

# Choosing the Optimal Production Strategy by Multi-Objective Optimization Methods

**Ján Čabala, Ján Jadlovský**

jan.cabala@tuke.sk, jan.jadlovsky@tuke.sk

Faculty of Electrical Engineering and Informatics, Technical University of Košice,  
Letná 9, 04001 Košice, Slovakia

---

*Abstract: This paper presents the solution of multi-objective optimization of the production process of an automated assembly line model, where combination of conventional mathematical methods and methods of artificial intelligence is used. Paper provides the description of methods used in this process, modifications that were realized in the computational process of NSGA - II evolutionary algorithm as well as the solution of the production process optimization respecting all the defined constraints. The first part of the solution, the definition of the set of non-dominated (Pareto optimal) alternatives, is realized by the modified NSGA – II evolutionary algorithm. From the Pareto optimal solutions, choosing the best solution using various mathematical metrics is presented. Approach for the synthesis of the results obtained from various mathematical metrics used to resolve the task is also mentioned with the scope of objectivization of the optimization process.*

*Keywords: assembly systems; genetic algorithms; optimization methods; mathematical programming; Pareto optimization*

---

## 1 Introduction

The optimization process is usually used for choosing the best possible solution of a particular task. To ensure that this solution can be found in a qualified manner, it is necessary to create a mathematical model as accurately as possible in order to describe the optimization task properly. The model itself includes quantifiable parameters (objective functions) for measuring the rate of success of optimized criteria (e.g. profit). The model may also contain constraints (e.g. maximum amount of invested capital). Modelling is followed by finding a solution for a given optimization task using a suitably chosen algorithm. It is also necessary to verify and evaluate the obtained solution (whether it is a valid solution to the resolved task) and to interpret the result correctly.

Methods of mathematical programming are used to solve optimization tasks with one objective function. Depending on the type of objective function, these methods can be divided into linear or non-linear programming methods, integer programming methods, parameter programming methods, stochastic programming methods, etc. The overview of optimization algorithms for solving various optimization tasks can be found in [24].

In the process of optimization of production lines, one of the main requirements is the definition of optimization goals, what may include the minimizing number of workstations (posts), minimizing the time of production cycle, maximizing production line efficiency, cost minimization, maximizing profit, maximizing or minimizing the various factors affecting operations performed at a weak spots of the production line. Each of these objectives should be defined by an objective function, which values should be minimized or maximized (depending on the goal of the optimization process). [3]

In most cases, there is more than one objective, which should be optimized to satisfy the needs of the decision-maker. There are two basic types of methods used for resolving multi-objective optimization (MOO) tasks:

- Conventional methods – these are represented by mathematical metrics for choosing the best solution within the defined portfolio of alternative solutions
- Methods of artificial intelligence (AI) - represented by a number of algorithms VEGA (Vector Evaluated Genetic Algorithms) group, mostly used for defining the set of non-dominated solutions.

For defining the Pareto optimal set of solutions, conventional approaches aggregate the objective functions into a simple parametrized objective function. Several runs with different parameters of this objective function are realized in order to approximate the Pareto front. [31]

Except for conventional methods, artificial intelligence methods (especially evolutionary and genetic algorithms) are also used to solve the problems of MOO. Evolutionary algorithms represent the approach for finding the best solutions with trying a relatively small number of possible solutions, as the scope of possible solutions is very extensive in many cases. Some evolutionary algorithms from this group can cope with various forms of objective functions and resolve tasks with complicated Pareto sets (MOEA/D or NSGA-II) [18]. This complexity was one of the reasons for choosing the NSGA-II algorithm for finding the Pareto optimal solutions. The evolutionary algorithm is based on the population of individuals. This population usually contains more individuals, often hundreds or even thousands. The first population is typically generated randomly. This population is then reproduced and the best individuals are kept in the evolutionary process, while the worst are excluded. [5]

For solving multi-objective optimization problems, VEGA (Vector Evaluated Genetic Algorithms) are used. Closer description of this group of algorithms can

be found in [5] and [1]. The motivation for choosing the combination of conventional and AI methods to solve the MOO problem of the production process on assembly line model placed within Center of Modern Control Techniques and Industrial Informatics (CMCT&II) within DCAI FEEI TUKE arose seeing the wide application potential of these methods. In system reliability, redundancy allocation problem was resolved using the NSGA algorithm in [28]. In assembly line balancing, the use of multi-objective genetic algorithm (MOGA) is presented in [19], ant colony algorithms are described and applied in [2] and [25], tabu search algorithm is mentioned in [17]. Genetic algorithms can be also be used for solving assembly sequence planning, which is shown in [11]. Other application possibilities are mentioned in [14]. Other possibilities of using AI methods for optimization is described in [29], while the novel AI optimization approaches and algorithms are presented in [23], [27], [22].

The goal of the MOO process described in the paper is definition of the number of different types of products, satisfying the goals (maximizing profit and maximizing the amount of saved time) and respecting the constraints (limited supplies, limited storage capacity) of the optimization process. Combination of 2 different approaches used for solving the MOO task, as well as synthesis of partial results obtained by using different conventional methods is considered as a novel approach in the field assembly line optimization.

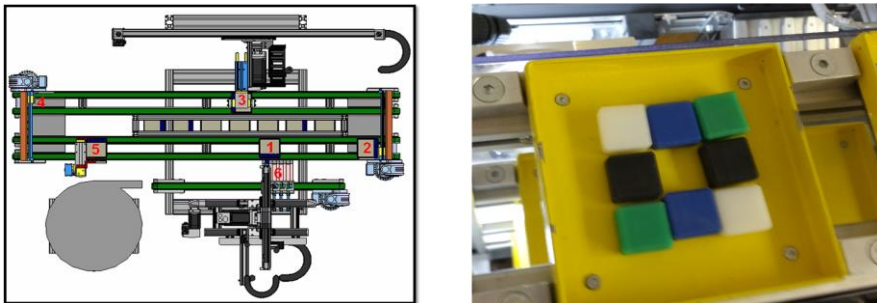


Figure 1

Schematic view on the assembly line model and the final product

In the first phase of the multi-objective optimization, artificial intelligence methods were used, namely the modified NSGA-II evolutionary algorithm for specifying the Pareto front of this multi-objective optimization task. It is also possible to use conventional methods to identify the Pareto front, but especially in the complex types of objective functions, it is easier to use evolutionary algorithms. In the second phase, from the Pareto front solutions, the optimal solution to the MOO task is chosen using conventional methods. Since there are many optimal solution selection methods, synthesis of solutions obtained by various methods was realized and the optimal solution for this task was chosen.

Within the paper, methods used for solving MOO tasks are presented in the second chapter, where 2 main approaches are described: conventional mathematical approach and artificial intelligence approach. Using the combination of these approaches for solving the optimal production process of assembly line model is presented in the Chapter 3.

## 2 Multi-Objective Optimization Task and Methods

The optimization task generally has the form of minimizing (maximizing) an objective function

$$f = \{x_1, x_2, \dots, x_n\} \quad (1)$$

while taking into account the restrictive conditions (constraints)

$$g_i = \{x_1, x_2, \dots, x_n\}, \text{ for } i = 1, 2, \dots, n \quad (2)$$

The multi-objective optimization (MOO) task is defined by the mathematical description of optimized system. This system is characterized by the functional  $J(\mathbf{x}) = (J_1(\mathbf{x}), J_2(\mathbf{x}), \dots, J_k(\mathbf{x}))$ , where  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  is the vector of variables used to define the mathematical model of the system. Solution of the MOO task is  $\mathbf{x}^* \in \{X\}$ , which is the optimal solution of the functionals  $J_1(\mathbf{x}), J_2(\mathbf{x}), \dots, J_k(\mathbf{x})$ .

### 2.1 Conventional Methods

Conventional methods of MOO are closely described in [16]. Solving the MOO task using the conventional methods applied on the economical investments is part of [7].

#### 2.1.1 Methods Defining the Set of Non-Improving Points

In this group of methods there is no hierarchy of objectives. MOO task can be defined as minimization of vector  $J(\mathbf{x}) - J(\mathbf{x}_\alpha)$ , where  $\mathbf{x}_\alpha$  is the optimal solution of  $\alpha^{\text{th}}$  objective of MOO task.

##### *Quadratic Metric*

The most common decision parameter using this metric is minimum of the squared difference between values of objective functions for solution  $\mathbf{x}$  and values of objective functions for ideal solution  $\mathbf{x}_\alpha$ .

$$R(x) = \sum_{\alpha=1}^k (J_\alpha(\mathbf{x}) - J_\alpha(\mathbf{x}_\alpha))^2 \quad \alpha = 1, 2, \dots, k \quad (3)$$

$$\text{opt } R(x) = \min_{x \in \{X\}} \sum_{\alpha=1}^k (J_{\alpha}(x) - J_{\alpha}(x_{\alpha}))^2 \quad (4)$$

Frequently, the deciding parameter of this metric is furtherly divided by values of optimal solution, in order to get a result in dimensionless form. The formula for this metric is defined as:

$$R(x) = \sum_{\alpha=1}^k \frac{(J_{\alpha}(x) - J_{\alpha}(x_{\alpha}))^2}{J_{\alpha}(x_{\alpha})^2} \quad (5)$$

$$\text{opt } R(x) = \min_{x \in \{X\}} \sum_{\alpha=1}^k \frac{(J_{\alpha}(x) - J_{\alpha}(x_{\alpha}))^2}{J_{\alpha}(x_{\alpha})^2} \quad (6)$$

### **Linear Metric**

This metric is defined as the sum of variations between the values of objective functions for every solution and optimal values of particular objective functions. Optimal value  $R(x)$  is counted as:

$$R(x) = \left| \sum_{\alpha=1}^k (J_{\alpha}(x) - J_{\alpha}(x_{\alpha})) \right| \quad (7)$$

$$\text{opt } R(x) = \min_{x \in \{X\}} \left( \sum_{\alpha=1}^k (J_{\alpha}(x) - J_{\alpha}(x_{\alpha})) \right) \quad (8)$$

### **Generalized Metric**

The formula for finding the optimal value  $R_L(x)$  is given as

$$R_L(x) = \sum_{\alpha=1}^k \left\{ (J_{\alpha}(x) - J_{\alpha}(x_{\alpha}))^L \right\}^{\frac{1}{L}} \quad (9)$$

#### **2.1.2 Compromising Methods**

This group of metrics is based on adding the weights of optimized objectives into the optimization process. The search for the optimal solution of the MOO task is realized by minimizing the function

$$\beta_1 J_1(x) + \beta_2 J_2(x) + \dots + \beta_k J_k(x) \quad (10)$$

where  $\beta_1, \dots, \beta_k$  are weight coefficients. It is recommended to set their values to  $\beta_1=1/J_{10}, \beta_2=1/J_{20}, \dots, \beta_k=1/J_{k0}$ , where  $J_{k0}$  are values reached by optimization of  $k^{\text{th}}$  criterion. These metrics are used, if the decision maker is able to define the importance of optimized criteria before the start of optimization process. Weights of particular optimized criterion will be labelled as  $\lambda$ . [16]

### ***Weighted Sum of Values of Objective Functions***

The metric is defined as

$$\text{opt } R(x) = \text{opt} \sum_{\alpha=1}^k \lambda_{\alpha} J_{\alpha}(x) \quad (11)$$

Optimal solution is represented as the maximum or minimum of the weighted sum of values of all objective functions.

### ***Weighted Sum of Deviations***

Using this metric, the formula for calculating the  $R(x)$  parameter is defined as:

$$R(x) = \sum_{\alpha=1}^k \lambda_{\alpha} * (J_{\alpha}(x) - J_{\alpha}(x_{\alpha})) \quad (12)$$

while the optimal solution is the minimum of the  $R(x)$ :

$$\text{opt } R(x) = \min_{x \in \{X\}} \left( \sum_{\alpha=1}^k \lambda_{\alpha} * (J_{\alpha}(x) - J_{\alpha}(x_{\alpha})) \right) \quad (13)$$

Some other mathematical metrics with their description and usage can be found in [7] and [30].

## **2.2 Methods of Artificial Intelligence**

When solving multi-objective optimization problems, an evaluation function is used which returns a real number representing the suitability of the solution. The higher the value, the better the solution. This function corresponds to the objective function of mathematical methods. This function can represent a number of criteria, which are frequently in conflict. In this case, the goal is to find the Pareto-optimal front, which consists of a set of non-dominated solutions.

Basic evolutionary algorithm process is shown in Fig. 2.

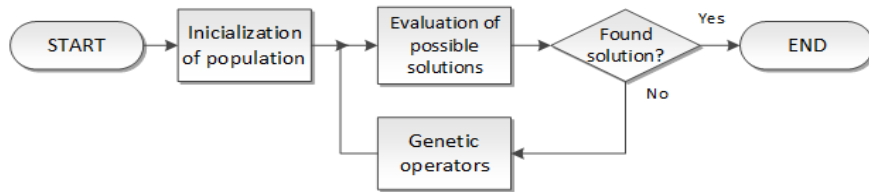


Figure 2

Fundamental evolutionary algorithm scheme

In this part of paper, NSGA-II algorithm is presented as algorithm chosen for defining the set of non-dominated solutions. This genetic algorithm was developed as an improved version of the NSGA genetic algorithm. The algorithm, compared

to its ancestor, is characterized by lower computational demands, better convergence and diversity. These improvements are reached by non-dominant sorting, elitism and crowding distance operations. [30]

### 2.2.1 Non-dominated Sorting

Non-dominated sorting is an operation sorting the chromosomes from the population into non-dominant fronts. The non-dominant front is a set of chromosomes that do not dominate each other. The dominance of the chromosome  $x$  above the chromosome  $y$  occurs when the value of all evaluation functions for the chromosome  $x$  is better than for the chromosome  $y$ . In the original NSGA algorithm, the sorting method was implemented in a way that all possible pairs of chromosomes were compared and the first non-dominant queue was found. These chromosomes were excluded from the population and the whole process was repeated without them. However, this method is computationally demanding. In the NSGA-II algorithm, queuing is accomplished by fast, non-dominant sorting - FNDS (closer description of the iterational process of FNDS can be found in [30]). Result of the process of FNDS is shown on Fig. 3.

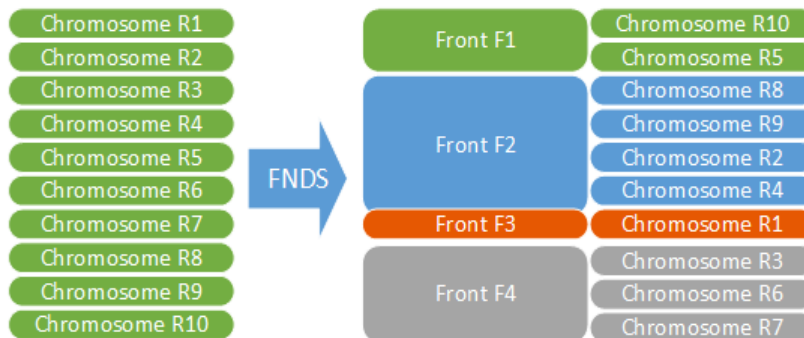


Figure 3

Diagram showing the results of fast non-dominated sorting

### 2.2.2 Crowding Distance

In the NSGA-II algorithm, the crowding distance is used to compare the chromosomes within one front. Crowding distance sorts the chromosomes according to their diversity (the chromosome most different from the others is considered the best). Procedure for implementing this part of the algorithm:

For each queue with the number of individuals  $n$ , individuals of every front are sorted according to the value of the  $m^{\text{th}}$  objective function

$$I = \text{sort}(F_i, m) \quad (14)$$

An infinite distance is assigned to the boundary chromosomes (first and last chromosome according to the sorting):

$$I(d_1) = \infty; I(d_n) = \infty \tag{15}$$

For other individuals ( $k = 2, 3, \dots, n-1$ ) the following formula is used:

$$I(d_k) = I(d_k) + \frac{I(k+1)m - I(k-1)m}{f_m^{max} - f_m^{min}} \tag{16}$$

, where  $I(k)m$  is the value of the  $m^{th}$  objective function of the individuals in sorting I. The metric by which chromosomes are organized, is defined as the sum of the chromosome's distances from the next chromosomes within the queue. Crowding distance is used in selecting chromosomes into a new generation, preferring chromosomes with the highest value of crowding distance. Results of applying the crowding distance are shown in Fig. 4.

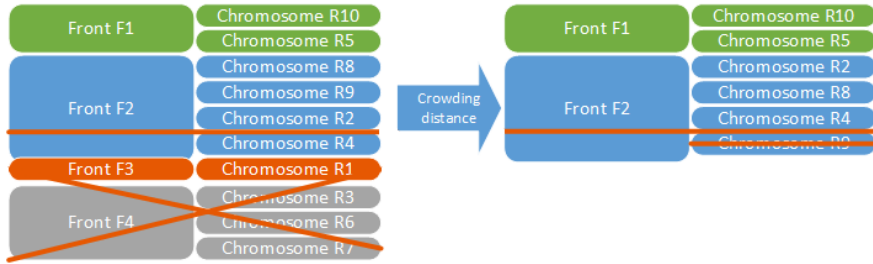


Figure 4

Diagram showing the results of rowding distance

### 2.2.3 Elitism

The principle of elitism keeps the chromosomes with the best results in the iteration process. The new generation is created by operations of crossing and mutation. Formulas used within the process of crossing and mutation are closely described in [6] and [26].

### 2.2.4 Iterative Process

The iterative process of the NSGA-II genetic algorithm can be described in the following steps:

1. Half of the population from the first iteration is generated with random genes, the other half is generated from the first iteration using crossing and mutation.
2. Chromosomes from the new population are sorted into non-dominant fronts by fast non-dominated sorting.



3. For the next iteration, half of the chromosomes with the best results are selected.
4. The second half of the population for the next iteration is generated in crossover and mutation operations.

The population is then again ranked and sorted etc. [12]

### 2.2.5 Modifications Realized in NSGA-II Algorithm

Fig. 5 shows the computational process of the modified NSGA-II algorithm (blue blocks represent the modified parts in comparison to the original NSGA-II computational process).

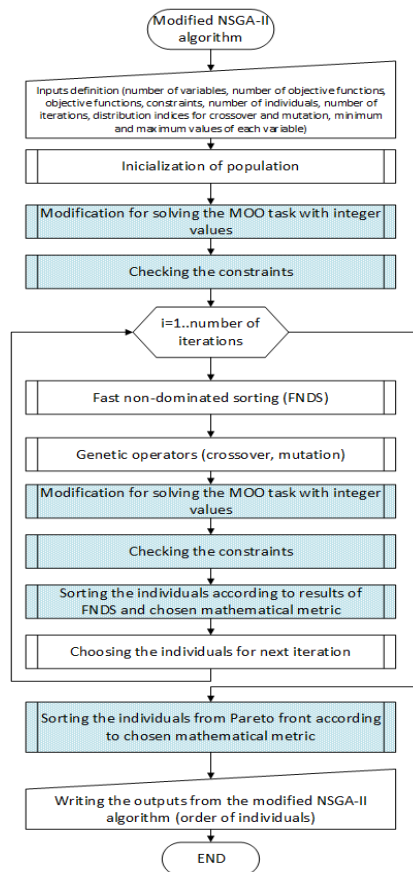


Figure 5  
Algorithm of modified NSGA-II

For the proper functionality of the algorithm, several modifications were made to the computational process described in [13]. Modifications were made in order to modify the computational process of the NSGA-II algorithm for requirements of the resolved task:

- the algorithm considers the values of selected mathematical method as a secondary sorting criterion, instead of original criterion – crowding distance;
- it is necessary for the algorithm to work with integer values, since we are looking for number of products;
- the algorithm must be able to control the fulfillment of all defined constraints.

### 3 Solving MOO Problem of Assembly Line

This part of the paper is focused on solving the task of defining the optimal production process of one of the assembly line models at the Department of Cybernetics and Artificial Intelligence (DCAI) FEEI TUKE. Schematic view of the assembly line model is shown in Fig. 1.

#### 3.1 Definition of MOO Task

The production line from Fig. 1 (closer description in [8]), is going to produce 4 different product types (mosaic) made from 4 different types of colored squared pieces: blue, white, green and black (Fig. 1).

The number of individual types of square pieces needed to manufacture the product is shown in Table 1, as well as the capacity of particular parts for one production cycle of this production line.

Table 1  
Number of square pieces needed for manufacturing the products and their capacity

Part/Product	A	B	C	D	Capacity
Blue	5	7	7	3	100
White	3	4	2	4	80
Green	4	6	3	2	90
Black	5	5	6	6	120

The profit obtained from each blue part contained in the mosaic is € 3, of a white cube it is € 5, profit of using every green part is a € 4 and profit from every black part is € 2. Production time also depends on the number of parts included in the mosaic (placing one part of the product lasts 1,6 seconds). In a single production cycle, a maximum of 24 products can be produced, because only 24 products can

be stored simultaneously. The purpose of this MOO task is to define the number of products manufactured during the production process, in order to fulfill all constraints and to maximize both of the objective functions (profit from the production process as well as the saved time during the production process).

### 3.2 Definition of Objective Functions

As mentioned in the MOO task, the goal is to maximize profit as well as to maximize time savings. For the purpose of calculating the profit function, it is necessary to calculate the profit for each product (formula (17)).

$$\sum_{j=1}^{pv} z_i p k_{ij}, \text{ for } i = 1, 2, 3, 4. \quad (17)$$

where  $z_i$  is the profit from  $i^{\text{th}}$  part,  $p k_{ij}$  is the number of parts of  $i^{\text{th}}$  type in  $j^{\text{th}}$  product and  $pv$  is the number of products. Using this formula, the profit from each type of product is obtained:

Table 2  
Profit from manufacturing products

Product	A	B	C	D
Profit	56 €	75 €	55 €	49 €

From these values we can then define the objective function to maximize profit as

$$U_1(\mathbf{x}) = 56x_1 + 75x_2 + 55x_3 + 49x_4 \rightarrow \max \quad (18)$$

For the second objective function, we need to know the value of the time that is saved by producing this product, compared to producing the product with the maximum number of parts. It is not possible to define this objective function as a minimization of production time, since the ideal value would be 0 (doing nothing), which would affect the results in an undesirable manner. Since the templates for mosaic production have a size of 5 rows with 5 columns, one product can contain a maximum of 25 parts

$$\sum_{j=1}^{pv} p k_{ij}, \text{ for } i = 1, 2, 3, 4. \quad (19)$$

In (30),  $p k_{ij}$  the number of parts of the  $i^{\text{th}}$  type in the  $j^{\text{th}}$  product and  $pv$  is the number of products, we calculate the total number of parts used for each type of product. These values are listed in Table 3.

Table 3  
Number of parts used in manufacturing products

Product	A	B	C	D
Parts	17	22	18	15

The time savings obtained from the production of the product ( $co_j$ ) when compared to the product consisting of the maximum number of parts (25) can be defined by formula (20). The obtained values are written in Table 4:

$$co_j = 1,6 (25 - pk_j) \quad (20)$$

Table 4  
Amount of saved time by manufacturing products

Product	A	B	C	D
Saved time	12,8 s	4,8 s	11,2 s	16 s

Based on these values, we can define a second objective function as:

$$U_2(\mathbf{x}) = 12,8x_1 + 4,8x_2 + 11,2x_3 + 16x_4 \rightarrow \max \quad (21)$$

### 3.3 Definition of Constraints

To define the constraints, we start from the Table 1, which lists the number of parts needed to produce each type of product, as well their capacity. Based on this table, we can define the following constraints for this task:

$$5x_1 + 7x_2 + 7x_3 + 3x_4 \leq 100 \quad (22)$$

$$3x_1 + 4x_2 + 2x_3 + 4x_4 \leq 80 \quad (23)$$

$$4x_1 + 6x_2 + 3x_3 + 2x_4 \leq 90 \quad (24)$$

$$5x_1 + 5x_2 + 6x_3 + 6x_4 \leq 120 \quad (25)$$

Since the task is focused on one production cycle of the production line and the number of storage spaces is limited to 24, the number of products produced must not exceed this value:

$$x_1 + x_2 + x_3 + x_4 \leq 24 \quad (26)$$

The last constraint results from the logical assumption that the number of each of produced products can not be negative:

$$x_1, x_2, x_3, x_4 \geq 0 \quad (27)$$

### 3.4 Definition of Parameters for Iteration Process

To identify the Pareto front, we chose the NSGA-II algorithm, which was modified in order to deal with this MOO task. After defining of the Pareto front, the results will be sorted according to some of the mathematical methods in order to find the solution of the MOO task. Different mathematical metrics were used to

resolve the task and compare results, namely quadratic metric in a dimensionless form (metric 1), quadratic metric (metric 2 – chapter 2.1.1), weighted quadratic metric ( $w=[0.8;0.2]$ ) (metric 3 – chapter 2.1.2), weighted linear metric ( $w=[0.2;0.8]$ ) (metric 4 - chapter 2.1.2), linear metric (metric 5 - chapter 2.1.1) and percentual fulfillment of each objective function (metric 6 – chapter 3.5.2).

The NSGA - II algorithm was configured to process 2 objective functions with 4 variables, every population consisted of 200 individuals. 50 iterations were run with the crossing and mutation distribution parameter equaling 0. Vector of minimum parameter values was defined as  $[0,0,0,0]$  (minimum number of products) and vector of maximum parameter values was set as  $[24,24,24,24]$  (maximum number of products from each type).

## 3.5 Solving the MOO Task

### 3.5.1 Searching the Pareto Front Solutions

After the computational process of the modified NSGA – II algorithm was implemented in MATLAB, all possible MOO solutions were found, sorted according to their membership to front. To find the solution, the first (Pareto) front is important. The Pareto front contains non-dominated solutions (solutions that are not inferior in both objective functions than any other solution). Therefore, we will choose the solutions from the Pareto front (Table 6). On the Fig. 6 ,objective values are negative because the algorithms was built to minimize the objective functions. This is why the functions were multiplied by -1. Pareto front members are shown by red dots, chromosomes from the second front are represented as green dots, third front has blue dots and other fronts are displayed by black dots.

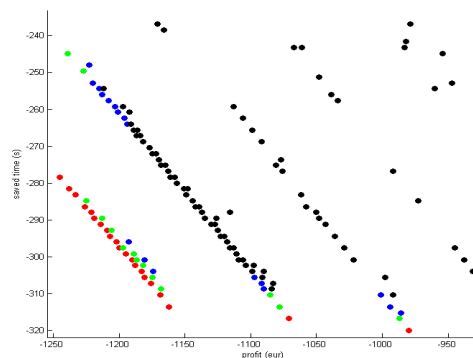


Figure 6

Graph showing the individuals from different non-dominant fronts

### 3.5.2 Application of Mathematical Metrics

Out of the individuals belonging to the Pareto front (listed in Table 6), we looked for the best possible solution of the MOO task. The solution using the percentual fulfillment of every objective function is shown closely. To calculate the percentual fulfillment of every objective function, we needed to know the maximum value of each of the objective functions. Values were calculated in the iterative process of the modified NSGA-II algorithm.

Table 5  
Extreme values of objective functions

extr $U_1(x)$	extr $U_2(x)$
-1245	-320

For each individual from the Pareto front, the value of the percentual deviation was calculated from the ideal values and was realized using the formula

$$\delta U_i(\mathbf{x}) = \frac{\left| \min U_i(\mathbf{x}) - |U_i(\mathbf{x})| \right|}{\left| \min U_i(\mathbf{x}) \right|} \quad (28)$$

The sum of these deviations was then calculated according to ( $p$  is the number of objective functions):

$$\sum \delta U_i(\mathbf{x}) = \sum_{i=1}^p \frac{\left| \min U_i(\mathbf{x}) - |U_i(\mathbf{x})| \right|}{\left| \min U_i(\mathbf{x}) \right|} \quad (29)$$

The individual belonging to the Pareto front with the lowest value of the parameter  $\sum \delta U_i(\mathbf{x})$  was chosen as the solution of the MOO task. In Table 6, Pareto front members are listed according to the parameter  $\sum \delta U_i(\mathbf{x})$ .

### 3.5.3 Interpretation of Results

The solution of the presented MOO task is represented by vector  $\mathbf{x} = [12,0,0,10]$ . This means that, according to the percentual fulfillment of each of the objective functions, the production line would have to produce 12 products of type A and 10 products of type D. The profit from one production cycle would be € 1162 and a time saved compared to the production of products with 25 parts would be 313,6 seconds. The deviation from the extreme values represents 6.67% from the maximum profit amount and 2% of the maximum saved time. Therefore, the deviation from the maximum values of the objective functions is 8.67%.

Table 6 shows that vector  $[12,0,0,10]$  has the lowest value of deciding parameter. Therefore, it represents an optimal solution according to the chosen metric.

Table 6  
Pareto optimal solutions ordered by percental fulfillment of objective functions

$x_1$	$x_2$	$x_3$	$x_4$	$U_1(x)$	$U_2(x)$	$\delta U_1(x)$	$\delta U_2(x)$	$\sum \delta U_i(x)$
12	0	0	10	1162	313,6	0,066667	0,02	0,086667
13	0	0	9	1169	310,4	0,061044	0,03	0,091044
14	0	0	8	1176	307,2	0,055422	0,04	0,095422
11	1	0	10	1181	305,6	0,051406	0,045	0,096406
15	0	0	7	1183	304	0,049799	0,05	0,099799
12	1	0	9	1188	302,4	0,045783	0,055	0,100783
16	0	0	6	1190	300,8	0,044177	0,06	0,104177
13	1	0	8	1195	299,2	0,040161	0,065	0,105161
10	2	0	10	1200	297,6	0,036145	0,07	0,106145
14	1	0	7	1202	296	0,034538	0,075	0,109538
11	2	0	9	1207	294,4	0,030522	0,08	0,110522
15	1	0	6	1209	292,8	0,028916	0,085	0,113916
12	2	0	8	1214	291,2	0,0249	0,09	0,1149
9	3	0	10	1219	289,6	0,020884	0,095	0,115884
13	2	0	7	1221	288	0,019277	0,1	0,119277
10	3	0	9	1226	286,4	0,015261	0,105	0,120261
11	3	0	8	1233	283,2	0,009639	0,115	0,124639
8	4	0	10	1238	281,6	0,005622	0,12	0,125622
9	4	0	9	1245	278,4	0	0,13	0,13
6	0	0	15	1071	316,8	0,139759	0,01	0,149759
0	0	0	20	980	320	0,212851	0	0,212851

### 3.5.4 Synthesis of Results Obtained by Using other Mathematical Metrics

In Table 7 it can be seen that the choice of the best alternative depends on the chosen metric. Only the best solution for a particular MOO metric is displayed (M stands for metric and numbers corresponds with metrics mentioned in 3.4). It can be seen that the choice of the best alternative depends on the chosen metric.

Table 7  
Best solutions using various mathematical metrics

M	$x_1$	$x_2$	$x_3$	$x_4$	$U_1(x)$	$U_2(x)$	$\delta U_1(x)$	$\delta U_2(x)$	$\sum \delta U_i(x)$
1	13	0	0	9	1169	310,4	0,0037	0,0009	0,0046
2	13	2	0	7	1221	288	576	1024	1600
3	11	2	0	9	1207	294,4	0,0007	0,0012	0,0019
4	12	0	0	10	1162	313,6	16,6	5,12	21,72
5	9	4	0	9	1245	278,4	0	41,6	41,6
6	12	0	0	10	1162	313,6	0,066667	0,02	0,086667

One way to realize the synthesis of the results of different MOO metrics is to sort out the alternatives according to their standings in the optimization process

realized by every one of the metrics (the best alternative for each of the metrics obtains 1 point, the worst obtains  $pv$  points) and realize the subsequent synthesis of this score by formula

$$pb_i = \sum_{j=1}^{pn} b_{ij}, \text{ for } i = 1, 2, \dots, pv \tag{30}$$

where  $pb_i$  is the result of alternative  $I$ ,  $b_{ij}$  is the score of alternative  $I$  using the  $j^{th}$  metric,  $pn$  is the number of metrics and  $pv$  is the number of alternatives. The results of Pareto front solutions according to this synthesis of results is in Table 8:

Table 8  
Synthesis of solutions using various mathematical metrics

i	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	b <sub>i1</sub>	b <sub>i2</sub>	b <sub>i3</sub>	b <sub>i4</sub>	b <sub>i5</sub>	b <sub>i6</sub>	pb <sub>i</sub>
1	10	2	0	10	9	11	1	9	11	9	50
2	11	2	0	9	11	9	2	11	9	11	53
3	14	1	0	7	10	10	3	10	10	10	53
4	13	1	0	8	8	12	6	8	12	8	54
5	16	0	0	6	7	13	8	7	13	7	55
6	12	1	0	9	6	14	9	6	14	6	55
7	11	1	0	10	2	16	13	4	16	4	55
8	15	1	0	6	12	8	4	12	8	12	56
9	15	0	0	7	5	15	12	5	15	5	57
10	12	2	0	8	13	7	5	13	7	13	58
11	14	0	0	8	3	17	15	3	17	3	58
12	13	0	0	9	1	18	17	2	18	2	58
13	9	3	0	10	14	5	7	14	6	14	60
14	12	0	0	10	4	19	19	1	19	1	63
15	10	3	0	9	16	1	11	16	4	16	64
16	13	2	0	7	15	4	10	15	5	15	64
17	11	3	0	8	17	2	14	17	3	17	70
18	8	4	0	10	18	3	16	18	2	18	75
19	9	4	0	9	19	6	18	19	1	19	82
20	6	0	0	15	20	20	20	20	20	20	120
21	0	0	0	20	21	21	21	21	21	21	126

As can be seen in the Table 8 and Fig. 7, according to synthesis of the results, the option  $\mathbf{x}=[10,2,0,10]$  seems to be the best solution, followed by vectors  $\mathbf{x}=[11,2,0,9]$  and  $\mathbf{x}=[14,1,0,7]$ . Winning solution from the percentual fulfillment of the objective functions (metric 6),  $\mathbf{x}=[12,0,0,10]$  is only the 14<sup>th</sup> best option according to the synthesis of metrics. This fact can be considered as the proof that using more than one MOO method can result in defining solution, which is more complex when compared to the solution found using only one method.



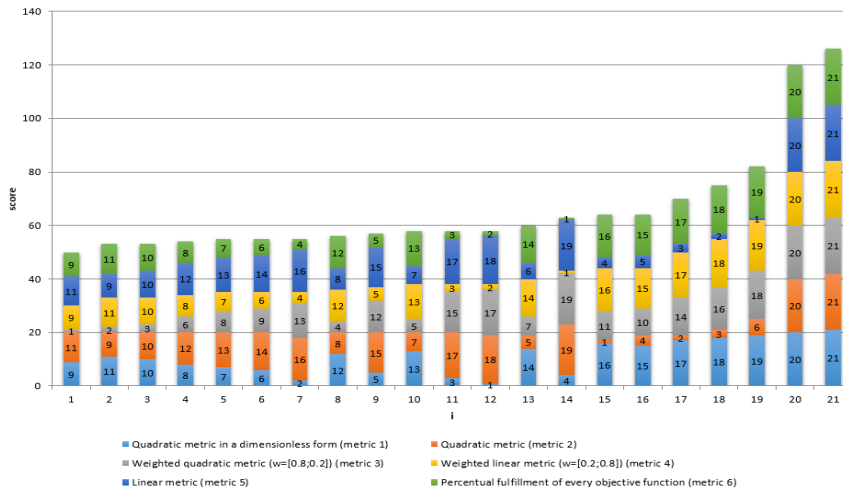


Figure 7

Graph of final score of alternatives using the synthesis of various mathematical metrics

On Fig. 8, user interface developed within our research for resolving multiobjective optimization tasks using combination of modified NSGA – II algorithm and various mathematical methods, is presented.

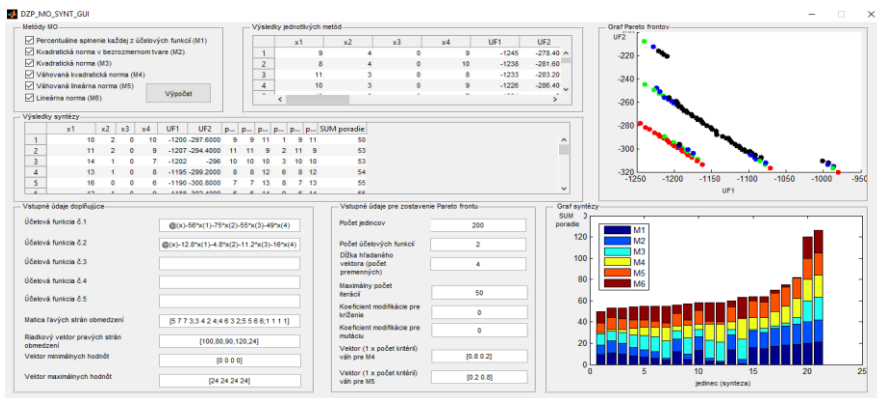


Figure 8

User interface for solving MOO tasks

### Conclusions

In the presented paper, multi objective optimization task of definition of the optimal production process of the assembly line model within Center of Modern Control Techniques and Industrial Informatics (CMCT&II) within the Department of Cybernetics and Artificial Intelligence of FEEI TUKE. For resolving the task of choosing the optimal production strategy of the automated assembly line,

combination of modified evolutionary algorithm NSGA-II and various mathematical metrics was used. In defining the set of Pareto optimal solutions, some modifications had to be realized in order to improve the computational process. From the Pareto optimal solutions, the best option for production process was chosen by the percentual fulfilment of every objective function. Another presented option involved the synthesis of the solutions from different mathematical metrics.

In conclusion, the decision for choosing the metric used to define the order of the alternatives from the Pareto set should be realized with respect to the preferences of the decision-maker. Moreover, not all possible approaches can be used to solve a particular assembly line balancing problem. A survey of problems and applicable methods in this area can be found in [4]. Approaches focused on dealing with mixed-model assembly lines, which was also the model described in this paper, are available in [5]. Weighted objective functions can be used, if the decision-maker prefers one of the objectives over the others. Some of the methods for objectivization of the defined weights of objectives can be found in [9].

During the research in this area, we were focused on the optimization of the assembly lines, particularly on creating the simulation models of assembly lines with a focus on their time optimization [10], as well as solving the task of optimal assembly line configuration using the methods of the multi-criterial decision-making [9]. Results obtained within the research are described in [6]. The MOO of the production process of the assembly line, which is closely described within this paper, is another task resolved in the assembly line optimization area, which contributes to the portfolio of problems resolved in this area within our department.

### **Acknowledgement**

This work has been supported by grant KEGA Implementation of research results in the area of modelling and simulation of cyber-physical systems into the teaching process - development of modern university textbooks – 072TUKE-4/2018 (80%). It was supported as well by the Research and Development Operational Program for project: University Science Park Technicom for innovative applications with knowledge technology support – 2nd phase (20%), ITMS code 313011D232.

### **References**

- [1] A. Abraham and L. Jain. Evolutionary multiobjective optimization, Springer, London, 2005
- [2] J. Bautista, and J. Pereira. Ant algorithms for a time and space constrained assembly line balancing problem. European journal of operational research, 177.3, 2016-2032, 2007

- 
- [3] N. Boysen, M. Fliedner and A. Scholl. A classification of assembly line balancing problems. *European journal of operational research*, 183.2, 674-693, 2007
- [4] N. Boysen, M. Fliedner and A. Scholl. Assembly line balancing: Which model to use when?. *International Journal of Production Economics*, 111.2, 509-528, 2008
- [5] N. Boysen, M. Fliedner and A. Scholl. Sequencing mixed-model assembly lines: Survey, classification and model critique. *European Journal of Operational Research*, 192.2, 349-373, 2009
- [6] J. Čabala. Creation of Methodology of Multicriterial Assembly Lines Optimization, Dissertation Thesis. CMCT&II DCAI FEEI TU Košice, 163 p., 2018 (in Slovak), supervisor: doc.Ing. J.Jadlovský, CSc.
- [7] J. Čabala and J. Jadlovský. Solving vector optimization problems using Optimization Toolbox. 22<sup>nd</sup> Annual Conference Proceedings of the International Scientific Conference - Technical Computing Bratislava 2014, published online, November 2014
- [8] J. Čabala, J. Jadlovský, and K. Dujavová. Realization of production process information system on RDBS Oracle base, *Electrical Engineering and Informatics: proceedings of the FEI of Technical University of Košice*, 6, 791-796, 2015
- [9] J. Čabala and J. Jadlovský. Solving optimal assembly line configuration task by multi-objective decision making methods, *Acta Electrotechnica et Informatica*, 2, 53-60, 2017
- [10] J. Čabala and J. Jadlovský. Application of Stateflow Diagrams in Production Line Modeling Danvers : IEEE, SAMI 2016 (Herľany, Slovakia), 125-130, 2016
- [11] Y. K. Choi, D. M. Lee and Y. B. Cho. An approach to multi-criteria assembly sequence planning using genetic algorithms. *The International Journal of Advanced Manufacturing Technology* 42. 1-2, 180-188, 2009
- [12] K. Deb, and S. Karthik. Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling." *International conference on evolutionary multi-criterion optimization*. Springer, Berlin, Heidelberg, 803-817, 2007
- [13] K. Deb, S. Agarwal, A. Pratap, T. Meyarivan. A fast elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: *Proceedings of the parallel problem solving from nature VI conference*, Paris, France, 849-858, 2000
- [14] M. Farina, K. Deb, and P. Amato. Dynamic multiobjective optimization problems: test cases, approximations, and applications. *IEEE Transactions on evolutionary computation* 8.5, 425-442, 2004
- [15] M. Fonseca, and P. J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization. *Icga*, 93, 416-423, 1993

- [16] K. Hrubina, A. Jadlovská and S. Hrehová: Algorithms of optimization methods with using programming systems. Prešov – Košice, Vol. 1, 393 p., 2005
- [17] S. D. Lapierre, A. Ruiz, and P. Soriano. Balancing assembly lines with tabu search. *European Journal of Operational Research*, 168.3, 826-837, 2006
- [18] H. Li, and Q. Zhang. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on evolutionary computation*, 13.2, 284-302, 2009
- [19] S. A. Mansouri. A multi-objective genetic algorithm for mixed-model sequencing on JIT assembly lines. *European Journal of Operational Research* 167.3, 696-716, 2005
- [20] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization* 26.6, 369-395, 2004
- [21] M. Pelikan. Evolutionary algorithms. *Artificial intelligence and cognitive science I*, 335-353, 2009 (in Czech)
- [22] R. E. Precup, and R. C. David. *Nature-Inspired Optimization Algorithms for Fuzzy Controlled Servo Systems*. Butterworth-Heinemann, 2019
- [23] C. Purcaru, et al. Optimal robot path planning using gravitational search algorithm. *International Journal of Artificial Intelligence* 10, S13, 1-20, 2013
- [24] D. Rosinová, and M. Dúbravská. *Optimization*. Bratislava: STU, 1<sup>st</sup> ed., 190 p., 2008 (in Slovak)
- [25] I. Sabuncuoglu, E. Erdal and A. Arda. Ant colony optimization for the single model U-type assembly line balancing problem. *International Journal of Production Economics* 120.2, 287-300, 2009
- [26] A. Seshadri. A fast elitist multiobjective genetic algorithm: NSGA-II, *Matlab Central, file exchange, mathworks*, 2007
- [27] M. Shams, et al. Ideal gas optimization algorithm. *International Journal of Artificial Intelligence* 15.2, 116-130, 2007
- [28] H. A. Taboada, F. Baheranwala, D. W. Coit, and N. Wattanapongsakorn. Practical solutions for multi-objective optimization: An application to system reliability design problems. *Reliability Engineering & System Safety*, 92(3), 314-322, 2007
- [29] J. Vaščák. Adaptation of fuzzy cognitive maps by migration algorithms. *Kybernetes* 41.3/4, 429-443, 2012
- [30] T. Weise. *Global optimization algorithms-theory and Application*. Self-Published, 2009
- [31] E. Zitzler. Evolutionary algorithms for multiobjective optimization, *CIMNE*, 19-26, 2002