Flexible Assembly Line Balancing with Alternate Assembly Plans and Duplicate Task Assignments

Tadeusz Sawik
Department of Computer Integrated Manufacturing
University of Mining & Metallurgy
30-059 Cracow, Poland

Abstract – **The paper presents integer programming formulations and a heuristic solution procedure for a bicriterion loading and assembly plan selection problem in a flexible assembly line. The problem objective is to simultaneously determine an allocation of assembly tasks among the stations and to select assembly sequences and assembly routes for a set of products so as to balance station workloads and minimize total transportation time in a unidirectional flow system. In the approach proposed, first the station workloads are balanced using a linear relaxation-based heuristic and then assembly sequences and routes are selected for all products, based on a network flow model. An illustrative example is provided.**

I. INTRODUCTION

A flexible assembly line (FAL) is a a unidirectional flow system made up of a set of assembly stations in series and a loading/unloading (L/U) station, linked with an automated material handling system. Each station consists of assembly robots with a finite work space available for component feeders and gripper magazines, e.g., [1].

In a FAL different product types are assembled simultaneously. A typical assembly process proceeds as follows. A base part of a product is loaded on a pallet and enters the line at the L/U station. As the pallet is carried by a conveyor or an automated guided vehicle through a series of assembly stations, components are assembled with the base part. A product may bypass some stations but does not revisit any station. When all the required components are assembled with the base part, it is carried back to the L/U station and the complete product leaves the system.

The two main short-term planning issues are to assign tasks and products to stations with limited capabilities in order to equalize station workloads and to minimize interstation product movements subject to precedence relations among the tasks for a mix of product types.

Productivity of a flexible assembly system can be enhanced by allowing greater flexibility at the short-term planning. There are two sources that may increase the flexibility of the loading decision making: flexibility in the product assembly plans and duplicate assembly task assignments. In this paper both options are considered simultaneously.

One can argue that fixing the sequence of assembly tasks for each product before the loading stage, without the knowledge of the task assignments and the product mix to be simultaneously assembled in a flexible flow line, decreases the chances of getting optimal or good workload balance. Avoiding premature selection of product assembly sequences leads to a better balancing methodology. When balancing of a FAL all duplicate task assignments and product assembly routes that simultaneously satisfy the assembly precedence relations for all products in a unidirectional flow line should be considered.

In the balancing problem studied in this paper alternative assembly sequences (i.e., alternative chain-type precedence relations among the assembly tasks) for each product are assumed to be available) as well as duplicate assignments of assembly tasks to different stations are allowed, which leads to enhanced routing flexibility. The short-term planning problem considered is actually a combination of simultaneous machine loading, product routing and assembly sequence selection. The problem is an extension of the loading problem with prefixed single assembly sequence for each product, presented in [5,6] for a general flexible assembly system, where multidirectional product flows and revisiting of stations are allowed.

One of the widely used modelling techniques employed to solve FAL loading problem is integer programming, e.g., [3,4,5,6]. The modelling and solution approach proposed in this paper is similar to that given in [4,5] for a general flexible assembly system. The FAL loading, assembly routing and assembly sequence selection problem is formulated as a bicriterion integer program with the objectives of balancing station workloads and minimizing total transportation time. A two-level solution procedure is proposed. First, the station workloads are balanced using a linear relaxation-based heuristic and then the best assembly sequences and assembly routes are selected based on a network flow model to minimize total transportation time.

The paper is organized as follows. In the next section

0-7803-4192-9/97/$10.00

integer programming formulation for the bicriterion simultaneous loading, routing and assembly sequence selection problem is presented. A two-level solution approach proposed is described in Section III. Numerical example is provided in Section IV and conclusions are given in the last section.

II. SIMULTANEOUS LOADING, ROUTING AND ASSEMBLY SEQUENCE SELECTION

In this section the bicriterion integer programming formulation is presented for simultaneous assembly sequence selection, machine loading and product routing in a flexible assembly line.

Let us consider a FAL made up of m assembly stations $i \in I = \{1, \ldots, m\}$ in series $1, \ldots, m$ connected by a unidirectional AGV path, and the L/U station. In the system n different types of assembly tasks $j \in J = \{1, \ldots, n\}$ can be performed to simultaneously assemble v products $k \in K = \{1, \ldots, v\}$ of various types. A product completed on station i is transferred to the next station $i+1$ or another downstream station $l > i$, depending on the product assembly route, and it cannot revisit any station.

Let $I_j \subset I$ be the subset of stations capable of performing task j. Each station i has a finite work space b_i where a limited number of component feeders and gripper magazines can be placed. As a result only a limited number of assembly tasks can be assigned to one assembly station. Let a_{ij} be the amount of station $i \in I_j$ working space required for task j.

Each product k requires a subset J_k of assembly tasks to be performed subject to precedence relations defined by the assembly sequence selected for this product. Let $S = \{1, \ldots, w\}$ be the set of all assembly sequences available and let T_s be the subset of tasks j in the sequence $s \in S$. Each assembly sequence $s \in S$ can be represented by the set R_s of immediate predecessor-successor pairs of assembly tasks (j, r) such that task $j \in T_s$ must be performed immediately before task $r \in T_s$.

For each product k a subset $S_k \subset S$ of alternative assembly sequences is available. The subsets S_k, $k \in K$ are disjoint so that each assembly sequence $s \in S_k$ can be applied only for product k, i.e., for any two different products $k_1, k_2 \in K$, $k_1 \neq k_2$: $S_{k_1} \bigcap S_{k_2} = \emptyset$. If the same sequence of tasks can be applied to assemble different products $k \in K$ of the same type, then in the model it is represented by assembly sequences in different sets S_k.
Let us notice that $T_s = J_k$ for all assembly sequences $s \in S_k$, $k \in K$.

Finally, denote by p_{jk} the assembly time required for task $j \in J_k$ of product k and by q_{il} the transportation time required to transfer a product from station i to station l, $l > i$.

The objective of the problem is to determine the optimal assignment of assembly tasks to station and to select best assembly sequences and assembly routes for all products so as to balance the station workloads and to minimize total transportation time.

A feasible solution of the combined loading, routing and sequence selection problem must satisfy the following six basic types of constraints:

- Each assembly task must be assigned to at least one station;
- One assembly sequence must be selected for each product;
- For each product and assembly sequence selected all assembly tasks required must be completed;
- The total space required for the tasks assigned to each station must not exceed the station finite work space available;
- Tasks must be assigned to stations in such a way that for each assembly sequence selected, precedence relations among the tasks are maintained in a unidirectional flow line with no revisiting of stations required;
- Each product must be successively routed to the stations where the required tasks have been assigned.

Table 1: NOTATION

		Indices
i	=	assembly station, $i \in I = \{1, \ldots, m\}$
j	=	assembly task, $j \in J = \{1, \ldots, n\}$
k	=	product, $k \in K = \{1, \ldots, v\}$
s	=	assembly sequence, $s \in S = \{1, \ldots, w\}$
		Input parameters
a_{ij}	=	working space required for task j at station i
b_i	=	working space of station i
p_{jk}	=	assembly time for task j of product k
q_{il}	=	transportation time from station i to station l
I_j	=	the set of stations capable of performing task j
J_k	=	the set of tasks required for product k
R_s	=	the set of immediate predecessor-successor pairs of tasks (j, r) for assembly sequence $s \in S$ such that task j must be performed immediately before task r
S_k	=	the set of assembly sequences available for product k
T_s	=	the set of tasks in assembly sequence s

The following decision variables are introduced to model the loading, routing and sequence selection problem (for notation used, see Table 1):

u_s = 1, if assembly sequence $s \in S$ is selected; otherwise $u_s = 0$

x_{ij} = 1, if task j is assigned to station $i \in I_j$; otherwise $x_{ij} = 0$

y_{iljs} = 1, if for assembly sequence s, after completion of task j on station i, product is transferred to station $l \geq i$ to perform next task; otherwise $y_{iljs} = 0$

z_{ijs} = 1, if for assembly sequence s task j is assigned to station i ; otherwise $z_{ijs} = 0$.

Let us notice that for each product $k \in K$, variables y_{iljs} (or z_{ijs}) define a unique assembly route, since only one assembly sequence $s \in S_k$ is selected for each $k \in K$ and subsets S_k are disjoint.

A bicriterion machine loading, product routing and assembly sequence selection problem is formulated below.

Problem LRS: *Balancing station workloads and minimizing total transportation time in a flexible assembly line*

Minimize

$$P_{max},\ Q_{sum} \tag{1}$$

subject to

$$\sum_{i\in I_j}\sum_{l\geq i} y_{iljs} = u_s;\ s\in S, j\in T_s \tag{2}$$

$$\sum_{l\leq i} y_{lijs} - \sum_{l\geq i} y_{ilrs} = 0;\ i\in I_r, (j,r)\in R_s, s\in S \tag{3}$$

$$\sum_{k\in K}\sum_{s\in S_k}\sum_{j\in J_k}\sum_{l\geq i} p_{jk} y_{iljs} \leq P_{max};\ i\in I \tag{4}$$

$$\sum_{i\in I}\sum_{l>i}\sum_{s\in S}\sum_{j\in T_s} q_{il} y_{iljs} = Q_{sum} \tag{5}$$

$$\sum_{i\in I_j} x_{ij} \geq 1;\ j\in J \tag{6}$$

$$\sum_{j\in J} a_{ij} x_{ij} \leq b_i;\ i\in I \tag{7}$$

$$y_{iljs} \leq x_{ij};\ i\in I_j, l\geq i, l\in I_r, (j,r)\in R_s, s\in S \tag{8}$$

$$y_{iljs} \leq x_{lr};\ i\in I_j, l\geq i, l\in I_r, (j,r)\in R_s, s\in S \tag{9}$$

$$y_{iljs} \leq u_s;\ i\in I_j, l\geq i, l\in I_r, (j,r)\in R_s, s\in S \tag{10}$$

$$\sum_{s\in S_k} u_s = 1;\ k\in K \tag{11}$$

$$y_{iljs} = 0;\ i\in I, l<i, j\in T_s, s\in S \tag{12}$$

$$u_s \in \{0,1\};\ \forall s \tag{13}$$

$$x_{ij} \in \{0,1\};\ \forall i,j \tag{14}$$

$$y_{iljs} \in \{0,1\};\ \forall i,l,j,s \tag{15}$$

The first objective function in (1) represents imbalance of the workload distribution. Minimization of the maximum workload P_{max} subject to (4) implicitly equalizes the station workloads. Constraint (2) ensures for each product and assembly sequence selected that all of its required tasks be allocated among the stations. Equalities (3) are the flow conservation equations for each station, assembly sequence and a pair of successively performed tasks. Constraints (4) and (5) define the workload of the bottleneck station and the total transportation time, respectively. Constraint (6) ensures that each task is assigned to at least one station, and by this admits alternative assembly routes for products. Constraint (7) is the station capacity constraint. Constraints (8), (9) and (10) ensure that each product successively visits such stations where the required tasks may be assembled subject to precedence relations defined by the assembly sequence selected. Constraint (11) ensures that only one assembly sequence is selected for each product. Finally, constraint (12) eliminates upstream flow of products in a unidirectional flow system.

III. A TWO-LEVEL LOADING, ROUTING AND ASSEMBLY SEQUENCE SELECTION

An efficient solution to the bicriterion problem **LRS** can be found by applying the lexicographic approach. In most practical situations the first objective of balancing the station workloads is more important for the FAL performance than the second one of minimizing the total transportation time. Hence, first one solves the **LRS** problem with the objective of minimizing P_{max} and with constraint (5) omitted.

Let us notice that the product assignment variables z_{ijs} and the assembly routing variables y_{iljs} are dependent via flow balance equations

$$z_{ijs} = \sum_{l\geq i} y_{iljs};\ \ j\in T_s,\ \ i\in I_j \tag{16}$$

Using relation (16) the single objective problem can be reformulated into the following loading problem **L**.

Problem L: *Balancing station workloads in a flexible assembly line*

Minimize

$$P_{max} \tag{17}$$

subject to

$$\sum_{i\in I_j} z_{ijs} = u_s;\ s\in S, j\in T_s \tag{18}$$

$$\sum_{k\in K}\sum_{s\in S_k}\sum_{j\in J_k} p_{jk} z_{ijs} \leq P_{max};\ i\in I \tag{19}$$

$$z_{irs} \leq \sum_{l\leq i} z_{ljs};\ i\in I, s\in S, (j,r)\in R_s \tag{20}$$

$$\sum_{j\in J} a_{ij}x_{ij} \le b_i;\ i \in I \quad (21)$$

$$\sum_{i\in I_j} x_{ij} \ge 1;\ j \in J \quad (22)$$

$$z_{ijs} \le x_{ij};\ s \in S, j \in T_s, i \in I_j \quad (23)$$

$$x_{ij} = 0;\ j \in J, i \notin I_j \quad (24)$$

$$z_{ijs} = 0;\ s \in S, j \notin T_s \text{ and/or } i \notin I_j \quad (25)$$

$$\sum_{s\in S_k} u_s = 1;\ k \in K \quad (26)$$

$$u_s \in \{0,1\};\ \forall s \quad (27)$$

$$x_{ij} \in \{0,1\};\ j \in J, i \in I_j \quad (28)$$

$$z_{ijs} \in \{0,1\};\ s \in S, j \in T_s, i \in I_j \quad (29)$$

The objective P_{max} (17) is a measure of system imbalance and represents workload of the bottleneck station defined by constraint (19). Equation (18) ensures that for each assembly sequence selected all required tasks are allocated among the stations. Constraint (20) for each assembly sequence selected maintains the precedence relations among the tasks and ensures that a product does not revisit any station in a unidirectional flow system. If for assembly sequence s, task r is assigned to station i, then task j that is to be performed immediately before r, i.e., $(j,r) \in R_s$, must be assigned to a station l such that $l \le i$. Constraint (21) is the station capacity constraint. Constraint (22) ensures that each task is assigned to at least one station. Constraint (23) ensures that each product is assigned to such stations where the required tasks may be assembled. Constraints (24) and (25) eliminate assignment of tasks and products to inappropriate stations.

The loading problem **L** can be solved by using a linear relaxation-based heuristic presented in [5], which finds a solution to the integer program **L** starting from the optimal solution of the LP relaxation LP(**L**) of **L**. The LP relaxation of **L** is formed by removing the integrality restrictions on u_s, x_{ij} and z_{ijk}. If one solves LP(**L**) with a simplex algorithm an optimal basic solution is found with a set of task assignments $\tilde{x}_{ij}$ for which some integrality constraint is violated. The linear relaxation-based heuristic proposed gives a "rounding" scheme for fractional assignments which leaves unchanged, as far as possible, the fixed assignments which already satisfy integrality. The resulting binary assignments obtained must satisfy the station capacity constraints (21). If the capacity constraint (21) for some station i is violated, then a violated valid inequality is found which is not satisfied by the fractional solution $\tilde{x}_{ij}$ of the linear programming relaxation LP(**L**) of **L**.

The violated valid inequalities are generated in the following form (see [2])

$$\sum_{j\in C} x_{ij} \le |C| - 1;\ i \in I \quad (30)$$

where C is an unknown subset of J such that the following conditions hold for station i

$$\sum_{j\in C} a_{ij} > b_i \ \text{ and } \ \sum_{j\in C} \tilde{x}_{ij} > |C| - 1$$

The violated valid inequalities are generated by solving the auxiliary knapsack separation problem.

$$\gamma_i = \min\{\sum_{j\in J}(1-\tilde{x}_{ij})\xi_j : \sum_{j\in J} a_{ij}\xi_j \ge b_i + 1,\ \xi_j \in \{0,1\};\ j \in J\} \quad (31)$$

If $\gamma_i < 1$, then subset C is determined by variables $\xi_j = 1$, i.e., $C = \{j : \xi_j = 1\}$.

The corresponding valid inequality (30) is violated by task assignments $\tilde{x}_{ij}$ by the amount $(1-\gamma_i)$.

The violated valid inequalities generated in each iteration of the heuristic are added to the linear programming relaxation of **L** solved in the previous iteration. The procedure is repeated until the rounding off scheme yields a feasible task assignment (for details, see [5]).

The model of the loading problem **L** can be further strengthened by the addition of valid constraints to the original formulation, which reduce the feasible region for the linear programming relaxation without eliminating the optimal integer solution. For example, the following equality implied by constraints (18) and (26) can be added

$$\sum_{s\in S_k}\sum_{i\in I_j} z_{ijs} = 1;\ k \in K, j \in J_k \quad (32)$$

Let P^L_{max} be the solution value of P_{max} (17) obtained by applying the linear relaxation-based heuristic. Having solved problem **L**, the biobjective problem **LRS** is next reduced into the following single objective simplified assembly routing and sequence selection problem **RS** of minimizing Q_{sum} for fixed task assignments x^L_{ij} obtained for **L** and with P_{max} bounded from above by P^L_{max}.

Problem RS: *Minimizing total transportation time in a flexible assembly line for fixed task assignments*

Minimize

$$Q_{sum} = \sum_{i\in I}\sum_{l>i}\sum_{s\in S}\sum_{j\in T_s} q_{il}y_{iljs} \quad (33)$$

subject to (2), (3), (11), (12), (13), (15)

$$\sum_{k\in K}\sum_{s\in S_k}\sum_{j\in J_k}\sum_{l\ge i} p_{jk}y_{iljs} \le P^L_{max};\ i \in I \quad (34)$$

$$y_{iljs} \le x^L_{ij}x^L_{lr}u_s;\ i \in I_j, l \ge i, l \in I_r, (j,r) \in R_s, s \in S \quad (35)$$

The objective Q_{sum} is a measure of the material handling system total workload. Constraint (34) defines an upper bound on each station workload. Constraint (35)

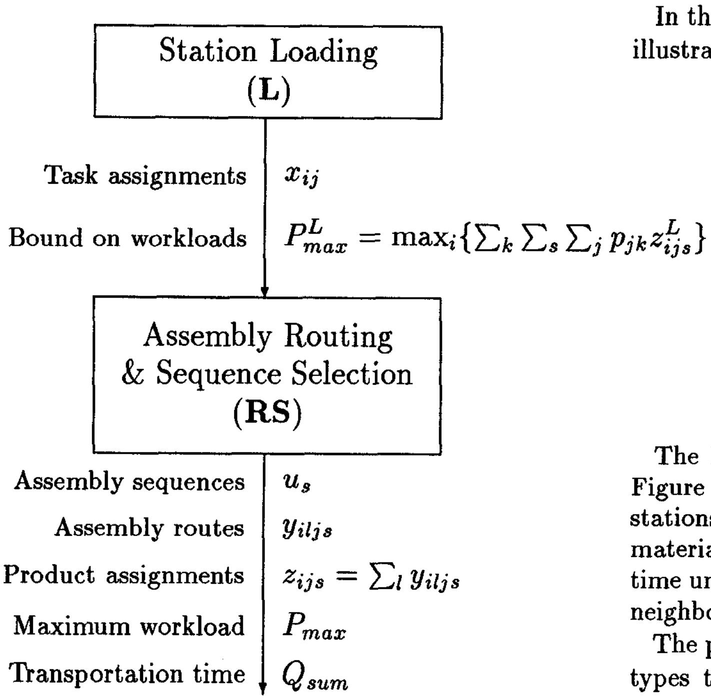

Fig. 1: A two-level task assignment, assembly routing and sequence selection in a FAL

is equivalent to variable upper bound constraints (8), (9) and (10) for fixed task assignments x^L_{ij}.

The routing problem has an embedded network flow structure and hence can be easily solved, e.g., by direct application of an LP code and some rounding off procedure, if nonintegral solution is obtained or by solving Lagrangian relaxation of **RS** with respect to the constraint (34).

The two-level approach for station loading, assembly routing and assembly sequence selection problem in a FAL is summarized in Figure 1.

It should be pointed out that, in addition to task assignments x_{ij}, solution to the upper level problem **L** determines also an assembly sequence and the corresponding assembly route selected for each product, and hence an upper bound

$$Q^L_{sum} = \sum_{\{s:u_s=1\}} \sum_{(j,r)\in R_s} \sum_{i\in I_j} \sum_{l\in I_r} q_{il} z^L_{ijs} z^L_{lrs}$$

on total transportation time. Additional cut constraint $Q_{sum} \le Q^L_{sum}$ can be incorporated into the model of **RS**.

However, the two sets of variables u_s, z_{ijs} remain free for the second step of the solution procedure and their final improved values are determined only by solving the lower level problem **RS**.

IV. EXAMPLE

In this section simple numerical example is presented to illustrate application of the approach proposed.

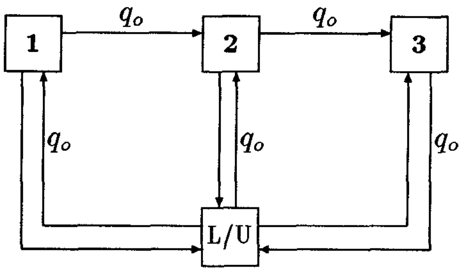

Fig. 2: FAL configuration

The FAL configuration for the example is provided in Figure 2. The system is made up of $m = 3$ assembly stations $i = 1,2,3$ in series, and one L/U station. The material handling system is unidirectional with $q_o = 2$ time units required for an AGV to move between any two neighbouring stations.

The production batch consists of $v = 5$ products of four types to be assembled of $n = 10$ types of components using $w = 10$ assembly sequences where two alternative assembly sequences are available for each product. The available sequences $s \in S_k$ of tasks $j \in J_k$ required to assemble each product $k = 1,2,3,4,5$ are the following:

$k = 1,\ s = 1:\ (L,1,2,3,4,6,8,U)$
$k = 1,\ s = 2:\ (L,1,2,4,3,6,8,U)$
$k = 2,\ s = 3:\ (L,1,2,4,5,6,7,9,10,U)$
$k = 2,\ s = 4:\ (L,1,2,6,4,5,7,9,10,U)$
$k = 3,\ s = 5:\ (L,2,3,4,5,7,8,9,10,U)$
$k = 3,\ s = 6:\ (L,2,7,3,4,5,8,9,10,U)$
$k = 4,\ s = 7:\ (L,1,3,5,6,7,8,9,10,U)$
$k = 4,\ s = 8:\ (L,1,3,8,5,6,7,9,10,U)$
$k = 5,\ s = 9:\ (L,1,3,5,6,7,8,9,10,U)$
$k = 5,\ s = 10:\ (L,1,3,8,5,6,7,9,10,U)$

where L/U denotes loading/unloading operations.

Let us notice that products $k = 4,5$ are of the same type, and hence the corresponding pairs of assembly sequences $s = 7,9$ and $s = 8,10$ are identical.

The assembly times p_{jk} are shown below ($p_{jk} = 0$ indicates that task j is not required to assemble product k).

$$[p_{jk}] = \begin{bmatrix} 4,4,0,4,4 \\ 2,2,2,0,0 \\ 2,0,2,2,2 \\ 2,2,2,0,0 \\ 0,4,4,4,4 \\ 2,2,0,2,2 \\ 0,3,3,3,3 \\ 5,0,5,5,5 \\ 0,2,2,2,2 \\ 0,4,4,4,4 \end{bmatrix}$$

For each assembly task j the working space a_{ij} required for the corresponding component feeders is independent on the station i, i.e., $a_{ij} = a_j$, $\forall i \in I_j$, $j \in J$ ($a_{ij} = 0$ indicates that station i is incapable of performing task j).

$$[a_{ij}] = \begin{bmatrix} 1,2,3,1,2,3,0,0,0,0 \\ 0,0,0,1,2,3,1,2,3,5 \\ 1,2,3,0,0,0,1,2,3,5 \end{bmatrix}$$

The available working space for each station is: $b_1 = 10$, $b_2 = 10$, $b_3 = 10$.

First, an approximate solution of the loading problem **L** strengthened by the addition of equality (32) has been determined applying the linear relaxation-based heuristic. A feasible solution has been found after 6 iterations of the heuristic. The violated valid inequalities (30) generated in each iteration and the final solution results obtained are presented below.

Iteration 1: $x_{1,1} + x_{1,2} + x_{1,3} + x_{1,5} + x_{1,6} \leq 4$

Iteration 2: $x_{3,7} + x_{3,8} + x_{3,9} + x_{3,10} \leq 3$

Iteration 3: $x_{3,1} + x_{3,8} + x_{3,9} + x_{3,10} \leq 3$

Iteration 4: $x_{2,4} + x_{2,5} + x_{2,7} + x_{2,8} + x_{2,10} \leq 4$,
$x_{3,2} + x_{3,8} + x_{3,9} + x_{3,10} \leq 3$

Iteration 5: $x_{1,2} + x_{1,3} + x_{1,4} + x_{1,5} + x_{1,6} \leq 4$

Iteration 6: $x_{2,5} + x_{2,6} + x_{2,7} + x_{2,8} + x_{2,9} \leq 4$

Task assignments: $x_{1,1} = 1, x_{1,2} = 1, x_{1,3} = 1,$
$x_{1,4} = 1, x_{2,4} = 1, x_{2,5} = 1, x_{2,6} = 1, x_{2,7} = 1,$
$x_{2,8} = 1, x_{3,8} = 1, x_{3,9} = 1, x_{3,10} = 1$

Maximum workload: $P^L_{max} = 41$.

Next, given the above solution x_{ij}, P^L_{max} of the loading problem **L**, the best assembly sequences u_s and assembly routes y_{iljs} (and the corresponding product assignments z_{ijs} (16)) have been determined by solving the routing problem **RS**. Direct application of an LP code to LP relaxation of **RS** yields an integer solution with the maximum workload $P_{max} = 41$ and total transportation time $Q_{sum} = 38$. The assembly sequence and route selected for each product are shown in Figure 3, where number of the station selected for each task to be performed is additionally indicated in parentheses.

$k = 1, \; s = 1:$

$L \rightarrow 1(1) \rightarrow 2(1) \rightarrow 3(1) \rightarrow 4(1) \rightarrow 6(2) \rightarrow 8(2) \rightarrow U$

$k = 2, \; s = 3:$

$L \rightarrow 1(1) \rightarrow 2(1) \rightarrow 4(1) \rightarrow 5(2) \rightarrow 6(2) \rightarrow 7(2) \rightarrow 9(3) \rightarrow 10(3) \rightarrow U$

$k = 3, \; s = 5:$

$L \rightarrow 2(1) \rightarrow 3(1) \rightarrow 4(1) \rightarrow 5(2) \rightarrow 7(2) \rightarrow 8(3) \rightarrow 9(3) \rightarrow 10(3) \rightarrow U$

$k = 4, \; s = 7:$

$L \rightarrow 1(1) \rightarrow 3(1) \rightarrow 5(2) \rightarrow 6(2) \rightarrow 7(2) \rightarrow 8(3) \rightarrow 9(3) \rightarrow 10(3) \rightarrow U$

$k = 5, \; s = 9:$

$L \rightarrow 1(1) \rightarrow 3(1) \rightarrow 5(2) \rightarrow 6(2) \rightarrow 7(2) \rightarrow 8(3) \rightarrow 9(3) \rightarrow 10(3) \rightarrow U$

Fig. 3: Graph of assembly sequences and routes selected

The computational experiments with the two-level approach performed for a series of medium sized test problems have indicated that good solution results can be obtained in a short CPU time.

V. CONCLUSION

Flexibility in the short-term planning is an important issue in enhancing productivity of a flexible assembly system. An efficient utilization of the system capabilities requires alternative assembly plans for each product to be considered at the loading level. In order to achieve the best results, allocation of assembly tasks and selection of assembly sequences and assembly routes should be simultaneously determined for all products. The two-level approach with the linear relaxation-based heuristic for loading and a network flow model for routing and sequence selection seems to be applicable in practice for such a short-term allocation of the system resources.

ACKNOWLEDGMENTS

This research has been supported by KBN grant # 8 T11F 015 13.

References

[1] H.F. Lee and R.V. Johnson, "A line-balancing strategy for designing flexible assembly systems," *International Journal of Flexible Manufacturing Systems*, vol. 3, 1991, pp. 91-120.

[2] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization*. John Wiley, New York, 1988.

[3] T. Sawik, "Integer programming models for the design and balancing of flexible assembly systems," *Mathematical and Computer Modelling*, vol. 21, no. 4, 1995, pp. 1-12.

[4] T. Sawik, "Sequential loading and routing in a FAS," *Proceedings of Rensselaer's Fifth International Conference on Computer Integrated Manufacturing & Automation Technology*, Grenoble May 29-31, 1996, pp. 48-53.

[5] T. Sawik, "A two-level heuristic for machine loading and assembly routing in a FAS", *Proceedings of the 5th IEEE International Conference on Emerging Technologies and Factory Automation*, Kauai, Hawaii November 18-21, 1996, pp. 143-149.

[6] T. Sawik, "An interactive approach to bicriterion loading of a flexible assembly system", *Mathematical and Computer Modelling*, vol. 25, 1997, pp. 71-83.