*Production Systems and Information Engineering*
Volume 4 (2006), pp. 125-145


# COMPARISON OF DIRECT AND INDIRECT DISTRIBUTION OF A NETWORK-LIKE OPERATING LOGISTICS INTEGRATED ASSEMBLY SYSTEM

BÉLA OLÁH
University of Miskolc, Hungary
Department of Materials Handling and Logistics
altbela@uni-miskolc.hu

TAMÁS BÁNYAI
University of Miskolc, Hungary
Department of Materials Handling and Logistics
alttamas@uni-miskolc.hu

JÓZSEF CSELÉNYI
University of Miskolc, Hungary
Department of Materials Handling and Logistics
cselenyi@snowwhite.alt.uni-miskolc.hu

[Received October 2005 and accepted June 2006]

**Abstract.** This paper introduces the limitations and objective functions of planning of a network-like operating logistics integrated assembly systems. The optimal assignment of assembly plants to the final product requirements of the end users is discussed in detail, and related cost functions are worked out. Solution methods of optimisation are described in the next chapters. The sensitivity analysis of the assignment algorithms concerning to products and assembly plants is completed by a simple example and comparison of different variations is showed. Finally the system of one distribution warehouse model and the description of its algorithm is showed. Novelty of this paper appears in its network-like nature, in its methods of optimising as well as in its approach to the objective functions, especially to the cost functions.

*Keywords*: assignment, logistics, optimisation, assembly system

## 1. Introduction

The network-like operating logistics integrated assembly system means when the production planning is planned integrated by the purchasing and distribution logistics system, accordingly we search aggregate optimum of not merely the production but also the logistics resources and factors. The network-like means that the same product can be assembled by several assembly plants in different points, and the components needful to assembling can be purchased from several different sited suppliers. Additionally the network-like means that the procurement of components and the distribution of final products may be direct and indirect, in other words by the help of distribution warehouse. In case of the network-like operating systems the logistics integrated production planning details how search the optimal result having regard to capacity-limits and conditions, fulfill to the requirements of the end users according to described objective functions.

Mathematical modelling and optimisation of a network-like operating assembly system as an integrated logistics system there is no even early attempt in the international scientific literature. A mere scattering of publications for the logistics integrated production scheduling can be found in the international literature [9, 10, 11, 12]. Therefrom can be determined that the used objective functions and conditions are in accord with the used by us. At the same time in the system drawn by us the objective functions and the conditions can be demonstrated in an other form, it follows from this, that this defined principle can be used for the optimisation. All these require that we lean on the considerable results made in the Department of Materials Handling and Logistics at the University of Miskolc for solution of this logistics model [1, 4, 13, 14].

The globalisation and the decreasing of the production depth leaded to sweeping changes of the market of firms. These changes can be recognised by the increasing of cooperative industrial processes based on horizontal and vertical networking depending on the depth of competitiveness [7]. In the model (Fig. 1) the amount of the final products ordered by an end user in a given time interval is given.

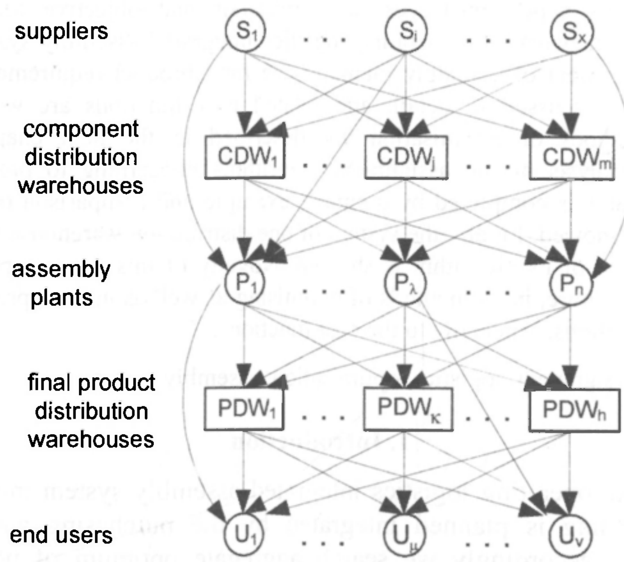



**Figure 1.** Network-like operating assembly system

The optimal operation of this complex and large cooperative logistics system requires absolutely modern theoretical establishment of planning and control methods [1]. The task to be completed is the logistics integrated assignment task, which includes the distribution and storage of final products and the storage of components. Different objective functions and conditions must and can be taken into consideration during the solution of these tasks. In the first case the cost function is chosen as objective function, whose components are detailed in [3]. The optimisation is completed by a hierarchical jointed feedback heuristic method due to the high number of cost function parameters to be optimised. The modules of a multistage optimisation and the principles [4], solution methods and heuristic algorithm of the assignment are elaborated in [5].

This system consists of the following units: assembly plants ($P_\lambda$), component distribution warehouses ($CDW_j$) and final product distribution warehouses ($PDW_\kappa$). Transports of component parts can take place into the above mentioned assembly plants: indirectly, i.e. through component distribution warehouses, or by direct transports from the suppliers ($S_i$) in case of bypassing the distribution warehouse. Transports of final products occur to the end users ($U_\mu$) in like manner, i.e. direct delivery from the assembly plants or indirectly through final product distribution warehouses.

The authors worked out the assignment of final products requirements of separate end users to the assembly plants relating to time-intervals $\vartheta$ with the following simplyfied coditions in paper [3]:

- we have taken no notice of the preparation cost separate from the assembly cost, in this planning level not come to optimise the assembly lot size;
- the specific assembly cost is conditioned by only final product type;
- only the direct distribution model is analysed by the help of shuttle tours.

## 2. Aims of the paper

This paper shows the followings:

- assignment algorithms of the before-mentioned simplyfied network-like operating system are stated;
- exact data-model is elaborated, by the help of it three optimal assignment variations are worked out;
- the optimum sensitivity analysis are accomplished, the specific assembly cost is conditioned by the assembly plants per final products;
- that model and algorithms are worked out which is suited to sort out the better from the direct (without distribution warehouse) and indirect distribution.

In any case only the distribution with shuttle tours is analysed, but it can be under further investigation, what cost reduction can be achieved with the solution of distribution tasks by the help of round tours.

### 2.1. The total cost function of the model

$$C = C^P + C^T + C^S + C^A + C^{RT} + C^{\Psi} + C^W + C^D \rightarrow min.\ [€] \qquad (2.1)$$

which can be obtained as a sum of the following costs: purchase costs of components ($C^P$), transportation costs of components ($C^T$), storage costs of components ($C^S$), assembly costs ($C^A$), changeover costs of assembly lines ($C^{RT}$), costs of standby of lines ($C^{\Psi}$), warehousing charges of final products ($C^W$), and distribution costs of final products ($C^D$).

We simplify the total cost function (2.1) for the determination of the annual amount of the final products of the individual user and then only the distributional and assembly costs should be considered. Because this paper considers the optimisation of the assembly and the delivery, the warehousing costs of components and final products cannot be taken into consideration and the considered costs are also global and simplified. The above-mentioned cost-components have not to be taken into account by optimisation, because these components are not known by this step of assignment, but we will take these into account in after modules and effects of these components will appear from the principle of feedback.

### 2.2. The objective function of the assignment in case of the product k

$$C_k^I = C_k^A + C_k^D \rightarrow min.\ [€] \qquad (2.2)$$

where $C_k^A$ is the assembly cost, $C_k^D$ is the distribution cost. The matrix Q gives the annual quantity ordered from product $k$ by the user $\mu$. The searched matrix Y shows that the user μ gets the final product $k$ from the assembly plant $\lambda$ or not. $y_{k\mu\lambda}$ can take values 0 or 1 with the following condition: $\sum_{\lambda=1}^{n} y_{k\mu\lambda} = 1$ (case a), or what part of the final product $k$ will be transported into the end user μ from the assembly plant λ (case b). Conditions are: $0 \le y_{k\mu\lambda} \le 1$ and $\sum_{\lambda=1}^{n} y_{k\mu\lambda} = 1$

### 2.3. The considered and simplified objective functions in case of the product k

#### *2.3.1. Distribution cost*

$$C_k^D = \sum_{\lambda=1}^{n} \sum_{\mu=1}^{\nu} c_k^D Q_{k\mu} y_{k\mu\lambda} s_{\mu\lambda} \quad [€] \qquad (2.3)$$

$c_k^D$ is the specific delivery cost of final product $k$, $s_{\mu\lambda}$ is the length of the delivery route between user $\mu$ and assembly plant $\lambda$.

*2.3.2. Assembly cost*

$$C_k^A = \sum_{\lambda=1}^{n}\sum_{\mu=1}^{\nu} Q_{k\mu} y_{k\mu\lambda} c_{k\lambda}^A \ [€] \tag{2.4}$$

where $c_{k\lambda}^A$ is the specific assembly cost in case of the product $k$ in the plant $\lambda$. The assembly cost by the plants has to be calculated as the weighted average cost of the capacity of assembly lines because we do not know yet onto which line will be assembled it. As well the lower $L^L = \left[\ell_{k\lambda}^L\right]$ and the upper limit $L^U = \left[\ell_{k\lambda}^U\right]$ of annual produced amount of every assembly plant have to be defined. An exclusion matrix has to be defined which gives which plant which product does not able to assemble. Conditions are

$$\ell_{k\lambda}^L \le \sum_{\mu=1}^{\nu} Q_{k\mu} y_{k\mu\lambda} \le \ell_{k\lambda}^U . \tag{2.5}$$

*2.3.3. Objective function*

The objective function (3.1) becomes the following formula by the considered and simplified objective functions:

$$C_k^I = \sum_{\lambda=1}^{n}\sum_{\mu=1}^{\nu} c_k^D Q_{k\mu} y_{k\mu\lambda} s_{\mu\lambda} + \sum_{\lambda=1}^{n}\sum_{\mu=1}^{\nu} Q_{k\mu} y_{k\mu\lambda} c_{k\lambda}^A \to min. \ [€] \tag{2.6}$$

It can be seen that each element is a function of $y_{k\mu\lambda}$ The following formula arises if the $Q_{k\mu} y_{k\mu\lambda}$ is put before the brackets:

$$C_k^I = \sum_{\lambda=1}^{n}\sum_{\mu=1}^{\nu} Q_{k\mu} y_{k\mu\lambda} (c_k^D s_{\mu\lambda} + c_{k\lambda}^A) \to min. \ [€] \tag{2.7}$$

which is a multivariable linear programming (LP) problem [2] with *n*x*v* pieces (decision) variables ( $y_{k\mu\lambda}$ ), and with *n*+*v*+*n*x*v* pieces (limiting) conditions:

$$\sum_{\lambda=1}^{n} y_{k\mu\lambda} = 1 , \sum_{\mu=1}^{v} Q_{k\mu} y_{k\mu\lambda} \le \ell_{k\lambda}^{U} \ \& \ 0 \le y_{k\mu\lambda} \le 1 \tag{2.8}$$

The total cost function contains $n \times v \times p$ pieces variables ($y_{k\mu\lambda}$), additionally the number of conditions is $p \times (n+v+n \times v)$, so this problem requires for optimisation $2^{npv}$ steps ($p$ is the number of final product types, $n$ is the number of assembly plants and $v$ is the number of end users). It results from this that in the event of few plant and user the size of this problem grows exponentially. Some solutions of the (extensive) LP problem with several thousand variables and conditions are the following in [8]: Revised Simplex Algorithm, Product Form of the Inverse, Using Column Generation, Dantzig-Wolfe Decomposition Algorithm, Karmarkar's Method, etc. Because of the large size, the authors worked out two heuristic algorithms (Algorithm A and B) [6] for the solution of the problem.

## 3. Algorithms of assignment based on simplified cost functions

### 3.1. Algorithm A

1) Choose a final product and check which end users placed an order for this product. Choose the end user with the largest ordered quantity.
2) Find the plant, whose distribution cost is most favourable having regard to capacitance limits of plants. Take the next user in decreasing order of ordered quantity and choose assembly plant to it.
3) Find the plant, which can assemble this product at the least cost, and look at it has any capacitance, if yes then check the possibility of change for the other users in decreasing order of ordered quantity. Take the next plant in ascending order of assembly cost.
4) Take the following product and repeat step 1.

### 3.2. Algorithm B

1) Choose a final product and check which users placed an order for this product.
2) Constitute the all possible relations of assembly plant-end user, and we choose them, where distribution costs are most favourable having regard to capacitance limits of assembly plants.
3) Find the plant which can assemble this product with the least cost, and look at it has any capacitance, if yes then check the possibility of change for the other end users in ascending order of distribution cost. Take the next assembly plant in ascending order of assembly cost.
4) Take the following product and repeat step 1.

In step 2 for both algorithms have respect to those relations, where the capacitance limit of plant enables the ordered quantity of the user to be assembled. In step 3 partial ordered quantities are also changed.

## 4. Determination of datamodel needful to sensitivity analysis

A program has been implemented using Delphi programming language, which solves the assignment problem using Hungarian method, algorithm A and B showed in chapter 3. The program dynamically handles the number of plans, users and products. The order matrix, capacity matrix, route matrix, distribution cost and assembly cost matrix can be fed into the computer by automatic and manual. The program makes it possible to save, load and print parameters and results.

The basic data are the followings: $n=3$, $v=6$, $p=8$. Values of the matrix Q can change between 1000 and 6000 pieces, the average of these values is about 2000. The data structure is defined by relative variables for the farther easier changes. Accordingly the order matrix is given in next form: $q_{k\mu} = q_0 \mathrm{a}_{k\mu}$, where $q_0$ is the basic ordered quantity, which is independent of products and users, $\mathrm{a}_{k\mu}$ is the relative ordered quantity of user $\mu$ from product $k$. The modification of value $q_0$ can generate quantity change. The modification of $\mathrm{a}_{k\mu}$ can create structure change. In like manner the capacity matrix can be written down in next form: $\ell^U_{k\lambda} = \ell^U_0 \mathrm{a}^U_{k\lambda}$

$$Q = 2\begin{bmatrix} 0.5 & 2.5 & 0 & 0 & 1.5 & 1 \\ 0 & 1 & 2 & 3 & 2 & 0 \\ 3 & 0 & 1 & 0.5 & 0 & 0 \\ 1.5 & 0 & 0 & 1 & 2.5 & 0 \\ 2 & 1.5 & 1 & 0 & 3 & 0 \\ 0 & 0 & 2 & 0 & 0 & 3 \\ 0 & 0.5 & 0 & 0 & 1 & 2 \\ 1 & 0 & 3 & 0 & 0.5 & 0 \end{bmatrix}\left[\frac{1000\text{pieces}}{\text{cycle}}\right], L^U = 4\begin{bmatrix} 1.5 & 1 & 0.5 \\ 0 & 2 & 2 \\ 0.5 & 0.5 & 1.5 \\ 1 & 1.5 & 0 \\ 1.5 & 0 & 2.5 \\ 0.5 & 1 & 1 \\ 0 & 1 & 1 \\ 1.5 & 0 & 1 \end{bmatrix}\left[\frac{1000\text{pieces}}{\text{cycle}}\right]$$

The values of the route matrix S can change between 20 and 250 km, the average value is about 100 km.

Ratio of the specific distribution and assembly cost: the values of $\delta = c^D_0/c^A_0$ can be 0.2, …, 2, let $\delta$ be 1 now. The formula respecting the calculation of the specific distribution cost: $c^D_k = c^D_0 \mathrm{a}^D_k$, where $c^D_0 = c_0\delta$ is the distribution basic cost, which is independent of products, $\mathrm{a}^D_{k\beta} = \mathrm{a}^D_{00}\mathrm{a}^D_{k0}\mathrm{a}^D_{0\beta}$ is the parameter of proportionality, its value for average product and vehicle $a_{00}^{D} = 1$, $a_{k0}^{D} = 0.8 - 1.2$, $a_{0\beta}^{D} = 0.7 - 1.3$. The formula of the calculation of the specific assembly cost: $c_{k\lambda}^{A} = c_{0}^{A} a_{k\lambda}^{A}$, where $c_{0}^{A} = c_{0}$ is the assembly basic cost, which is independent of products, $a_{k\lambda}^{A} = a_{00}^{A} a_{k0}^{A} a_{0\lambda}^{A}$ is the parameter of proportionality, where $a_{00}^{A} = 1$, $a_{k0}^{A} = 0.7 - 1.4$, $a_{0\lambda}^{A} = 0.8 - 1.2$.

We defined a data-structure, which is suitable sensitivity analysis and comparison of the different optimisation methods too. The sensitivity analysis covers only the products, but its data-model is useable for sensitivity analysis of the specific costs. During investigation the matrix Q, $L^{U}$ and S are fixed. We suppose values of $c_0$, $\delta$, $a_{0\lambda}^{A}$ and $a_{0\beta}^{D}$ to be constant. Let the value of the last two parameters be 1, so the assembly parameter of proportionality is independent of assembly plants and the delivery parameter of proportionality is independent of delivery vehicles, so the vehicle is given. During the sensitivity analysis regarding product costs by both algorithm A and algorithm B only the value of $a_{k0}^{A}$ and $a_{k0}^{D}$ change between the above-defined limits (the parameters of proportionality depend on only the products). In the following we complete the comparison of the two methods for different products by a simple example.

$$S = 100 \begin{bmatrix} 0.2 & 0.8 & 1.5 \\ 2.5 & 0.6 & 1.2 \\ 1.8 & 2 & 1 \\ 0.6 & 0.5 & 1.5 \\ 2 & 1 & 2.5 \\ 2.2 & 1.2 & 0.2 \end{bmatrix} [\mathrm{km}]\ C^{D} = c_0 \delta \begin{bmatrix} 0.6 \\ 0.7 \\ 0.8 \\ 0.9 \\ 1 \\ 1 \\ 1.1 \\ 1.2 \end{bmatrix} \left[ \frac{/\,\mathrm{piece}}{100\mathrm{km}} \right], C^{A} = c_0 \begin{bmatrix} 0.7 \\ 1 \\ 1.2 \\ 0.8 \\ 1.3 \\ 0.9 \\ 1.4 \\ 1.1 \end{bmatrix} [\ /\,\mathrm{piece}]$$

## 5. Sensitivity analysis

### 5.1. Sensitivity analysis of algorithms concerning to products

By the help of the parameters the values of the specific costs and the orders presented in the objective function can be simply changed, and so parameter sensitivity analysis can be done. The three-dimensional matrix Y is converted in the interest of the briefer representation, that in the plane the matrix $y_{\mu k}$ can be seen and the values $\lambda$ are represented smaller numbers. The indexes H, A and B of the matrix Y refer to the methods. This example is solved using the Hungarian method in case of $\delta=1$ we get the following matrix Y in %:

$$Y=\begin{bmatrix}
\boxed{100^{0^{0}}} & \boxed{0^{0^{0}}} & 33^{17^{50}} & \boxed{100^{0^{0}}} & \boxed{100^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & 100^{0^{0}} \\
20^{80^{0}} & 0^{0^{100}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{100}}} & \boxed{0^{0^{0}}} & \boxed{0^{100^{0}}} & \boxed{0^{0^{0}}} \\
\boxed{0^{0^{0}}} & \boxed{0^{0^{100}}} & 0^{0^{100}} & \boxed{0^{0^{0}}} & 0^{0^{100}} & 50^{0^{50}} & \boxed{0^{0^{0}}} & 33^{0^{67}} \\
\boxed{0^{0^{0}}} & 0^{67^{33}} & \boxed{0^{100^{0}}} & 50^{50^{0}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} \\
100^{0^{0}} & 0^{100^{0}} & \boxed{0^{0^{0}}} & 0^{100^{0}} & 33^{0^{67}} & \boxed{0^{0^{0}}} & \boxed{0^{100^{0}}} & 100^{0^{0}} \\
\boxed{0^{0^{100}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & 0^{67^{33}} & \boxed{0^{0^{100}}} & \boxed{0^{0^{0}}}
\end{bmatrix}[\%]$$

This example is solved using the algorithm A, the results are the followings:

$$Y=\begin{bmatrix}
\boxed{100^{0^{0}}} & \boxed{0^{0^{0}}} & 0^{0^{100}} & \boxed{100^{0^{0}}} & \boxed{100^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & 0^{0^{100}} \\
100^{0^{0}} & 0^{100^{0}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{100}}} & \boxed{0^{0^{0}}} & \boxed{0^{100^{0}}} & \boxed{0^{0^{0}}} \\
\boxed{0^{0^{0}}} & \boxed{0^{0^{100}}} & 100^{0^{0}} & \boxed{0^{0^{0}}} & 100^{0^{0}} & 0^{100^{0}} & \boxed{0^{0^{0}}} & 100^{0^{0}} \\
\boxed{0^{0^{0}}} & 0^{100^{0}} & \boxed{0^{100^{0}}} & 50^{50^{0}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} \\
0^{100^{0}} & 0^{0^{100}} & \boxed{0^{0^{0}}} & 0^{100^{0}} & 0^{0^{100}} & \boxed{0^{0^{0}}} & \boxed{0^{100^{0}}} & 0^{0^{100}} \\
\boxed{0^{0^{100}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & 33^{0^{67}} & \boxed{0^{0^{100}}} & \boxed{0^{0^{0}}}
\end{bmatrix}[\%]$$

Finally it is solved using the heuristic algorithm B, the result is the matrix $Y_B$:

$$Y=\begin{bmatrix}
\boxed{100^{0^{0}}} & \boxed{0^{0^{0}}} & 0^{0^{100}} & \boxed{100^{0^{0}}} & \boxed{100^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & 0^{0^{100}} \\
100^{0^{0}} & 0^{100^{0}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{100}}} & \boxed{0^{0^{0}}} & \boxed{0^{100^{0}}} & \boxed{0^{0^{0}}} \\
\boxed{0^{0^{0}}} & \boxed{0^{0^{100}}} & 100^{0^{0}} & \boxed{0^{0^{0}}} & 100^{0^{0}} & 0^{0^{100}} & \boxed{0^{0^{0}}} & 100^{0^{0}} \\
\boxed{0^{0^{0}}} & 0^{100^{0}} & \boxed{0^{100^{0}}} & 0^{100^{0}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} \\
0^{100^{0}} & 0^{0^{100}} & \boxed{0^{0^{0}}} & 20^{80^{0}} & 0^{0^{100}} & \boxed{0^{0^{0}}} & \boxed{0^{100^{0}}} & 0^{0^{100}} \\
\boxed{0^{0^{100}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & \boxed{0^{0^{0}}} & 33^{67^{0}} & \boxed{0^{0^{100}}} & \boxed{0^{0^{0}}}
\end{bmatrix}[\%]$$

It is worth analysing, what the results of the three methods ($Y_H$; $Y_A$; $Y_B$) after the assignment of assembly plants to the final product requirements of the end users. Individual elements of matrix Y are labelled different tokens under the followings:

square - the suitable elements of matrix $Y_H$; $Y_A$ and $Y_B$ are same;
hexagon - the suitable elements of matrix $Y_A$ and $Y_B$ are same;
circle - the suitable elements of matrix $Y_H$ and $Y_A$ are same.

From the ended tokens can be traced, that from the 6x8=48 elements of the matrix:

there is full coincidence (square) by 32 elements, so (32/48)x100=66,67% consist. If these elements are projected for end users ($\mu^*$) and final products ($k^*$) the number of same elements can be comprised in vectors:

$$\mu^*=[6;6;4;6;3;7] \sum_{k=1}^{6} \mu_k^* = 32\,;\; k^*=[4;3;4;4;4;4;6;3] \sum_{\mu=1}^{8} k_\mu^* = 32 \qquad (4.1)$$

13 elements of the algorithm A and B (hexagon), thus (13/48)x100=27,08% is coincided. The results can be also detailed in vectors like previous:

$$\mu^{**}=[2;2;3;1;4;1] \sum_{k=1}^{6} \mu_k^{**} = 13\,;\; k^{**}=[2;3;2;0;2;1;0;3] \sum_{\mu=1}^{8} k_\mu^{**} = 13 \qquad (4.2)$$

up to two elements of algorithm A and Hungarian method (circle) (both of them in the event of product 4) so (2/48)x100=4,17% add.

the 3 methods gave several results by as far as 1 element (final product 6 of end user 3).

Testing results are summarised in Table 1. If the Hungarian method is compared to the heuristic methods it can be traced that the Hungarian method is over 11 per cent ($167.910c_0/189.110c_0=0{,}8879$) better than the algorithm A. It can be seen that the Hungarian method is also over 11 % ($167.910c_0/189.920c_0=0{,}8841$) better than the algorithm B. The algorithm A approaches to the optimal solution only 0,4 per cent ($189.110c_0/189.920c_0=0{,}9957$) better than the algorithm B, which arises therefrom, that the value $a_{0\lambda}^{A}$ is supposed constant. Testing results show that the Hungarian method guarantees the optimum much better as opposed to the heuristic methods.

**Table 1.** Testing results of assignment algorithms per product in costs ($c_0$)

| Pro-duct | Hungarian method | | | Algorithm A | | | Algorithm B | | |
|---|---|---|---|---|---|---|---|---|---|
| | assembly | delivery | summa | assembly | delivery | summa | assembly | delivery | summa |
| 1. | 7700 | 6900 | 14600 | 7700 | 9660 | 17360 | 7700 | 9660 | 17360 |
| 2. | 16000 | 10780 | 26780 | 16000 | 12740 | 28740 | 16000 | 12740 | 28740 |
| 3. | 10800 | 6560 | 17360 | 10800 | 10480 | 21280 | 10800 | 10480 | 21280 |
| 4. | 8000 | 6030 | 14030 | 8000 | 6030 | 14030 | 8000 | 6840 | 14840 |
| 5. | 19500 | 20400 | 39900 | 19500 | 23000 | 42500 | 19500 | 23000 | 42500 |
| 6. | 9000 | 10800 | 19800 | 9000 | 13200 | 22200 | 9000 | 13200 | 22200 |
| 7. | 9800 | 3740 | 13540 | 9800 | 3740 | 13540 | 9800 | 3740 | 13540 |
| 8. | 9900 | 12000 | 21900 | 9900 | 19560 | 29460 | 9900 | 19560 | 29460 |
| Total | 90700 | 77210 | 167910 | 90700 | 98410 | 189110 | 90700 | 99220 | 189920 |

This table proves the algorithm A and B to give the same results except for one case (final product 4), at the same time all algorithms give the same result just in case of product 7, additionally the result of only the algorithm A analogise with the optimum in case of product 4.

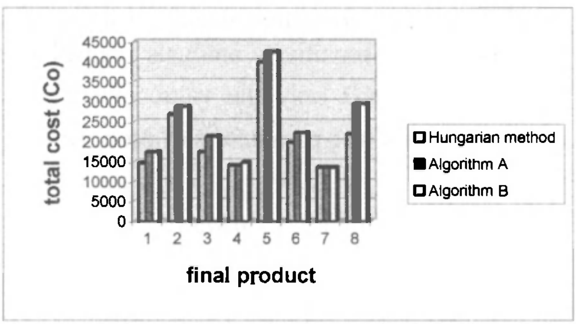



**Figure 2.** Results of the applied methods for products

The results of total cost are represented in Fig. 3 in the event of different values $\delta$ (0.2; 0.5; 0.8; 1; 1.2; 1.5; 1.8; 2) by both two algorithms and Hungarian method.

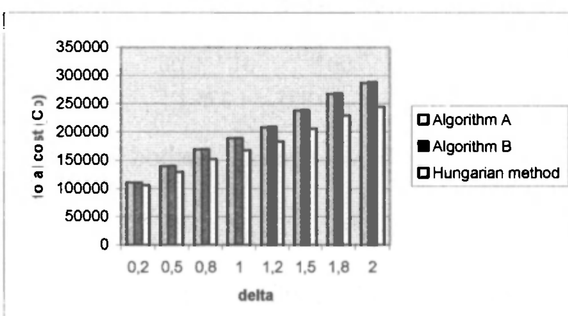



**Figure 3.** Results of the methods in case of different values δ

In Fig. 3 can be experienced that the given total cost results by the methods linear increase with increase of the value $\delta$. At the same time, if the value $\delta$ grows from 0,2 to 2, so it decuples, the total cost will increase 2,6-fold or 2,3-fold. Between two heuristic algorithms there is no great difference in case of different $\delta$, because in case of the specific assembly cost independent of assembly plants did not befall changes in second step of the algorithms, so in fact the final product requirements of end users assigned to assembly plants by only the distribution cost. In all probability, if our investigations are amplified for specific cost dependent on assembly plants, then the algorithm A generate better results, than the algorithm B. It must be observed, there is no represented matrices Y in case of different values $\delta$, but the accordant matrices are all in harmony.

### 5.2. Sensitivity analysis of algorithms concerning to assembly plants

The basic data are analogised with previous example except that the value of $a^{A}_{0\lambda}$ is not constant, so the assembly parameter of proportionality is conditioned by assembly plants too. During the sensitivity analysis regarding assembly costs by both algorithm A and algorithm B only the value of $a^{A}_{0\lambda}$, $a^{A}_{k0}$ and $a^{D}_{k0}$ change between the above-defined limits (the parameters of proportionality depend on only products and assembly plants).

$$C^{A} = c_{0}\begin{bmatrix} 0.7 & 0.525 & 0.875 \\ 1 & 0.75 & 1.25 \\ 1.2 & 0.9 & 1.5 \\ 0.8 & 0.6 & 1 \\ 1.3 & 0.975 & 1.625 \\ 0.9 & 0.675 & 1.125 \\ 1.4 & 1.05 & 1.75 \\ 1.1 & 0.825 & 1.375 \end{bmatrix} [\ \ /\text{piece}]$$

This example is solved using the Hungarian method in case of $\delta = 1$ we get the following matrix Y in %:

$$
Y_H = \begin{bmatrix}
100^{0^{0}} & 0^{0^{0}} & 33^{17^{50}} & 100^{0^{0}} & 100^{0^{0}} & 0^{0^{0}} & 0^{0^{0}} & 100^{0^{0}} \\
20^{80^{0}} & 0^{0^{100}} & 0^{0^{0}} & 0^{0^{0}} & 0^{0^{100}} & 0^{0^{0}} & 0^{100^{0}} & 0^{0^{0}} \\
0^{0^{0}} & 0^{0^{100}} & 0^{0^{100}} & 0^{0^{0}} & 0^{0^{100}} & 50^{50^{0}} & 0^{0^{0}} & 33^{0^{67}} \\
0^{0^{0}} & 0^{67^{33}} & 0^{100^{0}} & 50^{50^{0}} & 0^{0^{0}} & 0^{0^{0}} & 0^{0^{0}} & 0^{0^{0}} \\
100^{0^{0}} & 0^{100^{0}} & 0^{0^{0}} & 0^{100^{0}} & 33^{0^{67}} & 0^{0^{0}} & 0^{100^{0}} & 100^{0^{0}} \\
0^{0^{100}} & 0^{0^{0}} & 0^{0^{0}} & 0^{0^{0}} & 0^{0^{0}} & 0^{33^{67}} & 0^{0^{100}} & 0^{0^{0}}
\end{bmatrix} [\%]
$$

This example is solved using the heuristic algorithm A, the results are the followings:

$$
Y_A = \begin{bmatrix}
100^{0^{0}} & 0^{0^{0}} & 0^{33^{67}} & 100^{0^{0}} & 100^{0^{0}} & 0^{0^{0}} & 0^{0^{0}} & 100^{0^{0}} \\
100^{0^{0}} & 0^{0^{100}} & 0^{0^{0}} & 0^{0^{0}} & 0^{0^{100}} & 0^{0^{0}} & 0^{100^{0}} & 0^{0^{0}} \\
0^{0^{0}} & 0^{0^{100}} & 0^{0^{100}} & 0^{0^{0}} & 0^{0^{100}} & 50^{50^{0}} & 0^{0^{0}} & 33^{0^{67}} \\
0^{0^{0}} & 0^{100^{0}} & 100^{0^{0}} & 50^{50^{0}} & 0^{0^{0}} & 0^{0^{0}} & 0^{0^{0}} & 0^{0^{0}} \\
0^{100^{0}} & 0^{50^{50}} & 0^{0^{0}} & 0^{100^{0}} & 17^{0^{83}} & 0^{0^{0}} & 0^{100^{0}} & 100^{0^{0}} \\
0^{0^{100}} & 0^{0^{0}} & 0^{0^{0}} & 0^{0^{0}} & 0^{0^{0}} & 0^{33^{67}} & 0^{0^{100}} & 0^{0^{0}}
\end{bmatrix} [\%]
$$

Finally this example is solved using the heuristic algorithm B, the result is matrix $Y_B$:

$$
Y_B = \begin{bmatrix}
100^{0^{0}} & 0^{0^{0}} & 0^{0^{100}} & 100^{0^{0}} & 100^{0^{0}} & 0^{0^{0}} & 0^{0^{0}} & 100^{0^{0}} \\
100^{0^{0}} & 0^{0^{100}} & 0^{0^{0}} & 0^{0^{0}} & 0^{0^{100}} & 0^{0^{0}} & 0^{100^{0}} & 0^{0^{0}} \\
0^{0^{0}} & 0^{0^{100}} & 100^{0^{0}} & 0^{0^{0}} & 100^{0^{0}} & 0^{0^{100}} & 0^{0^{0}} & 33^{0^{67}} \\
0^{0^{0}} & 0^{67^{33}} & 0^{100^{0}} & 50^{50^{0}} & 0^{0^{0}} & 0^{0^{0}} & 0^{0^{0}} & 0^{0^{0}} \\
0^{100^{0}} & 0^{100^{0}} & 0^{0^{0}} & 0^{100^{0}} & 0^{0^{100}} & 0^{0^{0}} & 0^{100^{0}} & 100^{0^{0}} \\
0^{0^{100}} & 0^{0^{0}} & 0^{0^{0}} & 0^{0^{0}} & 0^{0^{0}} & 33^{67^{0}} & 0^{0^{100}} & 0^{0^{0}}
\end{bmatrix} [\%]
$$

It is worth analysing too, what the results of three methods ($Y_H$; $Y_A$; $Y_B$) after the assignment of assembly plants to the final product requirements of the end users. It can be traced, that from the 6x8=48 elements of the matrix:

- there are full coincidence by 37 elements, so (37/48)x!00=77,08% consist.

2 elements of the algorithm A and B, thus (2/48)x100=4,17% are coincided.
up to 4 elements of algorithm A and Hungarian method, so (4/48)x100=8,33% add.
3 elements of algorithm B and Hungarian method, so (3/48)x100=6,25% are coincided.
the three methods gave several results by as far as two elements.

Testing results are summarised in Table 2. If the Hungarian method is compared to the heuristic methods it can be traced that the Hungarian method is over 3 per cent ($147.085c_0/152.070c_0=0,9672$) better than the algorithm A. It can be seen that the Algorithm B is 8,3 per cent ($159.290c_0/147.085c_0=1,0830$) worse than the Hungarian method. The algorithm A approaches to the optimal solution more than 4,5 per cent ($152.070c_0/159.290c_0=0,9547$) better than the algorithm B.

**Table 2.** Testing results of assignment algorithms per product in costs ($c_0$)

| Pro-duct | Hungarian method | | | Algorithm A | | | Algorithm B | | |
|---|---|---|---|---|---|---|---|---|---|
| | assembly | delivery | summa | assembly | delivery | summa | assembly | delivery | summa |
| 1. | 7350 | 6900 | 14250 | 7525 | 9660 | 17185 | 7525 | 9660 | 17185 |
| 2. | 11200 | 10780 | 21980 | 11200 | 11480 | 22680 | 11200 | 10780 | 21980 |
| 3. | 7825 | 6560 | 14385 | 7500 | 8160 | 15660 | 8175 | 10480 | 18655 |
| 4. | 6350 | 6030 | 12380 | 6350 | 6030 | 12380 | 6350 | 6030 | 12380 |
| 5. | 15675 | 20400 | 36075 | 15250 | 20900 | 36150 | 15675 | 23000 | 38675 |
| 6. | 7400 | 10800 | 18200 | 7400 | 10800 | 18200 | 7400 | 13200 | 20600 |
| 7. | 5075 | 3740 | 8815 | 5075 | 3740 | 8815 | 5075 | 3740 | 8815 |
| 8. | 9000 | 12000 | 21000 | 9000 | 12000 | 21000 | 9000 | 12000 | 21000 |
| Total | 69875 | 77210 | 147085 | 69300 | 82770 | 152070 | 70400 | 88890 | 159290 |

On the analogy of the former example the Hungarian method guarantees the optimum but this is very time-consuming, on the contrary, the algorithm A and B give only approximation.

The Table 2 proves the algorithm A and B to give the same results in the moiety of cases (final product 1, 4, 7, 8), at the same time all algorithms give the same result in three cases (product 4, 7, 8), additionally the result of only the algorithm A analogise with the optimum in case of product 6 and the result of only the algorithm B analogise with the hungarian method in case of product 2.

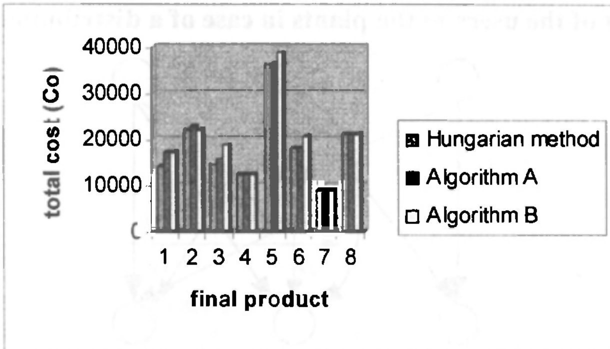



**Figure 4.** Results of the applied methods for products

The results of total cost are represented in Fig. 5 in the event of different values $\delta$ (0.2; 1; 2) by both two algorithms and Hungarian method too.

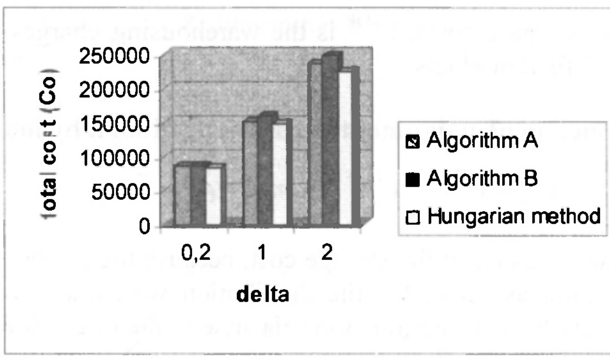



**Figure 5.** Results of the methods in case of different values δ

In Fig. 5 can be experienced that the given total cost results by the methods nonlinear increase with increase of the value $\delta$. At the same time, if the value $\delta$ grows from 0,2 to 2, so it decuples, the total cost will increase $234.840c_0/86.247c_0=2{,}7$ or $248.180c_0/87.715c_0=2{,}8$-fold. Between two heuristic algorithms there is a great difference in case of different $\delta$, because in case of the specific assembly cost dependents on assembly plants have already befallen changes in second step of the algorithms, so in fact the final product requirements of end users assigned to assembly plants by not only the distribution cost.

In the next step we analyse that in case of one-distribution warehouse model how modify the simplified cost function (purchase and delivery cost) in accordance with direct delivery. Subsequently we analyse that the indirect delivery when (for what conditions) become necessary and profitable.

## 6. Assignment of the users to the plants in case of a distribution warehouse

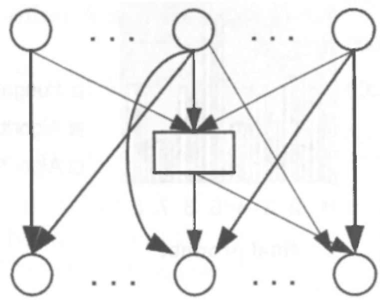


**Figure 6.** One distribution warehouse model

In case of distribution warehouse (indirect delivery) the costs will be as follows:

$$C = C^{PW} + C^{WW} + C^{DW} \rightarrow min.\ [€] \tag{6.1}$$

where $C^{PW}$ is the purchase costs, $C^{WW}$ is the warehousing charges and $C^{DW}$ is the distribution cost of final products.

### 6.1. Objective function of assignment in case of product k by indirect delivery

$$C_k^2 = C_k^{PW} + C_k^{DW} \rightarrow min.\ [€] \tag{6.2}$$

Do not have to take account of the storage cost, because the distribution warehouse built that in the purchase cost. By the distribution warehouse the cost function come the following (that is analogous formula arise to the direct delivery):

$$C_k^2 = C_k^{PW} + C_k^{DW} \quad \sum_{\mu=1}^{\nu} Q_{ku} y_{ku}^W (c_k^D s_u^W + c_k^{PW})\ \ [€] \tag{6.3}$$

### 6.2. Delivery from distribution warehouse and the specific purchase cost

The specific purchase cost from the distribution warehouse in case of product k:

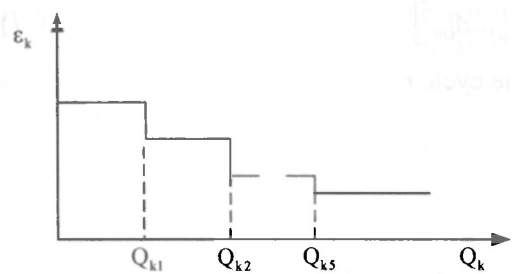
be conditioned by the weighted value of maximal assembly capacity of assembly plants;

$$\overline{c_k^A} = \sum_{\lambda=1}^{n} \frac{\ell_{k\lambda}}{\ell_{k0}} c_{k\lambda}^A\ \ [€], \text{ where } \ell_{k0} = \sum_{\lambda=1}^{n} \ell_{k\lambda} \tag{6.4}$$

be conditioned by the ordered amount and the ordering incoming time.

$$c_k^{PW} = \varepsilon_k\{Q_k\}\,\alpha_k\{t^{OW} - t^{OP}\}\,\overline{c_k^A}\ \ [€] \tag{6.5}$$

The function $\varepsilon_k\{Q_k\}$:




$$\varepsilon_{k1} = 1,3 \quad 0 < Q_k < 1000$$
$$\varepsilon_{k2} = 1,2 \quad 1000 < Q_k \leq 2000$$
$$\varepsilon_{k3} = 1,1 \quad 2000 < Q_k \leq 3000$$
$$\varepsilon_{k4} = 1,0 \quad 3000 < Q_k < 4000$$
$$\varepsilon_{k5} = 0,9 \quad 4000 < Q_k < 5000$$
$$\varepsilon_{k6} = 0,8 \quad 5000 < Q_k$$

**Figure 7.** Specific purchase cost

$t_k^{OP}$ is the ordering time of product *k*: $t_k^{OP} = t_0 \Delta_k$, that after ordering incoming into distribution warehouse must be fulfilled the demand by this time and $t_k^{OW}$ is the ordering time of the final product *k* by distribution warehouse. If

$t_k^{OP} > t_k^{OW}$, then $\alpha_k = 1$, there is no overcharge because of in retard order;

$t_k^{OP} < t_k^{OW}$, then the Figure 8. determines the value of $\alpha_k$.

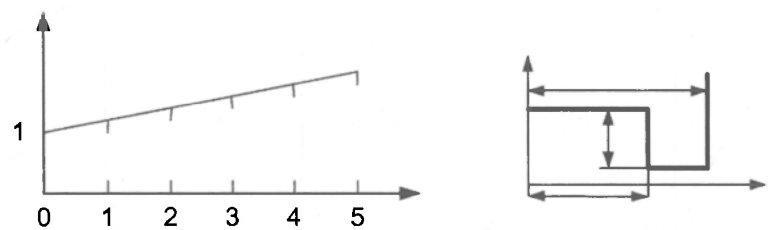



**Figure 8.** Determination of the value of $\alpha_k$ and the case of $t_k^{OP} < t_k^{OW}$

Remarks:

by $t_k^{OP} > t_k^{OW}$ $\alpha_k = 1$, because the distribution warehouse can get the product optimistically from assembly plants;

by $t_k^{OP} < t_k^{OW}$, the amount $Q_k$ have to storage in the distribution warehouse for term of $t_k^{OW} - t_k^{OP}$

Ordering amount from product *k* by the user $\mu$ in the cycle *r*:

$$\overline{Q}_{kr} = \left[ Q_{kr1} \cdots Q_{kr\mu} \cdots Q_{kr\nu} \right] \tag{6.6}$$

Ordering time of ordering amount from product *k* by the user $\mu$ in the cycle *r* is given in former cycles of the term of ordering appearance ($t_k^{OP}$ arises from this):

$$\Delta_{kr} = \left[\Delta_{kr1} \ldots \Delta_{kr\mu} \ldots \Delta_{krv}\right] \tag{6.7}$$

If the end user $\mu$ from the product $k$ in the cycle $r$

preorder a cycle before: $\Delta_{kr\mu} = 1$;

preorder $f$ cycles before: $\Delta_{kr\mu} = f$.

### 6.3. Exploration of possibility of direct, indirect delivery

We analyse, that the ordering amount of the end user $\mu$ exceed the cycle capacity of the assembly plant $\lambda$ or not. If the end user $\mu$ in the cycle $r$ by the plant $\lambda$ $Q_{kr\mu} \le \ell_{kr\lambda}$, then $\Theta_{kr\mu\lambda} = 1$ and $Q_{kr\mu} > \ell_{kr\lambda}$, then $\Theta_{kr\mu\lambda} = 0$, where $\ell_{kr\lambda}$ is the free cycle capacity of cycle $r$ in the assembly plant $\lambda$. The matrix $\Theta_{kr}$ means, which assembly plants can fulfill the product $k$ in the cycle $r$ to the end user $\mu$. The matrix $C_k^I = C_k^A + C_k^D$ which comprises the (direct) assembly and delivery cost is given.

$$\Theta_r^k = \begin{array}{c} \\ 1 \\ \mu \\ v \end{array} \begin{array}{c} \begin{array}{ccc} 1 & \lambda & n \end{array} \\ \begin{bmatrix} & & \\ & \Theta^k_{r\mu\lambda} & \\ & & \end{bmatrix} \end{array} \qquad C_l^k = \begin{array}{c} \\ 1 \\ \mu \\ v \end{array} \begin{array}{c} \begin{array}{ccc} 1 & \lambda & n \end{array} \\ \begin{bmatrix} & & \\ & C^k_{l\mu\lambda} & \\ & & \end{bmatrix} \end{array} \tag{6.8}$$

In every cycle the capacity of every assembly plant from the product $k$ is known.

$$\overline{L}_k = \left[\ell_{k1} \ldots \ell_{k\lambda} \ldots \ell_{kn}\right] \tag{6.9}$$

where $\ell_{k\lambda}$ is the maximal assembly capacity per cycles in case of final product $k$.

#### *6.3.1. Determination of marginal cases of direct delivery to the end users from the assembly plants*

The direct delivery consists in case of that end users ($\mu$) from the product $k$ in the cycle $r$, where the followings are fulfilled:

in case of end user $\mu$ it can be seen the assembly plant in ordering time of the cycle $r$, where the delivery amount is less, than the free assembly capacity (in case of more solution we have to choose where the total cost $C_k^I$ is minimal);

out of assembly plants which fulfill the former condition, those continue to exist the direct delivery, where the total cost of direct delivery is less than in case of indirect delivery from the distribution warehouse (if there are more that assembly plants, then we choose that whose total cost is minimal);

if by the analysed order $\mu$ there is no direct delivery, because by the ordering amount in the cycle $r$

- o do not have a sufficiency of free assembly capacity of every plant and/or
- o delivery from the distribution warehouse is soluble less total cost $C^1_{kr\mu} > C^2_{kr\mu}$ where $C^1_{kr\mu}$ and $C^2_{kr\mu}$ is the total cost of the direct delivery and the delivery from the distribution warehouse in case of final product $k$ in the cycle $r$ by the demand of the end user $\mu$.

### *6.3.2. Further principles to the algorithm for optimisation of the direct or indirect delivery possibility*

the schedule have to be accomplished by cycle time $t_0$;

in accordance with schedule by the order of the user the sequence per produce:

- o we begin that order, where the ordering time $t^{OP}$ is the maximum;
- o if in case of more end users there are equal ordering time, then we choose that $C^1_{k\mu\lambda}$ is the minimal having regard to the matrix (6.8/a) and (6.8/b);
- o the chosen cost form part of $\mu$ and $\lambda$ is less than the delivery from the distribution warehouse $C^1_{k\mu\lambda} < C^2_{k\mu}$, then the delivery to the end user $\mu$ will be from the assembly plant $\lambda$ in the cycle $r$. In this case the row $\mu$ of the matrix $\Theta_{kr}$ is cancelled and by the vector $\overline{L}_{kr}$ the value $\ell_{kr\lambda}$ will be decreased with the measure of capacity decrease, we fix the assembly amount in the matrix $G_{kr}$, where $G_{kr\mu\lambda}$ is the assembly amount from the final product $k$ in the cycle $r$ in the assembly plant $\lambda$ for the end user $\mu$;
- o if in the above case the: $C^1_{k\mu\lambda} \geq C^2_{k\mu}$, then the delivery comes from the distribution warehouse, the delivery amount of product $k$ can be ordered, in this case we cancel the row $\mu$ of the matrix $\Theta_{kr}$, but the vector $\overline{L}_{kr}$ do not change, the product get into the matrix $\pi_{k\mu r}$, which shows the delivery amount from the product $k$ in case of the end user $\mu$ in the cycle $r$;
- o we continue this algorithm until all order of cycle $r=1$ have graded;
- o we have to determine $\overline{L}_{kr0}$ which is the free assembly capacity in the cycle $r$, which arises therefrom that $y_{kr\mu\lambda}$ may 1 or 0, that is the demand of one end user will be fulfilled by only one assembly plant.

### *6.3.3. Additional algorithm for the case b)*

- we analyse the assembly capacity vector $\overline{L}_{kr0}$;

we search the minimal element of the matrix (6.8/b): $C^{l}_{k\mu\lambda}$ and we analyse the matrix $Y^{a}_{kr}$, that the end user $\mu_0$ get where the product $k$, if

- o from the assembly plant $\lambda_0$, then jump to the next step and search the next minimal $C^{l}_{k\mu_0\lambda_0}$;
- o it is fulfilled elsewhere, then we have to analyse, that pass to the free capacity's debit delivery some of the order from the assembly plant $\lambda_0$ the modified cost $C^{l*}_{k\mu\lambda} < C^{l}_{k\mu\lambda}$, or $C^{l*}_{k\mu\lambda} < C^{2}_{k\mu}$;
- o in case of the modification we get the lower cost, then the revised matrix $Y^{b}_{kr}$ fix the result (we choose the actual elements of the matrix) and revise the vector $\overline{L}_{kr0}$;
- o continue the analysis of the next minimal value $C^{l}_{k\mu_0\lambda_0}$ of matrix (6.8/b) until the every element of the matrix has analysed, the produced modified assignment matrix $Y^{b}_{kr}$ is better than $Y^{a}_{kr}$.

## 7. Conclusions and future works

The scientific paper proves that in the network-like operating assembly systems the Hungarian method hard to use by the large-sized problems and the two heuristic methods worked out for optimal assignment of assembly plants to the final product requirements of the end users by simplified cost function to give nearly equivalent result. If the ratio of the specific assembly and delivery basic cost is changed - provided that the specific assembly costs are constant and do not depend on the assembly plants - then the total cost is on the linear increase with the increase of the specific delivery basic cost, but tenfold increase of the delivery cost results in only about 2,5-fold increase of the total cost. If the specific assembly cost depends on assembly plants then the total cost has already increase nonlinear with increase of the specific delivery cost, and tenfold increase of the delivery cost results in only about $224.295c_0/84.745c_0=2{,}6$-fold increase of the total cost.

In the near future we would like to amplify the described model with more distribution warehouses and to analyse the change of the optimum in comparison to the solution of one-distribution warehouse model by the fulfilment of the final product requirements of the end users.

## REFERENCES

[1] CSELÉNYI, J. TÓTH, T.: *Mathematical model for optimisation of a product assembly system integrated by logistics and operating in a network like way*, WESIC 2001 Workshop on European Scientific and Industrial Collaboration, Published by Drubbel Institute for Mechatronics, Twente, pp. 81-92, 2001.

[2] EVANS, J. R.: *Applied Production and Operations Management*, West Publishing Company, Cincinnati. 1993.

[3] OLÁH, B., BÁNYAI, T., CSELÉNYI, J.: *Logistical tasks of co-operative assembly plants*, 3rd International Conference on AED, Prague, pp. 110, 2003.

[4] OLÁH, B., BÁNYAI, T. CSELÉNYI, J.: *Algorithm of optimal assignment of assembly plants and end users within the framework of products in co-operative assembly system*, Miskolcer Gespräche, Miskolc, pp. 145-150, 2003.

[5] OLÁH, B., BÁNYAI, T., CSELÉNYI, J.: *Sensitivity analysis of optimal assignment of assembly plants and end users within the framework of products in a cooperative assembly system*, microCAD 2004, University Press, Miskolc, pp. 97-102, 2004.

[6] OLÁH, B., BÁNYAI, T., CSELÉNYI, J.: *Optimal assignment of assembly plants to the final product requirements of the end users in a cooperative assembly system*, 15th International DAAAM Symposium, Vienna, pp. 321-322, 2004.

[7] TÓTH, T.: *Design and Planning Principles, Models and Methods in Computer Integrated Manufacturing*, University Press, Miskolc, 1998. (in Hungarian)

[8] WINSTON, W.: *Operation Research: Applications and Algorithms*, Aula Press, Budapest, pp. 765-793, 2003.

[9] SMILOWITZ, K. R. DAGANZO, C. F.: *Cost Modeling and Design Techniques for Integrated Package Distribution Systems*, Industrial Engineering and Management Science Working Paper, Northwestern University, pp. 04-06, 2004.

[10] JÜNEMANN, R.: *Planung- und Betriebsführung-Systeme für die Logistik*, Verlag TÜV Rheinland, 1990.

[11] SCHÖNENBURG, E., HEINZMANN, F., FEDDERSEN, S.: *Genetische Algorithmen und Evolutionsstrategien (Eine Einführung in Theorie und Praxis der simulierten Evolution)*, Addison-Wesley, 1994.

[12] SCHÖNSLEBEN, P.: *Integrates Logistik Management, (Planung und Steuerung von umfassenden Geschäftprozessen.)*, 22-5 Springer Verlag Berlin, Heidelberg, 1998.

[13] CSELÉNYI, J.: *Mathematisches Modell und Algorithmus des Montageprozesses für ein elektronisches Produkt*, Magdeburger Schifften zur Logistik, Logistik an der Universität Miskolc, Aktuelle Logistikforschung, Magdeburg,, pp. 35-44, 2002.

[14] CSELÉNYI, J., TÓTH, T.: *Interrelations between Logistics and Production Control*, Proceeding of MANSA'94 9th National Conference of the South African Institute of Industrial Engineers, Manufacturing in South Africa, pp. 365-370, 1994.