PRODUCTION SCHEDULING WITH GENETIC ALGORITHM

MIKLÓS GUBÁN[1], ÁKOS GUBÁN[2]

Abstract: In this paper, we deal with the scheduling problem of systems that have similar structure of the production-scheduling problem of assembly plants of discrete products. Firstly, the definition of order controlled, assembly plant of discrete productions is given, together with the problem of scheduling of inhomogeneous production lines. In the first part of this paper we give the model of the problem; thereafter we suite the model to genetic Algorithm. In this part we align the parameters of the well-known method to the parameters of our individual model. In the second part, we analyze and examine the run results of our computer program. In this section we present specific run results and show the system usability. In the last part we summarize when and in what circumstances the system can be used.

Keywords: mathematical model, Genetic Algorithm, production scheduling, analysis, ANOVA, normality test

1. Introduction

In this paper, we deal the scheduling problem of systems that have got similar structure of the production-scheduling problem of assembly plants of discrete products. These assembly plants have got more inhomogeneous production lines and the system is controlled by orders of a period. Similarly to the digital service channels, the problems of parallel transport routes, or the multichannel information transmission systems, etc. For simplicity, the problem of the production scheduling is dealt and the terms of the production scheduling are used for the modelling of the problem.

Firstly, the definition of order controlled, assembly plant of discrete productions is to give, together with the problem of scheduling of inhomogeneous production lines.

The system of discrete products formulated in the introduction is a measurable piece of product, and production time and cost of n-elements of the same products is k times the production time and cost of a product ($k \in \mathbb{N}^+$). The system is controlled by orders, so the orders of production period of the assembly plant have been known before manufacturing. Therefore it is known which parts and how many parts are necessary to assembly the products of the orders. In order to be compatible with other systems (for example pocket transmission) each same products of a (virtual) order will be manufactured in a row in the same production line. (In traditional production, this constraint can be omitted.) Inhomogeneous production line means that the turnaround time and cost of manufacturing of the products depend on the product type.

[1] college professor, Dennis Gabor College
guban@gdf.hu
H-1119 Budapest, Mérnök st. 39
[2] college professor, Budapest Business School
akos.guban@pszfb.bgf.hu
H-1149 Budapest, Buzogány st. 10-12

2. The production scheduling system

After the basic definitions have been defined, the original problem will be given. The assembly plant has got *n* production lines ($n \in N^+$). Denote C_i $(1 \leq i \leq n)$ is the flexible production line *i*. Every production line can manufacture every product. (If a line cannot manufacture a product, then the manufacturing of that particular product would be assigned to this line very long turnaround time and very high production cost.) The assembly plant can manufacture (assemble) *m* different product types($m \in N^+$). Our model do not contain deadline, because it is assumed that the all possible solutions satisfy the criteria of deadline of the orders. Formally it can be described: let $l \in \mathbb{N}^+$ orders waiting for manufacturing be in the system.

Denote S_{ij} the serial *j* of the order *i*.

$$S_{ij}: 1 \leq j \leq k_i \tag{1}$$

where k_i is the number of the serials of order *i*.

Denote *s* the total number of serials of all orders

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

For example, let $S_{ij} \coloneqq \begin{Bmatrix} e_k e_k e_k \dots e_k \\ 1\ 2\ 3 \dots |S_{ij}| \end{Bmatrix}$ be. This kind of serials cannot be divided into further parts. Let $\sigma(S_{ij}) = p$ be the number of serial S_{ij}. It will be used the function $\tau(p) := j$ for further references. The model of the original system contains three matrices. These matrices connect the products with lines. The first matrix is the turnaround time of the product types on the lines, the second is the unit cost of the product types on the lines, and the third is resetting time of the product types on the lines. Formally these are $\mathbf{T}_{n\times m} = [t_{ij}]$ is turnaround time of product *j* on line *i*. $\mathbf{P}_{n\times m} = [p_{ij}]$ is manufacturing cost 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 serial *j* to serial *l* on line *i*. $(1 \leq i \leq n), (1 \leq j \leq s), (1 \leq l \leq s)$

The main goal of solving this problem is to find the optimal or near-optimal allocation satisfied the conditions of the original model. Its aim is twofold. Two achieving goals are minimizing the costs and minimizing the lead time. It seems the objective function has got two opposite components. If the goal is to minimize the costs, then every serial will be assigned to its optimal production line in the best result. If the optimal production line of every serial is same, then the lead time would be maximal. And vice versa, if every serial is assigned to the line with optimal lead time, then every serial might be assigned to the production line with highest cost. It seems the optimal result is between them.

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

Let an optimization task formulate above system which will create all series on a production line, the following objectives are met at least one goal:

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, because the follow objective function handles cases 1 and 2:

$$\lambda \cdot \max_j T_j + \mu \cdot \sum_j K_j \to \min \tag{3}$$

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

3. The Mathematical Model

3.1. The Condition System. Let $\mathbf{X}_{s\times n\times s} = [x_{ijk}]$ where denote $x_{ijk} = 1$ that the serial *i* is assigned to manufacturing element *k* of production line *j*.

$$x_{ijk} \in \{0; 1\}\ (1 \le i \le s; 1 \le j \le n; 1 \le k \le s) \tag{4}$$

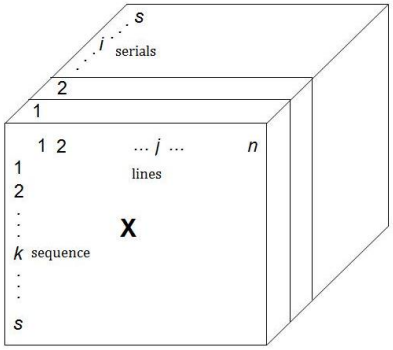

Fig. 1. The hypermatrix **X**

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

$$\sum_{j=1}^{n}\sum_{k=1}^{s} x_{ijk} = 1\ (1 \le i \le s) \tag{5}$$

Each production line, and within a production serial number is assigned to up one series:

$$\sum_{i=1}^{s} x_{ijk} \le 1\ (1 \le j \le n, 1 \le k \le s) \tag{6}$$

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

$$\sum_{i=1}^{s} x_{ijk} - \sum_{i=1}^{s} x_{ij(k+1)} \ge 0\ (1 \le j \le n, 1 \le k < s) \tag{7}$$

3.2. Objective function. Let *trunc* function be the following

$$\mathrm{trunc}(x) = \begin{cases} 1 \text{ if } x \ge 2 \\ 0 \text{ otherwise} \end{cases}. \tag{8}$$

The manufacturing cost of the production line *j* is

$$K_{p_j}(\mathbf{X}) = \sum_{i=1}^{s}\sum_{k=1}^{s} p_{j\tau(i)} x_{ijk}\,. \tag{9}$$

The resetting cost of the production line *j* is

$$K_{t_j}(\mathbf{X}) = \sum_{i=1}^{s}\sum_{k=1}^{s-1}\sum_{r=1}^{s} d_{\tau(i)\tau(r)} \cdot \mathrm{trunc}\left(x_{ijk} + x_{rj(k+1)}\right). \quad (10)$$

The total cost of the production line *j* is

$$K_j(\mathbf{X}) = K_{p_j}(\mathbf{X}) + K_{t_j}(\mathbf{X}) = \sum_{i=1}^{s}\sum_{k=1}^{s} p_{j\tau(i)} x_{ijk} + \\ + \sum_{i=1}^{s}\sum_{k=1}^{s-1}\sum_{r=1}^{s} d_{\tau(i)\tau(r)} \cdot \mathrm{trunc}\left(x_{ijk} + x_{rj(k+1)}\right) \quad (11)$$

The manufacturing time of the production line *j* is

$$T_{p_j}(\mathbf{X}) = \sum_{i=1}^{s}\sum_{k=1}^{s} t_{j\tau(i)} x_{ijk}. \quad (12)$$

The resetting time of the production line *j* is

$$T_{t_j}(\mathbf{X}) = \sum_{i=1}^{s}\sum_{k=1}^{s-1}\sum_{r=1}^{s} g_{\tau(i)\tau(r)} \cdot \mathrm{trunc}\left(x_{ijk} + x_{rj(k+1)}\right). \quad (13)$$

Total manufacturing time of the production line *j* is

$$T_j(\mathbf{X}) = T_{p_j}(\mathbf{X}) + T_{t_j}(\mathbf{X}) += \\ = \sum_{i=1}^{s}\sum_{k=1}^{s} t_{j\tau(i)} x_{ijk} + \sum_{i=1}^{s}\sum_{k=1}^{s-1}\sum_{r=1}^{s} g_{\tau(i)\tau(r)} \cdot trunc\left(x_{ijk} + x_{rj(k+1)}\right) \quad (14)$$

The total objective function is

$$\mathrm{f(X)} = \lambda \cdot \max_{\mathrm{j}\in\{1,\ldots,\mathrm{n}\}} \mathrm{T_j(X)} + \mu \cdot \sum_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{K_j(X)} = \lambda \cdot \max_{j\in\{1,\ldots,n\}}\left(\sum_{i=1}^{s}\sum_{k=1}^{s} t_{j\tau(i)} x_{ijk} + \right. \\ \left. \sum_{i=1}^{s}\sum_{k=1}^{s-1}\sum_{r=1}^{s} g_{\tau(i)\tau(r)} \cdot trunc\left(x_{ijk} + x_{rj(k+1)}\right)\right) + \mu \cdot \sum_{j=1}^{n}\left(\sum_{i=1}^{s}\sum_{k=1}^{s} p_{j\tau(i)} x_{ijk} + \right. \\ \left. \sum_{i=1}^{s}\sum_{k=1}^{s-1}\sum_{r=1}^{s} d_{\tau(i)\tau(r)} \cdot trunc\left(x_{ijk} + x_{rj(k+1)}\right)\right) \to \min. \quad (15)$$

4. Solving the task of the mathematical model

4.1. Previous solving methods and examinations. The main problem is the size of the model of the original problem. In case of n number of lines and 1 orders, then the size of problem will be $D_{li} = \sum_{i=1}^{l}(l-i-1)!\frac{n!}{(n-i)!}$. For example in case of 4 lines and 50 orders then the size would be $1.3 \cdot 10^{65}$.

Solving this problem with exact method is very difficult, so we used heuristic methods earlier.

The first used heuristic method was a special greedy algorithm. This solving method was based on combining the same product, and then the group of products assigned to the lines is depended on the parameters of objective function. Then the peak assignment was eliminated by "manual" tuning. This result was useful, but the efficiency analysis showed that the appropriate target was not reached for every order.

The next experiment was solving with the Ant Colony Algorithm (ACO). Running results of software were examined by statistical analysis and the results of ACO showed big difference. It was very difficult to determine whether the result was close to optimum. But

the advantage of this solving method is that two parameters (leading time and cost) were handled at the same time and the relationship between the two parameters was controlled easily.

As expected, if the number of the step and number of agents were increased then it would get better average objective value. Surprisingly minimum aggregate objective value was given by less step number. But average objective value was less with 1.5% for larger number of steps.

It is clear, this method is not convergent, only low objective value could be earned in case of many agents, and long runtime and pheromone value prorated path length. Not worthwhile to examine in detail, because the results of the examined test data (about 1000 runs) with same parameters can be very different. It was established that the ACO good for pre-processing to other methods. Hereafter Genetic Algorithm will be used which is more effective and proved convergent method. The results of ACO will be used for creating of initial population of GA.

The results of ACO give good individuals of initial population because the individuals are "away" from each other relatively.

4.2. The structure of a chromosome. The method is based on a well-designed chromosome. If the chromosome with locuses can be determined effectively then it will give a good chance to create a useful method.

In the first step the number of chromosomal genes is determined

$$g = 2 \cdot s. \tag{16}$$

Assign an integer number to every serial of the original problem with function $\sigma(S_{ij}) = p$ (Table I.). Structure of chromosome shows the Table II.

Table I. Function $\sigma(S_{ij})$

$\sigma(S_{ij})$	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}

Table II. Structure of 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 serial i.

N_{ij_i}: production sequence number of serial i on the line j_i (production sequence). It is not important the strict sequence (1,2,…,k_i), only the order of the values determines the sequence of serials on the line. If two serials have got same values, then the serial with lower sequence number will be manufactured first. This consideration will accelerate the algorithm (every result will be possible result). For example in Table III.

Table III. An example for a chromosome

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

In the Table III. two lines (#0 and #1) is used with 5 locuses of a chromosome. The first gray block shows that the serial #1 assigned to line #0

The serial #1 has got sequential number 10 and the serial #4 has got sequential number 2 on the line #0. The serial #2 has got sequential number 4, the serial #3 has got sequential number 3 and the serial #5 has got sequential number 4 on the line #1. The allocation can be seen on Table IV.

Table IV. An example for an allocation

Line #0	Serial #4	Serial #1	
Line #1	Serial #3	Serial #2	Serial #5

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

4.2.1. Crossover. Denote *p* the number of population.

The first operator of GA is the crossover. This operator creates two new individuals (offspring) from two parent individuals. In the scheduling problem two new allocations will be created from two older allocations. The algorithm will be the following:

Let *CRV* be a random integer value from interval [1,...,$\frac{p}{2}$] (half of number of population). *CRV* is the number of crossovers.

1. Choose two individuals randomly from the actual population.
2. Choose a crossover position *POS* from interval [1,...,2*s*] and a value *LEN* from interval [1,…,2*s* – *POS*]. (*POS* shows which locus is the start locus of the crossover and *LEN* shows the number of locus will be changed.)
3. Create the two new chromosomes changed locuses of two parents from position *POS* with length *LEN*. For example see Table V.
4. Do with step 2 and 3 *CRS* times.

For example, let two chromosomes be given in Table V. Let he start point be 2 and length of crossover be 6 (Table VI.).

Table V. Chromosomes of the parents

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

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

Table VI. Chromosomes of the offspring

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

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

Denote: The length of crossover can be an odd number. Then the value of sequence number of manufacturing is unchanged in the serial 4, only number of production line is changed. This result is possible as described above.

4.2.2. Mutation. The role of mutation is to add chromosomes into population which would never added to the population simply by crossover. Mutation is executed for only one gene. Gene can be a production line or a gene of sequence number.

Let *MUT* be a random integer value from interval [1,..., $\frac{p}{2}$] (half of number of population). This value determines how many mutations will be executed.

1. Choose *MUT* individuals from the population.
2. Choose the first individual.
3. Choose a locus randomly from this individual. If it has got even number then generate an integer value from interval [1,…,*s*] and change value of locus with this value.
4. If it has got odd number then generate an integer value from interval [0,…, $n-1$] and change value of locus with this value.
5. Choose the next (if there are elements) individual of selected individuals and go to step 3.

4.2.3. Selection. The combination of two selection methods will be used:

- fitness proportional selection,
- elite list selection.

Let $p + \left[\frac{p}{2}\right]$ be number of chromosomes created with genetic operations. Then the number of the old and new individuals is $q = 2p + \left[\frac{p}{2}\right]$. Search the chromosome with the best fitness value. If more than one individual are found then choose one of them randomly. This individual will be the first element of new population. After then choose $p - 1$ individuals from $q - 1$ new chromosomes.

Let the number of new population be p. Calculate the fitness value of all chromosomes (*evaluating*). Sort the individuals of population in descending order. After then calculate the next cumulative value for every individuals:

$$F_i = \sum_{j=1}^{i} f(k_j) \ (i = 1, \dots, q-1) \tag{17}$$

Generate a random integer value from interval [0,…,F_m]. Choose the index i based on the following criteria

$$F_{i-1} \le l < F_i, \text{if } i > 1 \tag{18}$$

and

$$l < F_i, \text{if } i = 1. \tag{19}$$

If this individual has been chosen then repeat this procedure again.

Eventually it will be got a population with p individuals.

4.2.4. Convergence of procedure. This procedure will converge to optimum. The convergence is provided by two facts. One of them is the fitness value is bounded below (the values of fitness cannot be negative). The other fact is the elite list selection. With the elite list selection the best individuals of population will be added in the new population. So

the smallest fitness value of the new population is less or equal than the smallest fitness value of the previous population. So the best fitness values of populations form a monotonically decreasing sequence. Consequently the sequence is convergent.

The question is this sequence will converge to the optimum. The convergence to optimum of GA is proved in [1].

5. Analysis of runtime results

150 random runtime results were analysed in every case. The results given by the method are examined depending on parameters. In the case of examined samples when turnaround time is optimized then the turnaround time cannot be less than 16163 time unit and cost cannot be less than 18494 cost units. If the cost is optimized then the turnaround time cannot be less than 18189.75 time unit and the cost cannot less be then 18189.75 cost units.

The following sample will be examined in several cases. Figure 2. shows the result with parameter $\alpha = 1$. It seems the system seeks a uniform load.

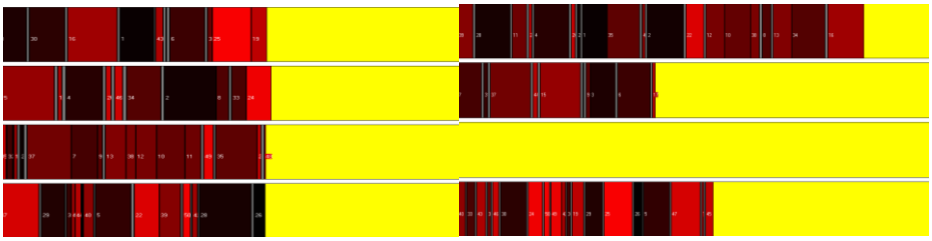

Fig. 2. Uniform allocation in case of parameter $\alpha = 1$; Allocation in case of high manufacturing cost

It seems that the system ignores the line with high cost.

5.1. Analysis of the dependence on the number of population. The results was examined with $\alpha = 0.9$ and $\beta = 0.1$ and 4000 new populations in which were 10 or 20 or 30 individuals. Table VII. shows the descriptive statistics data.

Table VII. Descriptive Statistics data

	Minim.	Maxim.	Mean				
	Minimum	Maximum	Mean	Std. Deviat.	Variance	Skewness	Kurtosi
aggr10	19834,90	20389,80	20055,52	104,211	10860,052	,533	,470
aggr20	19787,40	20256,50	19982,62	92,434	8544,115	,306	-,260
aggr30	19765,60	20201,40	19957,57	86,106	7414,326	,232	-,034

In the first step normality of sample was analyzed, Table VIII. shows that all three cases followed a normal distribution, so averages and deviations can analyzed. (Software SPSS V20 was used for examination.)

Homogeneity of variance is met according to The Levene test.

Table VIII. Normality test

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of összevont10 is normal with mean 20 055,52 and standard deviation 104,21.	One-Sample Kolmogorov-Smirnov Test	,856	Retain the null hypothesis.
2	The distribution of összevont20 is normal with mean 19 982,62 and standard deviation 92,43.	One-Sample Kolmogorov-Smirnov Test	,757	Retain the null hypothesis.
3	The distribution of összevont30 is normal with mean 19 957,58 and standard deviation 86,11.	One-Sample Kolmogorov-Smirnov Test	,768	Retain the null hypothesis.

Asymptotic significances are displayed. The significance level is ,05.

Table IX. Bonferroni test of aggregated data

ANOVA

Aggregated

Groups	Sum of Squares	df	Mean Square	F	Sig.
Between	776800,293	2	388400,146	43,45	,00
Within	3995955,547	447	8939,498		
Total	4772755,840	449			

According to ANOVA the equality of the average cannot be accepted, so Post –hoc tests have to be used:

Table X. An example for an allocation

Dependent Variable: Aggregated

(I) Group	(J) Group	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
10	20	72,904*	10,9175	,000	46,6688	99,139
	30	97,948*	10,9175	,000	71,7128	124,18
20	10	-72,904*	10,9175	,000	-99,1392	-46,668
	30	25,044	10,9175	,067	-1,1912	51,279
30	10	-97,948*	10,9175	,000	-124,183	-71,712
	20	-25,044	10,9175	,067	-51,2792	1,191

*. The mean difference is significant at the 0.05 level.

As shown in Table X. the population with 10 individuals different from populations with 20 or 30 significantly, but the population with 20 individuals is not significant difference from population with 30 (95%). Obviously the difference decreases stochastically since the method is convergent.

The cost parameter is normally distributed, too. The average of cost is same with previous case.

The expected values of turnaround time are significantly remained the same, because the main optimizing parameter is the cost (weighted with 0.9).

Table XI. ANOVA of turnaround

turnaround

Groups	Sum of Squares	df	Mean Square	F	Sig.
Between	48140,320	2	24070,160	,301	,740
Within	35710164,000	447	79888,510		
Total	35758304,320	449			

5.2. Examination of running number. In the following examination running number is the variable (for values 1000, 2000 an 3000) and the change will affect the results of the runs. Its value of total objective function will examined without its components. Table XII. shows that the sample following normal distribution can acceptable.

Table XII. Descriptive statistics and results of the test

	Minimum	Maximum	Mean	Std. Dev.	Variance	Skewness	Kurtosis
s1M	19898,40	20382,80	20141,17	106,63	11368,92	**,016**	-,684
s2M	19826,40	20298,10	20034,51	92,60	8574,03	**,099**	-,148
s3M	18643,00	19929,90	19264,88	238,03	56659,30	**-,003**	-,071

Hypothesis Test Summary

	Null Hypothesis	Test	Sig.	Decision
1	The distribution of pop1000 is normal with mean 20 141,17 and standard deviation 106,63.	One-Sample Kolmogorov-Smirnov Test	,248	Retain the null hypothesis.
2	The distribution of pop2000 is normal with mean 20 034,51 and standard deviation 92,60.	One-Sample Kolmogorov-Smirnov Test	,988	Retain the null hypothesis.
3	The distribution of pop3000 is normal with mean 19 264,89 and standard deviation 238,03.	One-Sample Kolmogorov-Smirnov Test	,999	Retain the null hypothesis.

Asymptotic significances are displayed. The significance level is ,05.

According to the ANOVA test and Post-hoc analysis the equality of averages can be rejected so the averages difference significantly for all groups.

Table XIII. Tests of step variables

Bonferroni

I	J	Mean Differ. (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
1	2	106,662*	18,45141	,000	62,323	151,00
	3	876,286*	18,45141	,000	831,9468	920,625
2	1	-106,662	18,45141	,000	-151,001	-62,3228
	3	769,624*	18,45141	,000	725,285	813,963
3	1	-876,286*	18,45141	,000	-920,625	-831,946
	2	-769,624*	18,45141	,000	-813,963	-725,284

*. The mean difference is significant at the 0.05 level.

steps

Groups	Sum of Squares	df	Mean Square	F	Sig.
Between	68578751,87	2	34289375,935	1342,886	,00
Withins	11413735,87	447	25534,085		

It can be stated that the average of the values of the objective function will be less for larger step number. The distributions cannot be same so there were very different for 3000 runs. The values are shifted to the left, so every value generated are lower than in other two cases.

5.3. Examination of the different parameters. In the following examination the parameters alpha and beta will changed. In order to examine the values of total objective function and its components will got during the runs. If the rate of components of the objective function is changed then it will got interesting results. (The values of total objective function are listed detailed. Every sample follows a normal distribution, so the expected value will be examined.)

Table XIV. Test of aggregated data

aggregated

Groups	Sum of Squares	df	Mean Square	F	Sig.
Between	128810534,4	4	32202633,6	707,733	,000
Within	33898315,9	745	45501,1		
Total	162708850,3	749			

Bonferroni

I	J	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					L.Bound	U.Bound
5	1	-2732,29333*	42,092	,00	-2850,80	-2613,79
	2	-501,60667*	42,092	,00	-620,11	-383,10
	3	-1050,29333*	42,092	,00	-1168,80	-931,79
	4	142,64067*	42,092	,00	24,13	261,15

*. The mean difference is significant at the 0.05 level.

It is proved by ANOVA test and Post-hoc analysis. The first and fourth samples are consistencies, and third and fourth are consistencies.

The results of the turnaround time are worse; between the averages of samples was no consistency.

It can see (Table XIV.) if the parameter beta is increased then the costs of results will be less.

The examination shows that the values of total objective function have no relationship with parameters for changing parameters alpha and beta. This may be because orders of magnitude of cost and turnaround time are very different in the objective function.

According to analysis changing of parameters alpha and beta cannot be arbitrary. Increasing of number of the population or increasing of step number of runs reduce stochastically the difference between the optimum given by the runs and the real optimum, so

$$P\left(\left[f_{li} - f_{opt}\right] < \varepsilon\right) \geq P\left(\left[f_{ki} - f_{opt}\right] < \varepsilon\right) \text{ if } l > k \tag{20}$$

and

$$P\left(\left[f_{li} - f_{opt}\right] < \varepsilon\right) \geq P\left(\left[f_{lj} - f_{opt}\right] < \varepsilon\right) \text{ if } i > j \tag{21}$$

The selecting of parameters alpha and beta is more interesting question. The total objective function are very sensitive for these parameters, so these parameters must be defined precisely for the practically problems. The values of the parameters depend on which parameter is preferred in the given production system. The selection of appropriate values should be run more runs of simulation real orders.

6. Conclusions

The first goal was to create an exact mathematical model of the problem. It was important to create an informatics model of GA.

What caused the unreliability of the earlier heuristic method? According to analysis the results only can be near the optimum in special case for this problem. Unfortunately the solution of ACO behaved differently. The advantage of this solution method was the bypassing of problem of parameters alpha and beta. The method handled automatically cost and turnaround time and therefore the solution method did not give a perfect solution. Unfortunately, the results were possible solutions, but not optimal solutions.

The third goal was to give a GA model connected to the mathematical model. The original model of GA can be adapted with a small change, because chromosomes of scheduling problem are specially. The genes have got two special members, so the genetic

operators must be adapted for this special structure. The created chromosomes and operators are suitable for the task of the scheduling model.

The role of parameters was showed by analysis. This analysis helped to present that the given solution of GA is useful in the practice. The result of this analysis was the parameter expect two parameters (alpha and beta) can be predetermined. In this paper the special production line was not presented. These special cases have been analysed and these lines behaved as expected.

Further plans are the handling of the stochastically requirement of the production lines (JIT) and handling of special rescheduling problems (accidental failures, etc.) with handling of spare time generated automatically.

Summarize, it can be told that a mathematical model has been created for the original problem, and the GA method connected to the model solves the scheduling problem effectively. The conducted analyse proves the corresponding of the results and shows the sensitivity of the system.

Literature

[1] Greenwood, G. W.; Zhu, Q.: *Convergence in evolutionary programs with selfadaptation in evolution strategies*. Evolutionary Computation, 9(2): 2001. pp. 147-157.

[2] Sen, S.; Hwang, S. T.: *GA-based resource-constrained flow-shop scheduling model for mixed precast production*, Automation in Construction, Volume 11, Issue 4, June 2002, pp. 439–452

[3] Guban, M.; Cselényi, J.: *The method and analysis of establishment of logistic-oriented postponement assembly plants*, Chapter 25, DAAAM International Scientific Book, 2004, Wien, B. Katalinic (Ed.) pp. 255-264, Published by DAAAM International, ISBN 3 901509 38 0, ISSN 1726 9687, Vienna, Austria

[4] Guban, M.; Cselényi, J.: *The ILP model and a solving method the first phase of the establishment of postponed assembly plants*. Internationalen Maschinenbaukonferenz COMEC 2006 Santa Clara, Cuba (07.11 bis 09.11.06) ISBN 959-250-295-I

[5] Psarras, J.: *GA-based decision support systems in production scheduling*, International Journal of Intelligent Systems Technologies and Applications (IJISTA), Volume 2 - Issue 1 – 2007

[6] Sakaguchi, T.; Kamimura, T.; Shirase, K.; Tanimizu, Y.: GA *Based Reactive Scheduling for Aggregate Production Scheduling*, Manufacturing Systems and Technologies for the New Frontier 2008, Part 7, pp. 275-278, DOI: 10.1007/978-1-84800-267-8_56

[7] Hu, H.; Tong, S. J.: *A Hybrid GA-CP Approach for Production Scheduling*, Natural Computation, 2009. ICNC '09. Fifth International Conference on, Volume: 4, 2009. pp. 86-91

[8] Gubán, M.; Gubán M. Jr.; Kerepeszki, I.: *Mathematical model and solution method of ordering controlled virtual assembly plants and service centres* 3rd Central European Conferenc in Regional Science October 7th-9th, 2009. Kosice, Slovak Republic ISBN 978-80-553-0329-1 pp. 316-325

[9] Gubán, M.: *Non-linear programming model and solution method of ordering controlled virtual assembly plants* LOGISTICS – THE EURASIAN BRIDGE Materials of V International scientifically-practical conference Krasnoyarsk Russia 2011.03.2-3 ISBN 978-5-94617-223-3 pp. 49-58

[10] Lengyel, Zs.: Szív és ész. *Elméleti meggondolások az intelligencia vizsgálatáról*. Hadtudományi Szemle, Budapest, 2010. 3, pp. 105-113. HU ISSN 2060-0437