system we have developed, we first define what is meant by the problem of sustainable production scheduling of the order-driven discrete product assembly system with heterogeneous production lines. The discrete product system described above means that the product can be measured in pieces, and the production time of the same product n is n times the product, and its production cost is also n times of the product, and $n \in N^+$. The system is controlled by orders; that is, prior to the operation of the system, orders are known for a production (assembly) period of the plant. We know what kind and how many parts the orders must be produced. To maintain compatibility with other systems (e.g., packet data transfer), all identical products from the (virtual) orders will be sequentially produced on the same production line. Obviously, this should not be a requirement for a conventional schedule. Inhomogeneity of the production lines means that both the lead time and the production cost of a product depend on the product and the production line.

After clarifying our basic concepts, we formulated the task set. The plant had an $n \in N^+$ flexible production line and, for further reference, C_j ($1 \leq j \leq n$) designates a production line. The production line is characterized by its ability to manufacture every product. If this is not the case, then the product on this production line has a high lead time and cost, which will never be allocated. The plant can produce (assemble) different product types ($m \in N^+$) on these production lines. We did not deal with deadlines because we assumed that all acceptable solutions met the ordering deadlines.

Formally, it can be described as the following: $l \in N^+$ orders waiting for manufacturing be in the system.

$$S_{ij} : 1 \leq j \leq k_i \quad (1)$$

$$s = \sum_{i=1}^l \sum_{j=1}^{k_i} |S_{ij}|. \quad (2)$$

For example, $S_{ij} := \left\{ \begin{array}{c} e_k e_k e_k \dots e_k \\ 1 \ 2 \ 3 \dots |S_{ij}| \end{array} \right\}$. This kind of series cannot be divided into further parts. $\sigma(S_{ij}) = p$ is the number of series S_{ij} . It will be used the function $\tau(p) := j$ for further reference. The system model uses three essential matrices, all of which create a link between the product and production lines. One is the lead time of one type of product on one production line, and the other is the production cost of the product associated with the production line or the switch time between the products on the production line. These formally mean:

$\mathbf{P}_{n \times m} = [p_{ij}]$ is manufacturing cost of the product j on line i .

$\mathbf{T}_{n \times m} = [t_{ij}]$ is turnaround time of product j on line i .

$\mathbf{G}_{n \times s \times s} = [g_{i\tau(j)\tau(l)}]$, $\mathbf{D}_{n \times s \times s} = [d_{i\tau(j)\tau(l)}]$ is resetting time and cost from series j to series l on line i . ($1 \leq i \leq n$), ($1 \leq j \leq s$), ($1 \leq l \leq s$).

Our task was to find an optimal or close-to-optimal layout that complied with the set of conditions outlined below. As can be seen from the objective, a dual and counterpart parameter appear in the optimization targeting function of optimization. That is, if we are trying to minimize the cost, the best solution is to assign each series to its optimal production line. If this is achieved through a model in which all series designate a single production line as optimal, the lead time will be the maximum. In the same way, if it is possible to assign each sequence to a line with the optimum lead time, each series may be ordered at the highest cost line. Thus, it appears that the solution is somewhere between these two models.

For these cases, it is important to know the total leading time (T_j) and the manufacturing cost (K_j) of the production line j for an assignment X . For easier handling, suppose that $t_{i\emptyset j} := 0$, meaning before the first series is not reset.

An optimization task formulates the above system which will manufacture all series on a production line, then at least one following objectives are met:

1. Minimal lead time;
2. Minimal manufacturing cost;
3. The compromise objective function between time and cost.

In this paper, the third case will be examined, as the following objective function handles cases 1 and 2:

$$\lambda \cdot \max_j T_j + \mu \cdot \sum_j K_j \rightarrow \min \quad (3)$$

In the first case, $\mu = 0$, in the second case $\lambda = 0$.

4.2. The Mathematical Model

4.2.1. Constraints of the Model

This is the assignment hypermatrix (see Figure 1):

$$X_{s \times n \times s} = [x_{ijk}] \quad (4)$$

where this denotes:

$$x_{ijk} = 1 \quad (5)$$

that series i is assigned to manufacturing element k of production line j .

$$x_{ijk} \in \{0; 1\} \quad (1 \leq i \leq s; 1 \leq j \leq n; 1 \leq k \leq s) \quad (6)$$

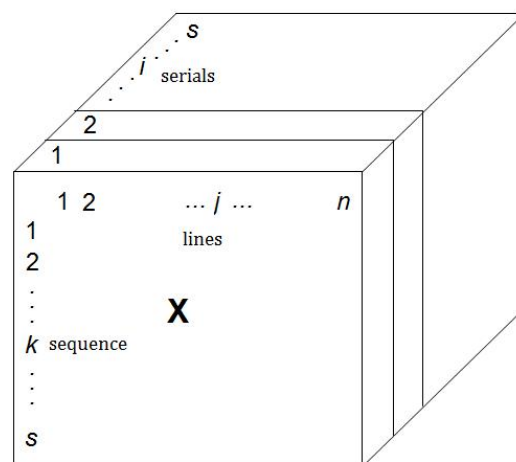


Figure 1. Hypermatrix of the assignment.

Each series is assigned to only one production line and only one manufacturing number:

$$\sum_{j=1}^n \sum_{k=1}^s x_{ijk} = 1 \quad (1 \leq i \leq s) \quad (7)$$

For each production line, and within a production series, a number is assigned up to one series:

$$\sum_{i=1}^s x_{ijk} \leq 1 \quad (1 \leq j \leq n, 1 \leq k \leq s) \quad (8)$$

The series is assigned to the first numbers of the production line. The following condition shows this:

$$\sum_{i=1}^s x_{ijk} - \sum_{i=1}^s x_{ij(k+1)} \geq 0 \quad (1 \leq j \leq n, 1 \leq k < s). \quad (9)$$

4.2.2. The Objective Function

Taking a step forward:

$$\text{trunc}(x) = \begin{cases} 1 & \text{if } x \geq 2 \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

The manufacturing cost of the production line j is:

$$K_{p_j}(\mathbf{X}) = \sum_{i=1}^s \sum_{k=1}^s p_{j\tau(i)} x_{ijk} \quad (11)$$

The resetting cost of the production line j is:

$$K_{t_j}(\mathbf{X}) = \sum_{i=1}^s \sum_{k=1}^{s-1} \sum_{r=1}^s d_{\tau(i)\tau(r)} \cdot \text{trunc}(x_{ijk} + x_{rj(k+1)}). \quad (12)$$

The total cost of the production line j is:

$$\begin{aligned} K_j(\mathbf{X}) &= K_{p_j}(\mathbf{X}) + K_{t_j}(\mathbf{X}) \\ &= \sum_{i=1}^s \sum_{k=1}^s p_{j\tau(i)} x_{ijk} + \sum_{i=1}^s \sum_{k=1}^{s-1} \sum_{r=1}^s d_{\tau(i)\tau(r)} \cdot \text{trunc}(x_{ijk} + x_{rj(k+1)}). \end{aligned} \quad (13)$$

The manufacturing time of the production line j is:

$$T_{p_j}(\mathbf{X}) = \sum_{i=1}^s \sum_{k=1}^s t_{j\tau(i)} x_{ijk}. \quad (14)$$

The reset time of the production line j is:

$$T_{t_j}(\mathbf{X}) = \sum_{i=1}^s \sum_{k=1}^{s-1} \sum_{r=1}^s g_{\tau(i)\tau(r)} \cdot \text{trunc}(x_{ijk} + x_{rj(k+1)}). \quad (15)$$

The total manufacturing time of the production line j is:

$$\begin{aligned} T_j(\mathbf{X}) &= T_{p_j}(\mathbf{X}) + T_{t_j}(\mathbf{X}) = \sum_{i=1}^s \sum_{k=1}^s t_{j\tau(i)} x_{ijk} + \\ &+ \sum_{i=1}^s \sum_{k=1}^{s-1} \sum_{r=1}^s g_{\tau(i)\tau(r)} \cdot \text{trunc}(x_{ijk} + x_{rj(k+1)}) \end{aligned} \quad (16)$$

The total objective function is:

$$\begin{aligned} f(\mathbf{X}) &= \lambda \cdot \max_{j \in \{1, \dots, n\}} T_j(\mathbf{X}) + \mu \cdot \sum_{j=1}^n K_j(\mathbf{X}) \\ &= \lambda \cdot \max_{j \in \{1, \dots, n\}} \left(\sum_{i=1}^s \sum_{k=1}^s t_{j\tau(i)} x_{ijk} + \sum_{i=1}^s \sum_{k=1}^{s-1} \sum_{r=1}^s g_{\tau(i)\tau(r)} \cdot \text{trunc}(x_{ijk} + x_{rj(k+1)}) \right) \\ &+ \mu \cdot \sum_{j=1}^n \left(\sum_{i=1}^s \sum_{k=1}^s p_{j\tau(i)} x_{ijk} + \sum_{i=1}^s \sum_{k=1}^{s-1} \sum_{r=1}^s d_{\tau(i)\tau(r)} \cdot \text{trunc}(x_{ijk} + x_{rj(k+1)}) \right) \rightarrow \min. \end{aligned} \quad (17)$$

The Complete Model:

$$x_{ijk} \in \{0; 1\} \quad (1 \leq i \leq s; 1 \leq j \leq n; 1 \leq k \leq s) \quad (18)$$

$$\sum_{j=1}^n \sum_{k=1}^s x_{ijk} = 1 \quad (1 \leq i \leq s)$$

$$\sum_{i=1}^s x_{ijk} \leq 1 \quad (1 \leq j \leq n, 1 \leq k \leq s)$$

$$\sum_{i=1}^s x_{ijk} - \sum_{i=1}^s x_{ij(k+1)} \geq 0 \quad (1 \leq j \leq n, 1 \leq k < s)$$

$$\begin{aligned}
f(\mathbf{X}) &= \lambda \cdot \max_{j \in \{1, \dots, n\}} T_j(\mathbf{X}) + \mu \cdot \sum_{j=1}^n K_j(\mathbf{X}) \\
&= \lambda \cdot \max_{j \in \{1, \dots, n\}} \left(\sum_{i=1}^s \sum_{k=1}^s t_{j\tau(i)} x_{ijk} + \sum_{i=1}^s \sum_{k=1}^{s-1} \sum_{r=1}^s g_{\tau(i)\tau(r)} \cdot \text{trunc}(x_{ijk} + x_{rj(k+1)}) \right) \\
&\quad + \mu \cdot \sum_{j=1}^n \left(\sum_{i=1}^s \sum_{k=1}^s p_{j\tau(i)} x_{ijk} + \sum_{i=1}^s \sum_{k=1}^{s-1} \sum_{r=1}^s d_{\tau(i)\tau(r)} \cdot \text{trunc}(x_{ijk} + x_{rj(k+1)}) \right) \rightarrow \min
\end{aligned}$$

5. Description of the Heuristic Approach

5.1. Our Previous Solutions and Examinations

The basic problem of optimization is the size of the solution space. If n is the number of production lines and l is the order number, then $D_{li} = \sum_{i=1}^l (l-i-1)! \frac{n!}{(n-i)!}$ is the solution space number. For example, 4 production lines and 50 orders will have a solution space of $1.3 \cdot 10^{65}$ elements. The optimal solution for such a large task is very difficult to ensure, so we have been looking for heuristic solutions for the last few years.

Our first reasoning was built on the greedy algorithm. The solution was based on the fact that the same products were merged and, depending on how the weight values in the target function were configured, the assignment was assigned to the production line with the shortest lead time or with the lowest cost line. Then, with manual tuning, we eliminated the outstanding assignments by smoothing the mergers. This kind of solution provided a usable result. However, the efficiency test found that we did not reach the right target for each order system. Our attention was directed to Ant Colony Optimization (ACO). The results of the completed software were subjected to statistical analysis, and we found that the deterministic sustainable production scheduling tasks realized with the ant algorithm resulted in very large variations. Thus, it was difficult to determine how close this would be to the optimal. However, the advantage of this solution was that it simultaneously handled two parameters—lead time and cost—so the one-on-one contact could be easily controlled.

As expected, with a higher run number and more agents, the average run result yielded a better target than a lower run number or fewer agents. However, the lowest aggregate target value was obtained from the lower run increments. On the contrary, the average target value was close to 1.5% lower for the higher runner step [4]. It was clear that the process did not converge, as can be seen in the examples. Only in the case of a high number of agents and a long run, or an evaporation factor proportional to the maximum path length, could a relatively low target value be found. A detailed examination was unnecessary since, in the examined samples (about 1000 runs), we found that the solution obtained could be very different at the same parameter value. Thus, it was found that usage of the ant algorithm primarily provided good ground for pre-processing other processes. Therefore, we will use the more efficient and verifiable convergent GA solution to play a role in pre-processing [6]. Accordingly, the solutions provided by the ant algorithm form the initial population of the GA. This is positive from the point of view of good-quality initial chromosomes, and, from experience, it is also beneficial that each individual is relatively distant from the other.

5.2. The Structure of the Chromosome

The method is based on a well-designed chromosome. If it is possible to determine the chromosome together with its locus of limbs, such that we can efficiently apply the GA operators to the genes on it, then we can develop a good and usable method. First, we determine the chromosome whose loci number is:

$$g = 2 \cdot s$$

We assign an integer number to every series of the original problem with function (see Table 1) $\sigma(S_{ij}) = p$.

Table 1. $\sigma(i)$ Function.

$\sigma(S_{ij})$	1	2	...	k_1	$k_1 + 1$...	$k_1 + k_2$...	s
S_{ij}	S_{11}	S_{12}	...	S_{1k_1}	S_{22}	...	S_{2k_2}	...	S_{lk_l}

The chromosome will then be structured as follows (see Table 2).

Table 2. Structure of a chromosome.

1	2	...	s
L_{1j_1} N_{1j_1}	L_{2j_2} N_{2j_2}	...	L_{lj_l} N_{lj_l}

Denote

L_{ij_i} : production line j_i is assigned to series i .

N_{ij_i} : production sequence number of series i on line j_i (production sequence). It is not important that the strict sequence $(1, 2, \dots, k_i)$ is followed; only the order of the values determines the sequence of series on the line. If two series have the same values, then the series with a lower sequence number will be manufactured first. This consideration will accelerate the algorithm (every result will be a possible result). For example Table 3.

Table 3. An example of a chromosome.

1	2	3	4	5
0 10	1 4	1 3	0 2	1 4

In Table 3, two lines (0 and 1) are used with 5 locusts of a chromosome. The first gray block shows that series 1 is assigned to line 0.

The series 1 has sequential number 10 and the series 4 has sequential number 2 on line 0. Series 2 has got sequential number 4, series 3 has got sequential number 3, and series 5 has got sequential number 4 on line 1. The allocation will be as follows Table 4.

Table 4. Example of production lines.

Line 0	Series 4	Series 1	
Line 1	Series 3	Series 2	Series 5

The order number and the order series number can be decoded from Table 1. The type of product and the number of an order can be decoded from these numbers with function $\tau(S_{ij})$. The number of an order is assigned to series S_{ij} . Resetting cost and leading time can be easily determined from these numbers.

(a) Crossover

P denotes the number of elements in the population.

The first operator of GA is the crossover that creates two new chromosomes by means of any two chromosomes from the population. That is, two new possible solutions are possible in our assignment from two possible assignments. During the crossing, the genes on a given loci interval of one of the parent chromosomes, in which rows are strung to the loci, are exchanged with the genes of this locus domain of the other parent genome, and vice versa. The crossing is based on the starting point and the length of the chromosomes. The procedure is outlined by the following algorithm.

KER be a random integer between 1 and $\frac{P}{2}$ (half of the population's number). The KER represents the number of crosses applied to a particular population.

Select randomly two different individuals from the current population.

Select at random a POZ crossing point (between 1 and 2) and a HOSSZ value (between 1 and 2—of POZ). The POZ shows from which loci to begin the crossing, and the HOSSZ shows the number of those loci on which genes are to be exchanged.

Create two new chromosomes by replacing the genes of the two parents from POZ length to HOSSZ length. See the above example.

Perform the second and third steps KER times.

Consider the following two chromosomes (Table 5).

Table 5. Chromosome of parent entities.

1		2		3		4		5	
0	10	1	4	1	3	0	2	1	4
1	7	1	3	0	1	0	4	0	2

Take the starting point 2 and the length 6 (Table 6).

Table 6. Chromosome of successive entities.

1		2		3		4		5	
0	10	1	3	0	1	0	4	1	4
1	7	1	4	1	3	0	2	0	2

Note that our crossing solution is flexible. As it is not necessary to replace blocks, it is also possible to replace subblocks. This means that the length can not only be an odd value, but also an even value, such as 5. In the case of the fourth series, the value determining the production order does not change, only the number of the production line. Based on the above, this still provides a possible solution for all crossings.

(b) Mutation

The role of mutation is to include in the population chromosomes possible solutions that would never enter the optimization through initial design and crossing. Mutations always change the gene of a locus. This, in our case, can be a production line gene, but it can also be a production sequence gene.

MUT is a random integer between 1 and $\frac{p}{2}$ (half of the population's number). This value determines the number of chromosomes that will be allowed mutation in our current population.

Select a random MUT number of individuals from the population.

Take the first selected individual.

Select a locus from this individual randomly. If this is an even-value gene, then generate a value between 1 and s. Change the value of the selected gene to the value obtained.

If it is odd, generate a value between 0 and n – 1 (production number 1) and substitute it for the gene's existing value.

After this, select the next individual and perform the steps from step 3 until the mutation is performed for all selected individuals.

(c) Selection

The role of selection is, in addition to the new individuals that have been crossed and mutated, for the somewhat valuable individuals of the present population to be included in the new population (generation) and eventually aid in optimization (the list is in descending order). In our method, we combine the following two selections:

Proportional fitness selection;

Elite list selection.

$p + \left\lceil \frac{p}{2} \right\rceil$ be the number of genetically generated entities. The number of all entities is $q = 2p + \left\lceil \frac{p}{2} \right\rceil$. Find the best fit for fitness (f). If there is more than one, choose one at random and add this to the new population. $p - 1$ is selected from the $q - 1$ chromosome. Determine the fitness of each individual. This value is weighted using the lead time and total production cost assigned to the solution (evaluation).

5.3. Convergence of the Process

The above procedure converges. Convergence is ensured by the fact that fitness values are limited from the bottom (the value cannot be negative) and because of elite list selection, where at least one of the best individuals of the previous population will be included in the population. Thus, the fitness value for a new population with the smallest fitness value cannot be higher than the best fit in the previous population. Therefore, the smallest fitness value of the population is a monotonous downward series. Thus, it will be a convergent series. The question is whether this series converges to the optimum value. The convergence of the genetic algorithm to optimum is provided in the study [3].

6. Discussions and Results

We have developed a simple application for our investigation, which functions according to the above. During the visualization, we strived for simplicity, since our objective consisted of the adequacy of the model and the applicability of the method.

The examined sample task is presented below. The structure of the sample task corresponds to the outlines of the model. This sample task is from a real company, but it was simplified for reasons of transparency. In some cases, we significantly deviated from the actual numbers to demonstrate the functionality of our developed method.

The task is presented below using the elements of the software setup window.

The Structure of the Sample Task:

The company had four production lines, and our goal was to schedule four production lines (channel).

Our aim was to produce six different products on these production lines.

The number of orders was 40.

The table of the specific orders shows how many units of a specific product had to be produced. The figure shows the products related to the first 16 orders (order) and their number of units.

$C\#i$ shows, for the production line i (channel), the turnaround time necessary to produce the product j after the product i (T_i).

The turn-around timetable presents the lead time of specific products (T_i) on the specified production line (C_j).

The cost table shows the production costs of the specific products (T_i) on the specified production line (C_j).

Time is measured in minutes and the cost is measured in euros. In this case, we can dispense the currency and the unit of time, as these do not influence the system (Figure 2).

C#1	T1	T2	T3	T4	T5	T6
T1	0	6	6	7	6	5
T2	6	0	6	7	6	7
T3	5	6	0	6	7	5
T4	5	5	5	0	5	6
T5	5	5	6	7	0	7
T6	6	5	5	7	5	0

Turnaround	C1	C2	C3	C4
T1	81	85	74	89
T2	70	89	71	72
T3	78	87	73	75
T4	90	69	73	87
T5	75	92	74	87
T6	73	92	88	73

Costs	C1	C2	C3	C4
T1	16	18	19	20
T2	21	21	23	21
T3	18	20	21	22
T4	18	20	19	22
T5	21	16	21	17
T6	18	18	18	17

Orders	Product	No of Serial
1	3	32
2	1	45
3	5	27
4	4	39
5	6	36
6	2	37
7	5	26
8	1	11
9	5	5
10	4	29
11	4	16
12	4	21
13	4	21
14	7	7
15	5	44
16	1	45

Figure 2. The data of the production scheduling problem (C#i in the computer application means the selected C4 production line).

Our aim was to place the elements in the orders table on the four production lines in such a manner that it was optimal from a perspective of (the list is in descending order):

Lead time;

Minimal costs.

This can be controlled through the alpha and beta parameters shown in Figure 3. If $\alpha = 1$ and $\beta = 0$, then we optimized for lead time. If $\alpha = 0$ and $\beta = 1$, we optimized for costs. The two criteria were considered in weighted form for all subsequent results.

Under the set conditions, along with the two aforementioned cases, we also presented a case in which both criteria were taken into account with different weights.

The run included 200 steps for each investigation. The run results from the following figures had the chromosomes obtained during the selection, with the individual lines representing the chromosomes. The fields for setting the two parameters can be seen in the upper part of Figure 3.

Genetic Algorithm

OK

GA

Alpha

Beta

Lépésszám

Population

1

1

200

15954.6	1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40																																								
23887	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23892	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23906	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23908	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23911	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23911	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	16	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23913	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23913	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23913	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23915	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23915	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	16	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23917	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23917	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23917	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	16	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23917	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23917	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23917	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23917	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23917	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23917	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23917	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23917	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0	39	3	26	3	37	0	4	0	21	3	39	2	16	3	17	2	14	0	36	0	27	1	12	3	18
23917	1	2	7	3	3	3	3	1	18	3	20	3	12	3	39	2	38	1	39	1	19	2	7	1	28	2	36	3	37	0	14	2	3	0	9	3	22	3	35	1	35	1	19	0	6	0	4	3	12	3	26	3	1	3	1	0</																									

Depending on the change in parameters, we investigated the result provided by the method. In the case of examined samples, when turnaround time was optimized, the turnaround time could not be less than 16,163 time units, and the cost could not be less than 18,494 cost units.

Figure 4 shows a run result in a case where the lead time parameter $\alpha = 1$, the figure shows that the system seeks equal load. The figure can be interpreted as follows.

Each order is associated with a color. The quantities to be produced for each order are shown by the white numbers. The gray gaps represent inactive times. The breadth of the individual orders symbolize the necessary production time (the broader the order, the longer the production time), and the color yellow represents the production lines (of which there are currently four).



Figure 4. Uniform allocation in case of parameter $\alpha = 1$.

If the cost is considered for the purpose of optimization, the 18,189.75 time units cannot amount to less than 16,273 units of expenditure. The example is examined for the expenditure case in which the parameter of the expenditure (beta) is 1 and the lead time (alpha) is 0 (Figure 5).

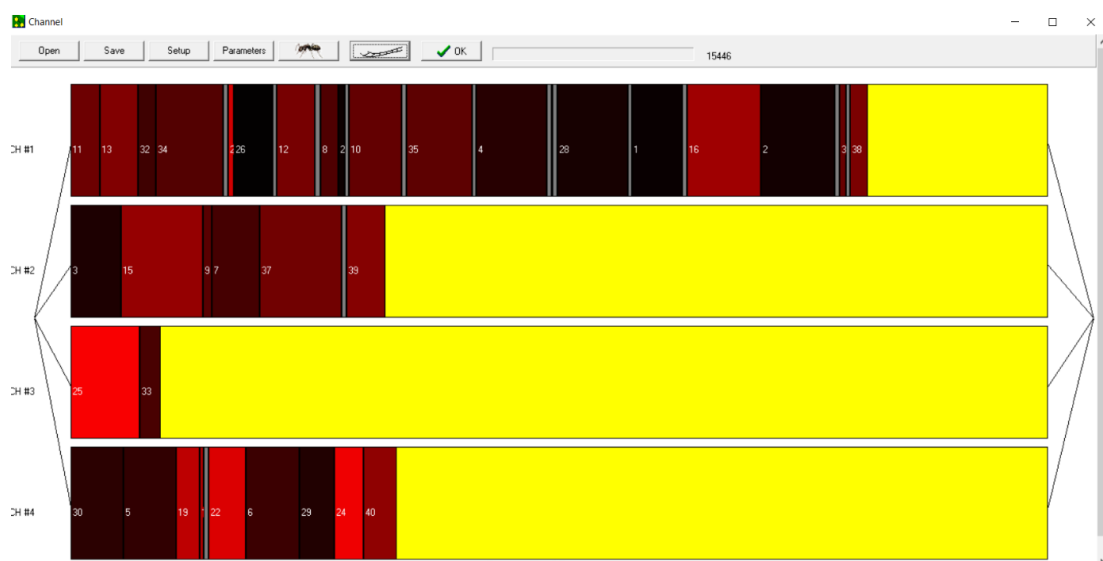


Figure 5. Allocation for high-cost production lines (3 lines).

Based on the description of the sample task, the third production line was among one of the most expensive lines, from the perspective of the products to be manufactured. The solution shows that the assembly line that was expensive relative to the ordered products was only minimally used by the system. This can be seen on the third production line.

In practice, the joint investigation of the two criteria can be conducted by considering the experience. In our examined example, the experiential result was as follows (Figure 6).

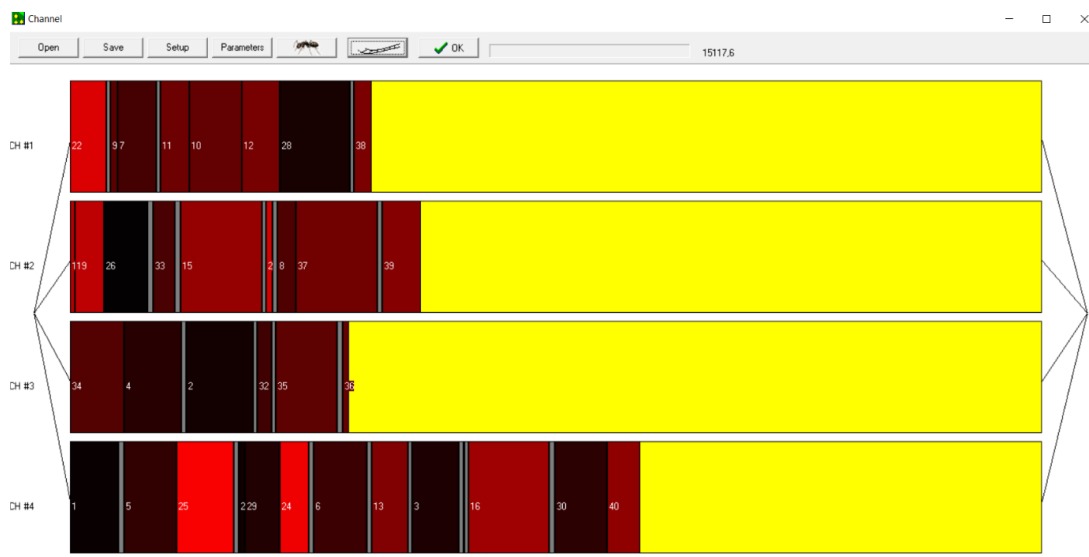


Figure 6. The result of the allocation for the parameters $\alpha = 0.6$ and $\beta = 0.4$.

This was in line with the practical experience that most orders were allocated to the fourth line.

7. Efficiency of the Algorithm

In the examined sample tasks, the program found the near-optimal solution in a maximum of 200 steps.

The run times of the individual cases did not exceed 30 s.

In general, based on experience to date, the program performs thousands of cycles in the optimal search for the most complex task. The running time for complex tasks was no more than 5 min. The program does not require special hardware.

8. Conclusions

8.1. Theoretical and Practical Implication

In this research, we presented the results of the objectives set out in the introduction. The first goal was to create an exact unique mathematical model for the task, so that the GA's IT of the task could be fitted onto this model.

What caused the unreliability of previous heuristic solutions? Studies have shown that, due to the robustness of such tasks, only in certain cases is the solution's optimum closeness of the solution ensured. In some cases, the solutions differed very much from the actual—that is, from the optimum obtained by us through the analysis of a well-analyzed test system—since the methods were largely based on the conditions of an average system. Unfortunately, the ant colony optimization was different in behavior. This solution had the great advantage of avoiding the alpha and beta parameters of the current method. The method itself tried to adjust to optimize costs and lead time together, and that is what caused its failure. As demonstrated by the combined effect of the two objectives, we did not obtain targets changing in the same way as the two parameters analyzed with the methods we employed, so the system could not produce good solutions. Sadly, through its use, we obtained possible solutions but not an optimal solution.

The third objective was to provide a GA model that would fit the task. Basic models can be adapted with little modification, since, in our case, the structure of the chromosomes and the genes to be replaced by the locus used were special in their structure. Genes contain two different elements, so genetic operators must adapt to this. The unique chromosomes and operators created fitted the mathematical model.

In this research, algorithms for generating time-optimized solutions were implemented in a software program. For this purpose, all necessary steps were presented, starting with a probabilistic movement planning algorithm with fast and accurate collision detection, a heuristic solver for process flow optimization and load balancing, up to a practical validation of the methodology. Due to the high complexity, a robust process automation or optimization with common planning and technical methods can only be implemented to a limited extent. Mathematical analyzes and computer-aided methods can make additional contributions to greater robustness and increased efficiency. Time-optimized programs for process optimization are supplied as a practical application. For this, it is necessary to integrate the methods into the production network. The evaluation of the results is based on precise simulations with the developed software. In this paper, we demonstrated that order-driven production can be mathematically described, and be assigned to a well-managed model. The model condition criteria were simple, but the complexity of the target function required that the task related to the model was not solved by using the exact method. Compared to previous studies, we have now solved the problem with a variation of the genetic algorithm. This solution method provided the best results compared to our previous experiments.

In the past, we performed numerous runs and evaluated the results using statistical methods. These findings also supported our hypothesis that soft calculation methods are well suited for production tasks as well.

8.2. Limitations and Future Research

For future research, it will be important to investigate the scalability of the solver so that the entire production line can be optimized, instead of a single process. This is particularly interesting for load balancing because the number of tasks is much larger. A promising approach is to first build a population of good and diverse solutions using heuristics, and then apply artificial intelligence methods to identify a common pattern among them according to the pattern that frequently occurs.

Integer values were four-byte integers, so these were not a limitation in the practical problem. Variables were dynamic, so the hardware set the limits. The results of the tests so far, which were used in a medium-sized production plant, were satisfactory. Very large serial numbers have not been published, but based on the experience so far, the previous day's production plan is prepared quickly—well within deadlines.

Furthermore, in the future, applications are to be examined in which robot programs are created solely from the specifications of product planning. Reinforcement learning can be used for this in order to train a model from the processes. In this way, all relevant process technology requirements and tolerances can be recorded.

Author Contributions: Conceptualization: Á.G., M.G. and M.T.F.; methodology Á.G. and M.G.; writing—original draft preparation, Á.G., M.G., M.T.F. and M.A.; writing—review and editing: M.G. and M.T.F.; visualization M.G. and M.A.; supervision M.T.F.; project administration M.T.F. and M.A.; funding acquisition M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Budapest Business School Research Fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research was carried out with the framework of the Centre of Excellence for Future Value Chains of Budapest Business School.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Notation	Range	Explanation
$n \in N^+$		the number of flexible production lines
$m \in N^+$		the number of different product types
$l \in N^+$		the number of orders waiting for manufacturing
i		the series number of the order
j		the series number of the production line
k		the manufacturing element
k_i		the number of the series of order i
C_j	$(1 \leq j \leq n)$	ID of production line j
S_{ij}	$S_{ij} : 1 \leq j \leq k_i$	the series j of the order i .
s		the total number of series of all orders
$\sigma(S_{ij}) = p$		the number of the series S_{ij}
$\tau(p) := j$		
$\mathbf{P}_{n \times m} = [p_{ij}]$		the manufacturing cost of the product j on line i .
$\mathbf{T}_{n \times m} = [t_{ij}]$		the turnaround time of product j on line i .
$\mathbf{G}_{n \times s \times s}$ $= [g_{i\tau(j)\tau(l)}],$ $\mathbf{D}_{n \times s \times s}$ $= [d_{i\tau(j)\tau(l)}]$	$(1 \leq i \leq n), (1 \leq j \leq s),$ $(1 \leq l \leq s)$	the resetting time and cost from series j to serial l on line i .
$\mathbf{X}_{s \times n \times s} = [x_{ijk}]$	$x_{ijk} \in \{0;1\}$ $\left(\begin{array}{l} 1 \leq i \leq s; 1 \leq j \leq n; \\ 1 \leq k \leq s \end{array} \right)$	$x_{ijk} = 1$ that series i is assigned to manufacturing element k of production line j .
$K_{p_j}(\mathbf{X})$		The manufacturing cost of the production line j
$K_{t_j}(\mathbf{X})$		The resetting cost of the production line j
$K_j(\mathbf{X})$		The total cost of the production line j
$T_{p_j}(\mathbf{X})$		The manufacturing time of the production line j
$T_{t_j}(\mathbf{X})$		The reset time of the production line j
$T_j(\mathbf{X})$		The total manufacturing time of the production line j
$f(\mathbf{X})$		The total objective function
$\lambda =$ alpha (α) in the software	$\left[\frac{1}{min} \right]$	Normalizing and at the same time weight factor Lead time parameter
$\mu =$ alpha (β) in the software	$\left[\frac{1}{Euro} \right]$	Normalizing and at the same time weight factor Cost parameter

References

1. Zambon, I.; Cecchini, M.; Egidi, G.; Saporito, M.G.; Colantoni, A. Revolution 4.0: Industry vs. Agriculture in a Future Development for SMEs. *Processes* **2019**, *7*, 36. [\[CrossRef\]](#)
2. Salah, B.; Khan, S.; Ramadan, M.; Gjeldum, N. Integrating the Concept of Industry 4.0 by Teaching Methodology in Industrial Engineering Curriculum. *Processes* **2020**, *8*, 1007. [\[CrossRef\]](#)
3. Sevinç, A.; Gür, Ş.; Eren, T. Analysis of the Difficulties of SMEs in Industry 4.0 Applications by Analytical Hierarchy Process and Analytical Network Process. *Processes* **2018**, *6*, 264. [\[CrossRef\]](#)
4. Liu, Y.; Dong, H.; Wang, S.; Lan, M.; Zeng, M.; Zhang, S.; Yang, M.; Yin, S. An Optimization Approach Considering User Utility for the PV-Storage Charging Station Planning Process. *Processes* **2020**, *8*, 83. [\[CrossRef\]](#)
5. Szentesi, S.; Illés, B.; Cservedák, Á.; Skapinyecz, R.; Tamás, P. Multi-Level Optimization Process for Rationalizing the Distribution Logistics Process of Companies Selling Dietary Supplements. *Processes* **2021**, *9*, 1480. [\[CrossRef\]](#)
6. Liu, W.; Luo, F.; Liu, Y.; Ding, W. Optimal Siting and Sizing of Distributed Generation Based on Improved Nondominated Sorting Genetic Algorithm II. *Processes* **2019**, *7*, 955. [\[CrossRef\]](#)
7. Ehyaei, M.A.; Ahmadi, A.; Rosen, M.A.; Davarpanah, A. Thermodynamic Optimization of a Geothermal Power Plant with a Genetic Algorithm in Two Stages. *Processes* **2020**, *8*, 1277. [\[CrossRef\]](#)
8. Aydemir, E.; Koruca, H.I. A New Production Scheduling Module Using Priority-Rule Based Genetic Algorithm. *Int. J. Simul. Model.* **2015**, *14*, 450–462. [\[CrossRef\]](#)
9. Akram, U.; Fülöp, M.T.; Tiron-Tudor, A.; Topor, D.I.; Căpușneanu, S. Impact of Digitalization on Customers' Well-Being in the Pandemic Period: Challenges and Opportunities for the Retail Industry. *Int. J. Environ. Res. Public Health* **2021**, *18*, 7533. [\[CrossRef\]](#)
10. He, Y.; Hui, C.-W. A binary coding genetic algorithm for multi-purpose process scheduling: A case study. *Chem. Eng. Sci.* **2010**, *65*, 4816–4828. [\[CrossRef\]](#)
11. Fülöp, M.T.; Szora Tamas, A.; Ivan, O.R.; Solovăstur, A.N. Regressive model regarding the necessary profit margin forecast for a new project in the construction field. *Econ. Comput. Econ. Cybern. Stud. Res.* **2020**, *54*, 181–198. [\[CrossRef\]](#)
12. Salido, M.A.; Escamilla, J.; Giret, A.; Barber, F. A genetic algorithm for energy-efficiency in job-shop scheduling. *Int. J. Adv. Manuf. Technol.* **2015**, *85*, 1303–1314. [\[CrossRef\]](#)
13. Gu, Z.; Chen, M.; Wang, C.; Zhuang, W. Static and Dynamic Analysis of a 6300 KN Cold Orbital Forging Machine. *Processes* **2020**, *9*, 7. [\[CrossRef\]](#)
14. Zhang, R. Sustainable Scheduling of Cloth Production Processes by Multi-Objective Genetic Algorithm with Tabu-Enhanced Local Search. *Sustainability* **2017**, *9*, 1754. [\[CrossRef\]](#)
15. Müller, J.M.; Däschle, S. Business Model Innovation of Industry 4.0 Solution Providers Towards Customer Process Innovation. *Processes* **2018**, *6*, 260. [\[CrossRef\]](#)
16. Borowski, P. Innovative Processes in Managing an Enterprise from the Energy and Food Sector in the Era of Industry 4. *Processes* **2021**, *9*, 381. [\[CrossRef\]](#)
17. Anser, M.; Khan, M.; Awan, U.; Batool, R.; Zaman, K.; Imran, M.; Sasmoko; Indrianti, Y.; Khan, A.; Bakar, Z. The Role of Technological Innovation in a Dynamic Model of the Environmental Supply Chain Curve: Evidence from a Panel of 102 Countries. *Processes* **2020**, *8*, 1033. [\[CrossRef\]](#)
18. Shim, S.-O.; Park, K. Technology for Production Scheduling of Jobs for Open Innovation and Sustainability with Fixed Processing Property on Parallel Machines. *Sustainability* **2016**, *8*, 904. [\[CrossRef\]](#)
19. Zhang, R.; Chiong, R. Solving the energy-efficient job shop scheduling problem: A multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption. *J. Clean. Prod.* **2016**, *112*, 3361–3375. [\[CrossRef\]](#)
20. Wadhwa, S.; Madaan, J.; Raina, R. A Genetic Algorithm Based Scheduling for a Flexible System. *Glob. J. Flex. Syst. Manag.* **2007**, *8*, 15–24. [\[CrossRef\]](#)
21. Sakaguchi, T.; Kamimura, T.; Shirase, K.; Tanimizu, Y. GA Based Reactive Scheduling for Aggregate Production Scheduling. *Manuf. Syst. Technol. New Front.* **2008**, *7*, 275–278. [\[CrossRef\]](#)
22. Dao, S.D.; Marian, R.M. Genetic Algorithms for Integrated Optimisation of Precedence-Constrained Production Sequencing and Scheduling. In *Lecture Notes in Electrical Engineering*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2012; Volume 130, pp. 65–80.
23. Ning, S.-S.; Wang, W.; Liu, Q.-L. An optimal scheduling algorithm for reheating furnace in steel production. *Control Decis.* **2006**, *21*, 1138–1142.
24. Gong, Z.; Li, J.; Luo, Z.; Wen, C.; Wang, C.; Zelek, J. Mapping and Semantic Modeling of Underground Parking Lots Using a Backpack LiDAR System. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 734–746. [\[CrossRef\]](#)
25. Ren, L.; Howard, D.; Jones, R.K. Mathematical Modelling of Biomechanical Interactions between Backpack and Bearer during Load Carriage. *J. Appl. Math.* **2013**, *2013*, 1–12. [\[CrossRef\]](#)
26. Kim, J.H.; Ma, S.B.; Kim, S.; Choi, Y.S.; Kim, K.Y. Design and verification of a single-channel pump model based on a hybrid optimization technique. *Processes* **2019**, *7*, 747. [\[CrossRef\]](#)
27. Li, C.; Zhai, R.; Liu, H.; Yang, Y.; Wu, H. Optimization of a heliostat field layout using hybrid PSO-GA algorithm. *Appl. Therm. Eng.* **2018**, *128*, 33–41. [\[CrossRef\]](#)

28. Choudhary, A.; Kumar, M.; Gupta, M.K.; Unune, D.K.; Mia, M. Mathematical modeling and intelligent optimization of submerged arc welding process parameters using hybrid PSO-GA evolutionary algorithms. *Neural Comput. Appl.* **2019**, *32*, 5761–5774. [\[CrossRef\]](#)
29. Xu, A.; Lu, Y.; Da, D.; Ti, N.; He, D. Hybrid direct hot charge rolling production for specific reheating furnace mode. *J. Univ. Sci. Technol. Beijing* **2012**, *34*, 1091–1096.
30. Mao, K.; Pan, Q.; Tasgetiren, M.F. Lagrangian heuristic for scheduling a steelmaking-continuous casting process. In Proceedings of the 2013 IEEE Symposium on Computational Intelligence in Scheduling (CISched), Singapore, 16–19 April 2013; pp. 68–74.
31. Hou, N.; He, F.; Zhou, Y.; Chen, Y. An efficient GPU-based parallel tabu search algorithm for hardware/software co-design. *Front. Comput. Sci.* **2020**, *14*, 1–18. [\[CrossRef\]](#)
32. Burduk, A.; Musiał, K.; Kochańska, J.; Górnicka, D.; Stetsenko, A. Tabu search and genetic algorithm for production process scheduling problem. *LogForum* **2019**, *15*, 181–189. [\[CrossRef\]](#)
33. Sun, L.; Luan, F.; Ying, Y.; Mao, K. Rescheduling optimization of steelmaking-continuous casting process based on the Lagrangian heuristic algorithm. *J. Ind. Manag. Optim.* **2017**, *13*, 1431–1448. [\[CrossRef\]](#)
34. Hu, H.X.; Lei, W.X.; Gao, X.; Zhang, Y. Job-Shop Scheduling Problem Based on Improved Cuckoo Search Algorithm. *Int. J. Simul. Model.* **2018**, *17*, 337–346. [\[CrossRef\]](#)
35. Zhang, L.; Yu, Y.; Luo, Y.; Zhang, S. Improved cuckoo search algorithm and its application to permutation flow shop scheduling problem. *J. Algorithms Comput. Technol.* **2020**, *14*, 1748302620962403. [\[CrossRef\]](#)
36. Ghosh, T.K.; Das, S.; Barman, S.; Goswami, R. Job scheduling in computational grid based on an improved cuckoo search method. *Int. J. Comput. Appl. Technol.* **2017**, *55*, 138–146. [\[CrossRef\]](#)
37. Wang, X.; Tang, L. Integration of batching and scheduling for hot rolling production in the steel industry. *Int. J. Adv. Manuf. Technol.* **2008**, *36*, 431–441. [\[CrossRef\]](#)
38. Lambiasi, F. Optimization of shape rolling sequences by integrated artificial intelligent techniques. *Int. J. Adv. Manuf. Technol.* **2013**, *68*, 443–452. [\[CrossRef\]](#)
39. Liu, H.; Liu, H.; Sun, F.; Fang, B. Kernel regularized nonlinear dictionary learning for sparse coding. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *99*, 1–10. [\[CrossRef\]](#)
40. Liu, H.; Wu, Y.; Sun, F.; Fang, B.; Guo, D. Weakly Paired Multimodal Fusion for Object Recognition. *IEEE Trans. Autom. Sci. Eng.* **2017**, *15*, 784–795. [\[CrossRef\]](#)
41. Leu, S.-S.; Hwang, S.-T. GA-based resource-constrained flow-shop scheduling model for mixed precast production. *Autom. Constr.* **2002**, *11*, 439–452. [\[CrossRef\]](#)
42. Yan, S.; Lai, W. An optimal scheduling model for ready mixed concrete supply with overtime considerations. *Autom. Constr.* **2007**, *16*, 734–744. [\[CrossRef\]](#)
43. Caridi, M.; Sianesi, A. Multi-agent systems in production planning and control: An application to the scheduling of mixed-model assembly lines. *Int. J. Prod. Econ.* **2000**, *68*, 29–42. [\[CrossRef\]](#)
44. de Prada, C.; Grossmann, I.; Sarabia, D.; Cristea, S. A strategy for predictive control of a mixed continuous batch process. *J. Process Control* **2009**, *19*, 123–137. [\[CrossRef\]](#)
45. Silva, A.F.D.; Marins, F.A.S.; Montevechi, J.A.B. Application of mixed binary goal programming in an enterprise in the sugar and energy sector. *Gestão Produção* **2013**, *20*, 321–336. [\[CrossRef\]](#)
46. Lim, M.; Zhang, D. An integrated agent-based approach for responsive control of manufacturing resources. *Comput. Ind. Eng.* **2004**, *46*, 221–232. [\[CrossRef\]](#)
47. Scarlat, E.; Boloş, M.; Popovici, I. Agent-based modeling in decision-making for project financing. *J. Econ. Comput. Econ. Cybern. Stud. Res.* **2011**, *5*–10, WOS:000292347200001.
48. Rody, R.; Mahmudy, W.F.; Tama, I.P. Using Guided Initial Chromosome of Genetic Algorithm for Scheduling Production-Distribution System. *J. Inf. Technol. Comput. Sci.* **2019**, *4*, 26–32. [\[CrossRef\]](#)
49. Que, Y.; Zhong, W.; Chen, H.; Chen, X.; Ji, X. Improved adaptive immune genetic algorithm for optimal QoS-aware service composition selection in cloud manufacturing. *Int. J. Adv. Manuf. Technol.* **2018**, *96*, 4455–4465. [\[CrossRef\]](#)
50. Sharma, S.; Chadha, M.; Kaur, H. Multi-step crossover genetic algorithm for bi-criteria parallel machine scheduling problems. *Int. J. Math. Oper. Res.* **2021**, *18*, 71. [\[CrossRef\]](#)
51. Li, S.; Yu, T.; Cao, X.; Pei, Z.; Yi, W.; Chen, Y.; Lv, R. Machine learning-based scheduling: A bibliometric perspective. *IET Collab. Intell. Manuf.* **2021**, *3*, 131–146. [\[CrossRef\]](#)
52. Li, M.W.; Hong, W.C.; Geng, J.; Wang, J. Berth and quay crane coordinated scheduling using multiobjective chaos cloud particle swarm optimization algorithm. *Neural Comput. Appl.* **2017**, *28*, 3163–3182. [\[CrossRef\]](#)
53. Lamghari, A.; Dimitrakopoulos, R.; A Ferland, J. A variable neighbourhood descent algorithm for the open-pit mine production scheduling problem with metal uncertainty. *J. Oper. Res. Soc.* **2014**, *65*, 1305–1314. [\[CrossRef\]](#)
54. Koo, J.; Kim, B.-I. Some comments on “Optimization of production scheduling with time-dependent and machine-dependent electricity cost for industrial energy efficiency”. *Int. J. Adv. Manuf. Technol.* **2016**, *86*, 2803–2806. [\[CrossRef\]](#)
55. Aghelinejad, M.; Ouazene, Y.; Yalaoui, A. Production scheduling optimisation with machine state and time-dependent energy costs. *Int. J. Prod. Res.* **2017**, *56*, 5558–5575. [\[CrossRef\]](#)
56. Yao, M.-J.; Huang, J.-X. Solving the economic lot scheduling problem with deteriorating items using genetic algorithms. *J. Food Eng.* **2005**, *70*, 309–322. [\[CrossRef\]](#)

57. Salga, P.; Szilágyi, R.; Herdon, M. Genetikusan alkalmazása a mezőgazdasági termelés optimalizálásában (The use of genetic algorithms in the optimization of agricultural production). In *E-agrárium & E-vidék: Agrárinformatikai Nyári Egyetem és Agrárinformatikai Fórum*; Magyar Agrárinformatikai Szövetség: Debrecen, Hungary, 2013; pp. 1–7.
58. Capusneanu, S.; Topor, D.I.; Hint, M.S.; Ionescu, C.A.; Coman, M.D.; Paschia, L.; Nicolau, N.L.G.; Ivan, O.R. Mathematical model for identifying and quantifying the overall environmental cost. *J. Bus. Econ. Manag.* **2020**, *21*, 1307–1328. [[CrossRef](#)]
59. Coita, I.F.; Cioban, S.; Mare, C. Is Trust a Valid Indicator of Tax Compliance Behaviour? A Study on Taxpayers' Public Perception Using Sentiment Analysis Tools. In *Proceedings of the 4th International Conference on Resilience and Economic Intelligence through Digitalization and Big Data Analytics*, Bucharest, Romania, 10–11 June 2021; ICESS Springer in Business and Economics: Bucharest, Romania, 2021.
60. Ionescu, C.A.; Fülöp, M.T.; Topor, D.I.; Căpușneanu, S.; Breaz, T.O.; Stănescu, S.G.; Coman, M.D. The New Era of Business Digitization through the Implementation of 5G Technology in Romania. *Sustainability* **2021**, *13*, 13401. [[CrossRef](#)]
61. Gubán, M. *Matematikai modellezés Az önfenntartó falugazdaság modellje a hálózati gazdaságban (Mathematical Modelling: The Model of the Self-Sustaining Village Economy in Network Economies)*; Budapesti Gazdasági Főiskola: Salgótarján, Hungary, 2005.