TRANSPORT OPTIMIZATION IN JOB-SHOP PRODUCTION USING LINEAR PROGRAMING

Lado Lenart[1], Martin Tomšič[1], Janez Kušar[2], Marko Starbek[2]

[1]"Jožef Stefan" Institute
1000 Ljubljana, Jamova 39, Slovenia
[2]University of Ljubljana, Faculty of Mechanical Engineering
1000 Ljubljana, Tržaška 25, Slovenia

lado.lenart@ijs.si (Lado Lenart)

Abstract

In the medium-size factory in production of finally plastics products for household use the reconstruction plan was made to improve the utilization of the central automatic pallet storehouse using intermediate storage facilities. The locations and capacities of this additional storage places should be established considering the existent floor layout, the number and capacity of forklifts, the possibility of inter-process storage of half-products and restoring the pallets after the production cycle.

Linear programming (LP) was used to resolve the machine assignment problem for a set of characteristic production plans. Supported with non-delay scheduler program the group of human experts then produced the series of Gantt charts as they would be used in job shop production environment. The analysis of Gantt charts, with the production plan partitioned into appropriate time sections, was made with the optimization program. LP here was used to calculate optimal costs for transport alternatives, to locate the storage areas, to determine their capacity and in particular the inter-process storage possibilities were taken into account. In addition the circular flow of pallets was included into formulation. The results were the part of documentation for restructuring the factory.

The LP algorithm itself is a variant of transport algorithm, using time sections which correspond with the distinct vertical cross-sections of Gantt-charts. The specific task was the inclusion of partially full pallets into the way of transport and the treating of emptied pallets as well.

Keywords: Production layout, Machine assignment, Inter-storage calculation, Transport way alternatives, Linear programming.

Presenting Author's biography

Lado Lenart. Received his Ph.D. degree in Chemical Engineering by the Technical University Clausthal (Germany) in 1972 and his Ph.D. degree in Electrical Engineering by the University in Ljubljana (Slovenia) in 1993. He was with the ISKRA-Avtomatika R&D Institute and is now working in robotics field at "Jožef Stefan" Institute in Ljubljana (SI). He was assistant Professor at the University in Maribor (Sl) for 'Real Time Systems'. His interests are embedded systems and operational research.

1 General

Linear programming (LP) is widely used in operational research, the algorithm is recently often supported by interior point methods. In the context of our paper the LP techniques for scheduling, layout design, assignment and transport are of main importance. The broad field of practical solutions for these particular problems is reported in monograph [1], treating the linear assignment and scheduling, with accent on mixed LP. An other common reference is [2], also for scheduling. Layout design for chemical plants with some similarity to our problem is in [3]. The transport problem is structurally very near to the idea of LP and the sufficient information can be found in quite elementary books, e.g.[4]. The problematic of the project then is not in the theory, rather in large scale computation matrices. The corresponding LP solvers should be used.

In former project logistcs simulator [5] of factory producing domestic devices like vacuum cleaners, water cisterns and similar plastic products was developed with the goal to define the suitable machine pool and stock positioning or rearrange the existing locations and capacities. Technically feasible production variants were simulated and recommendations were provided using the material and semi-products transport as the criterion. Also the transport costs were combined with inventory costs in weighted object function. This paper presents an attempt to solve the same problem as in [5] analytically using LP model with objective function, which was slightly changed in the formulation to correspond with cyclic production. Exact solutions were produced for every defined production mix and for every machine pool and storage layout. It was necessary to solve this program repeatedly for the series of real production mixes, with different proposals for machine pools and storage allocations. Still more, every solution was corrected to eliminate the storages where their capacities were estimated to be too small.

Let first used standard LP models be sketched, for storage as well as for transport of goods.

For storage design in the simplified model the process variables, functions and constants are:

$x(t) =$ inventory at time t

$\alpha(t) =$ rate of ordering from manufacturers, $\alpha > 0$

$d(t) =$ customer demand (known)

$\gamma =$ cost of ordering 1 unit

$\beta =$ cost of storing 1 unit

The dynamic model:

$$\dot{x} = \alpha(t) - d(t) \tag{1.1}$$

Payoff function:

$$P(\alpha(.)) = -\int_0^T \gamma\alpha(t) + \beta x(t)\,dt \tag{1.2}$$

It is easy to guess the first optimal control strategy: if $x_0 > 0$ one should first not order anything $(\alpha = 0)$ and let the inventory fall to zero: thereafter one orders only enough to cover the demand. In the more realistic model the purchase is possible only at the beginning of time sections. In that case the second optimal strategy including stochastic is given with (1.3). J_k is the object function at the beginning of the stage *k*. (1.3) can be solved in recursion. The normal strategy then is the mean of the first and second strategy, requiring the purchase after the stock is less then the given value, the problem can even be solved with LP.

$$\begin{aligned} J_k(x_k) = \min_{\alpha_k > 0}\, \underset{d_k}{E}\left\{\gamma d_k + H(x_k + \alpha_k - d_k)\right\} + \\ + J_{k+1}(x_k + \alpha_k - d_k) \end{aligned} \tag{1.3}$$

The second factor influencing the economy of distributed warehouses are the transport costs. The classical transport problem is the linear transport problem, where the load shall be brought from *m* storehouses to *n* customers. If the *j*-th customer orders a product unit from the *i*-th storehouse, the transport cost for it is p_{ij}. Then one has the linear program:

$$\begin{aligned} \sum_{i=1}^{m} b_i &= \sum_{j=1}^{n} a_j \\ P &= \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} = \min \end{aligned} \tag{1.4}$$

In (1.4), b_i are the initial stocks and a_i are the customer demands. The variant of this transport problem is transshipment problem, with the necessity to organize transshipment storages.

The more general form of the transport algorithm is the transshipment version of it. The transshipment nodes are the 'buffer' storages, which enable better distribution of goods.

LP programming shall be also applied in the machine allocation program. In the basic form of the problem, machines say W_1, W_2, W_3, W_4 shall be operated by persons *A,B,C,D*. In the time unit some person *i* makes on machine W_j a specific profit, say P_{ij}. The LP formulation is the general transportation algorithm (1.4), the numerical efficiency can be improved by using Hungarian method.

In the paper LP is first applied to the machine-assignment problem. It is shown, how this results were used to simplify the schedule problem. Based on given schedule, stock and transport calculations were carried out. For the specification we consider not only the transport of parts from stock to machines and finished products to stocks, but also the inter-storage transport and the transport of empty box-pallets.

2 Machine assignment

Let production plan be presented as a set of product types $H = \{H_1, H_2, \ldots H_n\}$, which will be produced in quantities $\{h_1, h_2, \ldots, h_n\}$. Often assignment optimization algorithm can be reduced, if some fixed assignments exists because of physical singular mapping between parts types and machine subgroups. More formally: if there is some subset $\{H_r\} \in H$ and any subset $\{S_s\} \in S$ and single relation exists $R_{rs}(H_r, S_s)$, then the system (H, S) can be decomposed into the two classes $(H_r, R_s), (H - H_r, S - S_r)$. For each class then an independent assignment algorithm can be applied. The model tries to distribute the work on machines using the following objective function:

$$C^a = \sum_i \sum_k c_{ik} x_{ik} \tag{2.1}$$

In (2.1) i is the machine index and k is the part index. Variable x_{ik} is the quantity of parts of type H_k, produced on machine S_i. Coefficient c_{ik} is the cost of production of a single part of type H_k on machine S_i. Then the machines, which are able to produce, say parts of type H_k, must fulfill the plan h_k.

$$\sum_i x_{ik} = h_k; \quad \forall k \tag{2.2}$$

In practice, the available active time is not equal for all devices, as some machines operate in one and others in two shifts, they have various set-up times or the set-up times are schedule dependent. Let s_i be the available run time for each machine. To produce parts of type H_k on machine S_i the productivity a_{ik} is required which is generally not equal for all machines able to produce the same item. Eq. (2.3) simply describes the fact, that the sum of production times for all items, which are to be produced on machine S_i, must be less than the available run time s_i.

$$\sum_k a_{ik} x_{ik} \le s_i; \quad \forall i \tag{2.3}$$

The problem (2.1), together with restrictions (2.2), (2.3) is an LP of very moderate size even for monthly production plan for the case under study. Some remarks shall be given concerning the parameters and their influence on the LP- result. One has to run the LP several times, first setting the availabilities large enough to get the feasible solution. Then s_i will have to be gradually reduced according to additional set of technological restrictions until the solution satisfies the planning expert. The solution is searched by 'trial and error' method. One sees at this point, that in LP problem parameters are considered to be constant in time. If this approximation is too rough, the time axis is divided into distinct sections. The parameters in the single section must be constant. From factory's integrated data base, for every time section the following relations must be retrieved to run an appropriate LP-program:

$H \times S \to S$	producing types of parts on machines
$H \to h$	time section capacity
$S \to s$	machine availability plan
$H \times S \to \{a_{ik}\}$	production rate on machine per item
$H \times S \to \{c_{ik}\}$	production price per item

In this section the LP program is presented in his simplified form: in reality the products are assembled on machines and this requires some extra measures to be taken, which however are similar to the techniques described in section 4. Therefore the detailed description of assignment program shall here be omitted.

3 Sequencing and scheduling

For the scheduling problem the non-delay schedule algorithm was applied, the same as programmed in the logistic simulator. Normally the monthly production is scheduled. The big series of identical products are assembled in machine pools, whereby the tardiness of series and assembly tree are the main factors determining schedule plan. Therefore the schedule algorithm can be kept quite simple, after the machine assignment problem was solved as described in the section above. However, the initial stock conditions play the essential role when designing schedule. The non-delay schedule was constructed for two types of operational conditions. The first type, which guarantees the minimal global execution time, is based on the assumption, that in the given production cycle (e.g. monthly mix) none of the assembly lines ever waits for parts, either for parts delivered 'offline' from external suppliers nor for parts, which are made on own machines 'inline'. The first type schedule can easily be realized if the initial stock levels are set appropriate high. Let this type of schedule be called the 'maximal production schedule'. For this schedule the condition of cyclic production must be fulfilled too, which requires, that the stock levels for parts, which can be produced 'inline', at the end of schedule interval are equal to the initial stock levels. The second type of schedule starts with 'offline' stocks filled, but the stocks with 'inline' parts are empty. It follows for this type of schedule, that first 'inline' parts must be produced in the proper quantity and then the assembly lines can start. Let this type of schedule be called 'minimal stock inventory schedule'. This type of schedule must be cyclic too. As the parts in production are transported on pallets (the single pallet carries several tens or even thousand of parts of single type), unnecessary delay can occur if e.g. two assembly lines are waiting for the same pallet. Therefore the program enables the work with splitting pallets.

Before the schedule produced was used for subsequent calculation of transport and stock capabilities, it was checked by human operators. For the machine assigning phase, the most common intervention of him was the exclusive assignment of parts to the machine. The intervention of operator on schedule was mostly the splitting of extensive orders between two machines, having positive effect on schedule make-span. The Gantt chart produced interactively by the machine assignment algorithm, the non-delay schedule and planning staff was then partitioned to time slices. The next rules must be obeyed: a) any start of production of a new part type on any machine opens the new time interval; b) any end of part type production on any machine ends the current time interval.

Typically then the continuous occupancy interval of any machine in Gantt chart is divided in more time slices. An example of test schedule is given in fig.1. Here, the second time slice begins when machine S5000 starts to produce type 1003 parts.

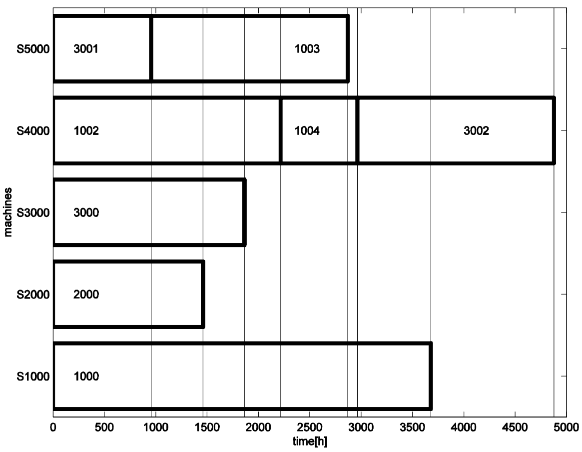

Fig. 1 Simplified Gantt-chart

4 Stock calculation

The cornerstone of our calculations is to determine the mass flow balance considering the constraints in stock capacities for each time slice separately and then compose the general criterion function for all time slices. Machine flow balance is presented in fig.2.

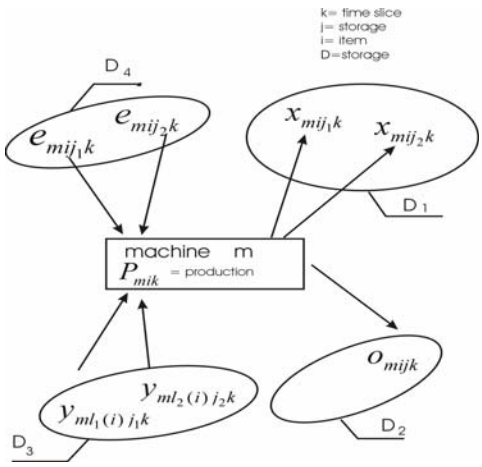

Fig. 2 Machine flow balance

In model construction the next specific indices are used, with a few text sensitive exceptions:

-*i* product or assembly part index

-*j* storage index in (D1,D2,D3,D4) sets of stocks

-*k* index of the time slice/section

-*m* machine index

Let the next variables be introduced:

P_{mik} ... production (in single units) on machine *m* of item *i* in time interval *k*

x_{mijk} ... transport flow of item *i* from machine *m* to stock *j* in time interval *k*

D_1 ... set of allowable stocks to store half-products or finished parts which are produced on machines

D_2 ... set of allowed stocks for empty pallets, which get free on machine input places

D_3 ... set of allowed stocks for assembly parts and raw materials

D_4 ... set of allowed stocks of empty box-pallets to supply machine *m*

O_{mik} ... output flow of empty box-pallets, which get free during the production on assembly lines on the input places

$c^e(i)$ ... proportional fraction of empty-box pallets, which get empty on machine input places, of production of item *i*

o_{mijk} ... flow of empty pallets, which get free on machine *m* producing item *i* and are transported to stock *j*

y_{mljk} ...transport flow of item l from stock j to machine m

E_{mik} flow of empty pallets to output buffer of machine m where they are loaded with finished i items

e_{mijk} components of the flow E_{mik} from more stores with empty box-pallets with index j.

w_{mij} ...transport costs for unit of product i on the shortest path from machine m to stock j

$\alpha(l,i)$...the number of parts of input item l to produce one output item i, normally integer number

$\beta(i)$...the fractional part of one pallet, which is necessary to carry one item i

h_{ji}^0 ...initial inventory of item i in storage j

C_j ...capacity of storage j

$f_{i,j,u,k}$ ...transfer of item i from stock j to stock u in the period k

4.1 Output flow balance

The subsection title refers output flow balance for machine m producing item i in time slice k. The term 'flow' is used always as a quantity of items, transferred in the given time-slice k. First the output flow balance for any machine will be considered (see fig.2) : the known production P_{mik} has to be divided among all storage facilities in the group D_1, which can accept an item with index i. This delivers

$$P_{mik} = \sum_{j \in D_1} x_{mijk} \ ; \quad \forall m,i,k \qquad (4.1)$$

The items are carried on box-pallets. Those box-pallets, which are emptied on the machine input places, contribute to the output flow balance. Clearly this flow is proportional to the production of finished parts with factor of proportion $c^e(i)$.

$$O_{mik} = c^e(i) P_{mik} = \sum_{j \in D_2} o_{mijk} \ ; \quad \forall m,i,k \qquad (4.2)$$

Factor $c^e(i)$ vanishes, if assembly parts are not carried on pallets, this is a case for jet machines with central distribution of plastic granulate.

4.2 Input flow balance

Input flow balance refers to machine m producing item i in time slice k. Input flow generally has next terms: plastic granulate, half-products and box-pallets. Again the machine producing P_{mik} (fig. 2) shall be considered. For example, for quantity P_{mik} of item i the quantities $P_{mik}\alpha(l,i)$ of part l has to be used. The necessary amount of part l can be delivered from more stocks, then the flow equation gets:

$$P_{mik}\alpha(l,i) = \sum_{j \in D_3} y_{mljk} \ ; \quad \forall (l,i), \forall m,k \qquad (4.3)$$

The term $\forall(l,i)$ defines the system of linear equations (4.3). The pallets here are not considered as input components, as they are transported together with other items and do not need transport devices for themselves.

The second term in input flow balance are empty box pallets, which must be transported from stocks D_4 to machine m where they are loaded with its products. The flow equation for these pallets is:

$$E_{mik} = \beta(i) P_{mik} = \sum_{j \in D_4} e_{mijk} \ ; \quad \forall m,i,k \qquad (4.4)$$

4.3 Inter-stock flow

The analysis of flows can be concluded with the supposition, that along with the flows calculated in previous subsections, the complementary transport takes place, enabling materials to move between stocks, so as to prevent stock overload. To incorporate this supposition in an LP model, new flows are introduced, denoted $f_{i,j,u,k}$ to define the flow of item i from source stock j to destination stock .

4.4 Stock capacity constraints

In the process described so far the storage facilities were filled and emptied without taking into account their capacities and state of filling ; even the capacities of paths between storage places were neglected. The next step will establish a system of inequalities, in order to prevent storage places to over – or under-flow. The current filling grade in a storage facility is not a new variable. It depends on variables in formulas which have been already developed, provided that the initial inventory is known. Let the initial states for all items over all stocks be given with parameters h_{ji}^0, and as usually j being the storage and i the item index. Then the quantity of item i relative to the initial state at the end of k-th time slice is given by the following formulas:

$$h_k^{D_1}(j,i) = \sum_{l=1}^{k} \sum_{m} x_{mijl} I(j,i,D_1) \qquad (4.5)$$

$$h_k^{D_4}(j,i) = -\sum_{l=1}^{k} \sum_{m} e_{mijl} I(j,i,D_4) \qquad (4.6)$$

ISBN 978-3-901608-32-2

$$h_k^{D_3}(j,i) = -\sum_{l=1}^{k}\sum_{m} y_{mijl} I(j,i,D_3) \tag{4.7}$$

$$h_k^{D_2}(j,i) = \sum_{l=1}^{k}\sum_{m} o_{mijl} I(j,i,D_2) \tag{4.8}$$

$$\begin{aligned} h_k^{f}(j,i) = \sum_{l=1}^{k}\sum_{m}\big(& f_{ijul} I(i,j,u,l) - \\ & - f_{iujl} I(i,u,j,l)\ \big) \end{aligned} \tag{4.9}$$

In (4.5)- (4.9) dummy binary variables I are introduced as multipliers, which activate the adjoining transport term. They are given a priori as constants. Furthermore, intersections $D_i \cap D_j$ are generally not empty, so index j passes over the union $D_1 \cup D_2 \cup D_3 \cup D_4$ of stacks. Then from (4.5)- (4.9) we get a set of buffer capacity constraints:

$$\begin{aligned} &\sum_{i(j)} h_{ji}^0 + \sum_{i(j)}\Big(h_k^{D_1}(j,i) + h_k^{D_4}(j,i) + \\ &+ h_k^{D_3}(j,i) + h_k^{D_2}(j,i) + h_k^{f}(j,i\ )\Big) \le C_j\ ; \\ &\forall k \end{aligned} \tag{4.10}$$

The summation term $i(j)$ in (4.10) runs over all items i, which can be stored in stock j. The similar summation terms appear in the series of any further formulas and are to be resolved by analogy. Then (4.10) also states that the stock on hand should be less than the storage capacity over all time slices. For simplicity reasons , the storage capacity in (4.10) is virtually expressed as the number of parts, the LP program however calculates the real volume. Indeed, the storage capacity must be positive:

$$\begin{aligned} &h_{ij}^0 \ge 0 \\ &\sum_{i(j)} h_{ji}^0 + \sum_{i(j)}\Big(h_k^{D_1}(j,i) + h_k^{D_4}(j,i) + \\ &+ h_k^{D_3}(j,i) + h_k^{D_2}(j,i) + h_k^{f}(j,i\ )\Big) \ge 0\ ; \quad \forall k \end{aligned} \tag{4.11}$$

5 Object function: transport and stock inventory costs, LP solution

The LP objective function construction begins by first considering time interval k. This criterion function deals with transport and stock inventory costs whereas, at later stage it will be expanded over the whole production period.

From (4.1) the transport costs of machine produced parts from all machines to all allowed stocks are, if the specific transport costs are denoted with *'w'*:

$$C_{trx}(k) = \sum_{m(k)}\ \sum_{i(k,m)}\ \sum_{j(m,i,k)} w_{mij}^{x} x_{mijk} \tag{4.12}$$

In analogy with (4.2) it follows for stream of emptied box-pallets to all allowed stocks for empty box pallets:

$$C_{tro}(k) = \sum_{m(k)}\ \sum_{i(k,m)}\ \sum_{j(m,i,k)} w_{mij}^{ep} o_{mijk} \tag{4.13}$$

From (4.3) we get (4.14) for transport of half-products from all stocks to all machines. It is a technical detail, that the products are never transported from machine to machine.

$$C_{try}(k) = \sum_{m(k)}\sum_{l(i)}\sum_{j(l)} w_{mlj}^{y} y_{mljk} \tag{4.14}$$

Eq. (4.15) describes cost for transport of empty box-pallets from all allowable stocks for empty box-pallets to all output buffers of machines, where they are uploaded with products:

$$C_{tre}(k) = \sum_{m(k)}\sum_{i(m,k)}\sum_{j} w_{mij}^{ep} e_{mijk} \tag{4.15}$$

From transport the inter-stock flows remain, they are put into formula:

$$C_{trf}(k) = \sum_{i}\sum_{j}\sum_{u} f_{ijuk} w_{iju}^{f} \tag{4.16}$$

The sum of all partial costs is:

$$\begin{aligned} C_{tr}(k) = C_{trx}(k) + C_{tro}(k) + \\ + C_{try}(k) + C_{tre}(k) + C_{trf}(k) \end{aligned} \tag{4.17}$$

Additionally to transport costs, stock inventory costs $C_{in}(k)$ are important, too. If the initial stock levels h_{ij}^0 are kept constant, then the stock inventory cost is constant too. The LP program work efficiently, if h_{ij}^0 become process variables. It is sufficient to adopt in the criterion function only the total stock inventory at the end of each time slice. So we get:

$$\begin{aligned} C_{in}(k) = \sum_{j}\sum_{i(j)} h_{ji}^0 + \sum_{j}\sum_{i(j)}\Big(h_k^{D_1}(j,i) + h_k^{D_4}(j,i) + \\ + h_k^{D_3}(j,i) + h_k^{D_2}(j,i) + h_k^{f}(j,i) \end{aligned} \tag{4.18}$$

Finally the objective function for total production time span is:

$$C = \sum_{k} C_{tr}(k) + \rho_1 \sum_{k} C_{in}(k) \tag{4.19}$$

The constant ρ_1 in (4.19) is the weighing factor that evaluates the interrelation between transport and process inventory costs.

The criterial function (4.19) subjected to flow constraints (4.3) ,(4.4), (4.5),(4.6), (4.7), (4.8), (4.9) along with buffer capacity constraints (4.10) ,(4.11) and the general restriction that all flows should be positive, form the LP problem.

Here the short outline of the LP with his supporting programs is given: the non-delay scheduler works with the list of user's orders and with assembly trees for each product. The result is Gantt chart of the process with a production plan for every machine or assembly line. From Gantt chart the integral flows can be read out for every machine and LP gives the sub-partition of integral flows into part flows between machines and storage areas. Besides the flows, all stock levels are known in every time slice.

The representative graphic solution of LP is given in fig.3. The density of transport flows is given for the real factory layout, the transport paths and stock locations were obtained from CAD- files and the shortest transport paths used in LP program were calculated by Dijkstra algorithm. In fig.3 the width of the connection line between source and destination points is proportional to the material flow. In the drawing, red points are machine pools, green points are assembly lines and yellow points are stocks and inter-stocks.

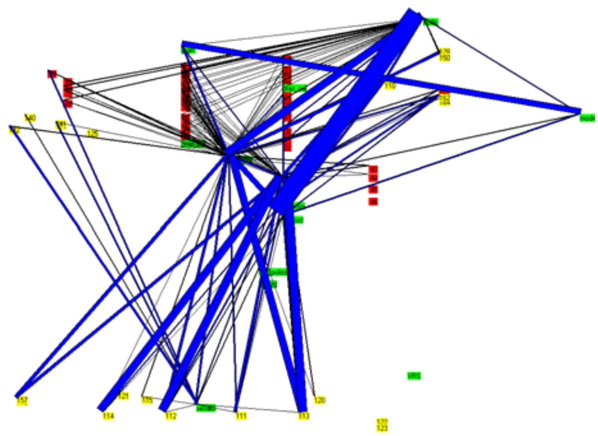

Fig. 3 The transport flows

6 Discussion and conclusion

The program technique has been designed for determining the disposition and capacities of machines and inter-stage buffers while optimizing the transport and inventory costs for arbitrary process layout and production schedule with LP. The benefit of this technique in comparison with standard simulator [5] is in use of analytic LP solution instead of varying the simulator parameters for distribution of material flows and buffer capacities until some satisfying results are obtained. The exponential growth of variables in LP program, with increasing complexity of schedule with respect to production time and the number of time sections in Gantt chart, can easily be mastered even for monthly production, particularly as the great part of flows is relative very small and can be excluded from LP without influencing the result. For a single schedule one can define the great number of storage places and machine pools and then run the LP repetitively with gradual elimination of uninteresting locations. The inclusion of long term production schedule (say a year instead of month now) would theoretically solve the problem but would immensely increase the number of LP-variables. So the set of monthly schedule plans was a preferable choice. The results then were averaged over long-term periods to get more reliable data.

As the flow density is given as time dependent function, the maximal transport flow over the schedule period can be analysed, the result is the number of transport devices. With LP program it is not possible to solve the problem of routing the transporting fork-lifts or avoid the transport bottlenecks, however the data from LP solution can be used as input to other programs controling transport facilities.

It is characteristic for the program, that beside the stock inventory costs the maximal stock levels are known: then the investment costs for rearranging stock and other facilities locations can be calculated and form together with the LP-results a data base for multi-decisional problem.

The further advantage of using LP is the possibility to organize the 'cyclic' production, enforcing that the stack levels for 'factory made' half-products after finishing the schedule are the ones at the beginning of the schedule.

The conclusion for the praxis then is, that LP techniques can be useful to determine the optimal disposal of transport resources and to locate the producing and storing facilities in optimal manner. The problem of credibility of results is not LP itself, but the reliability of future production planning, not to mention other managerial risks.

7 References

[1] M.W.Padberg,M.P.Rijal. Operation research in decision theory.Springer.1996

[2] C.A.Flouder,X.Lin. Mixed integer linear programming in process scheduling: Modelling, Algorithms, Applications. *Annals of operations research*, 139,1:131-162, 2001.

[3] M.C.Georgiadis,G.Schilling,G.E.Rotstein, S.Machietto. A general mathematical programing approach for process plant layout. *Computers & Chemical Engineering*, 35,7:823-840, 1999.

[4] R.Bronson,H.Naadimuthu.Operations research. MacGraw-Hill.1997

[5] M.Lenart,H.Golle,Z.Likar. Schedule,storage and transport simulation in the real job shop plant. In Viljem Rupnik, editor, *Proceedings of the 5th international symposium on operational research SOR'99,* Preddvor, Slovenia, September 30 , 1999, 227-232