Towards Physical Layout of Assembly Lines

B. REKIEK, F. PELLICHERO, P. DE LIT, E. FALKENAUER AND A. DELCHAMBRE
Department of Applied Mechanics, CAD Unit
University of Brussels (ULB)
Av. F. D. Roosevelt 50, CP 165/41
B- 1050 Brussels, BELGIUM

Abstract: **-** The scope of Concurrent Engineering has now worldwide recognition as being the method for the factory of the future. This concept permits to have the different levels of the design of flexible manufacturing systems interacted. In its simple form the design of a flexible production system requires to assign operations to a set of workstations (logical layout), and decide about the position of the workstations, conveyors (physical layout). A new method is introduced to solve the interrelated problems. We here present a new algorithm to treat the problem of the design of assembly systems (balance and architecture). The algorithm uses a Grouping Genetic Algorithm (GGA), based on an Equal Piles approach, and heavily modified to respect the precedence constraints between operations. The main concern is the quality of the resulting line in terms of balancing, and its suitability to the material flow requirements of the production system. The essential concept adopted by the method is described in this paper. Results of the approach are presented.

*Key-Words:*Physical Layout,Equal Piles,Grouping Genetic Algorithm,Assembly Lines. *CSCC'99* Proc.pp..4221-4226

1 Introduction

The success of many companies during the recent years can be attributed to the way they have managed the design and the operation stage of their systems. These working practices and tools adopted by companies to improve product development are known collectively as Concurrent Engineering (CE) [3]. Industrial practice of this tool is still being developed. Starting from the philosophy of divide and conquer, Cellular manufacturing (CM), which is an application of Group Technology (GT), introduced the grouping of tasks into cells or workstations. Processes and people are thereby assigned to cells responsible for manufacturing or assembly of parts or products. One tenet of group technology entails the division of the manufacturing facility into small groups of workstations, each cell being dedicated to a specified set of part types.

The central concern of the facility layout is the configuration of manufacturing facilities to facilitate the material flows and the execution of production plans. Today, small focused factories are created as independent operating centers within large facilities.

The tendency is due to the application of the theory of management based on the principle that similar things should be done similarly.

The planning can improve efficiency and save considerable effort and resources in facility location and allocation. The main questions are how to choose from the available locations those in which to install the facilities and then how to assign tasks to each facility to minimize the overall cost of operations.

We here propose an algorithm yielding a logical layout taking the topology of the line (facility) into account. This architecture represents a rough idea of the physical layout of the future line.

We are not concerned in this paper with fine tuning such as the specific position and angular orientation of worker's bench or location of the power outlets. The accent is put on the balancing of the workstations. The obtained clustering can serve as the input data of a more detailed physical layout module. Uncoupling the logical layout problem and the physical layout one makes the assembly line design less inefficient. So, the feedback from the balancing on the facility layout is of a great importance.

Background and motivation of the presented work is briefly described in section 2. In Section 3 we present the definition of our assembly line design problem. Our new approach to tackle the problem, which uses the Grouping Genetic Algorithm is described in section 4. We illustrate the method with an example in section 5, and we draw conclusions in section 6.

2 Prior research

The assembly line design (ALD) is well known as the elaboration of the logical and physical layout of the line. The logical layout consists in the distribution of operations among workstations along the line. The physical layout, decides about the disposition of the workstations, conveyor(s), etc. on the shop floor.

As shown in [2], and more recently in [13], despite hundreds of works on assembly line design, little number of companies use published techniques to balance their lines. One of the reasons is that the models usually adopted to solve the problem suffer from substantial loss of information. In fact, little work has been done on modeling the full range of practical considerations in assembly line design. Most of the time, the line balancing problems tackle linear assembly lines, without confluence of or separation into sub-lines. The common performance indices are the cycle time and the number of workstations, whereas other factors may also heavily affect the system performances. Some of these, such as traffic problems, workstation congestion [14] and transportation network are often considered to be marginal in the design stage of the assembly line.

Several very complete works were published about facilities planning [1, 6, 15, 16], but none seems to bridge the logical and physical layouting of the line. Most of the time flows are analyzed, but the planning is done at a department or factory level, not at the line level. Authors also tackled the cell formation problem in various ways [4, 9, 11] but these approaches are more focused on group technology (GT) and material flows, and are unable to deal with logical layout.

A global approach was a result of the SCOPES project [3]. It takes into account the main factors affecting the system performances. Lucertini *et al.* [10] present a unified framework for designing the configuration of a given production plant and the corresponding network of material flow.

3 Assembly lines design

Line balancing algorithms are generally suited to a unique linear assembly line, with possibly parallel workstations. The main idea behind our design of assembly lines philosophy is that, for complex products, the assembly system is most of the time decomposed into subsystems which are easier to manage than the entire one. The line is decomposed into several linked sub-lines (called workcenters in the remainder of this paper), with their own cycle time, reliability, stations requirements.

The routing of a product from one workcenter to another is fixed, as we work according to a line flow topology. But the main topology of the line is not necessarily a linear one. Some workcenters may serve to assemble a subassembly or module that is injected as a whole in a main line. Some stations, like packaging, may be used for several products in the same facility, and so are at the confluence of two lines. Different lines or workcenters are thus linked, yielding several line topologies. Fig. 1 illustrates our words. Four workcenters are linked to a main line according to a “fishbone” topology, and the main line separates into two other ones at its end.

With a classical line balancing algorithm, a way to tackle the line balancing problem would be to balance each workcenter separately. But in real conditions, some operations allocated to a given workcenter could be affected to another one, linked to the former.

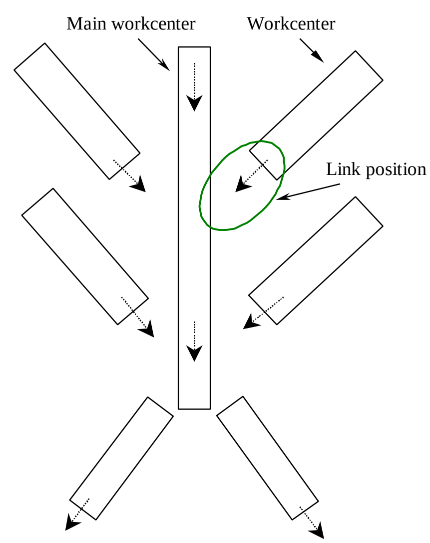

Fig. 1. Example of plant topology.

The design problem of organizing an assembly process into workcenters (a set of linked workstations) along a production plant is the well known facilities layout problem (logical and physical layout). The position of each workcenter is important, since it determines the costs of transportation and storage. Most research on ALD considers the physical layout problem after the line balancing. By separating the two problems, sub-optimal solutions are often obtained. Better solutions can be found be using the premises of the physical layout as a data for the logical layout. The main questions to consider are: which tasks should be grouped in the same workcenter, and how can we link the different cells to achieve a well balanced set of workcenters?

Our Assembly Line Design problem can be defined as follows:
Given a set of W directed non cyclic graphs $G_i = (T_i, P_i)$ (the nodes of T_i represent the tasks and the arc of P_i represent the precedence constraints) with a duration of each task $L_{i,j}$ (task length), assigned to each node T_i, and a constant N_i (number of workstation); given a set of links between graphs, (graphs can be connected by their ends or their beginning), can the nodes T_i of each graph be partitioned into the N_i subsets S_{mi} (the m-th station tasks) in such a way that: there exists an ordering of the subsets such that whenever two nodes in distinct subsets are joined by an arrow in G_i, the arrow goes from a higher ordered (earlier one) to a lower ordered (later one)?
It is easy to show that our ALD is a generalization of the Simple Assembly Line Balancing Problem (SALBP). The aim is the balancing of the given work centers using the different links between them.

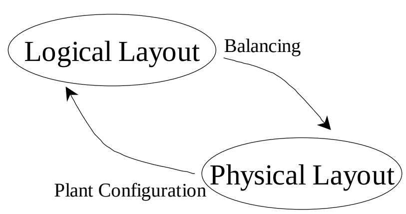

Fig. 2. Logical and physical layout interaction.

The proposed method is an iterative and interactive procedure which philosophy is illustrated at Fig.2. The results of the balancing module permit to know the distribution of the tasks and resources along the line, and the physical layout module thereafter determines the space requirements taking into account congestion and material storage, handling systems and so on.
The whole methodology can be described as follows:

1) set the desired workcenters, and for each of them
    - Assign tasks into workcenters, dealing with precedence graph,
    - set the desired number of stations,
    - set the desired cycle time,
    - set the preferences,
2) set the desired links between workcenters,
3) balance the whole plant (set of workcenters),
4) position workcenters and workstations,
5) check for congestion of the plant, analyze the flow of the products, the material handling problem, storage area requirements, and so on. Evaluate the efficiency of the corresponding plant layout using a simulation package,
6) if no satisfying solution is obtained, exchange the tasks (without violating precedence) and change the links between work centers.

The overall architecture of the balancing module is illustrated at Fig. 3.

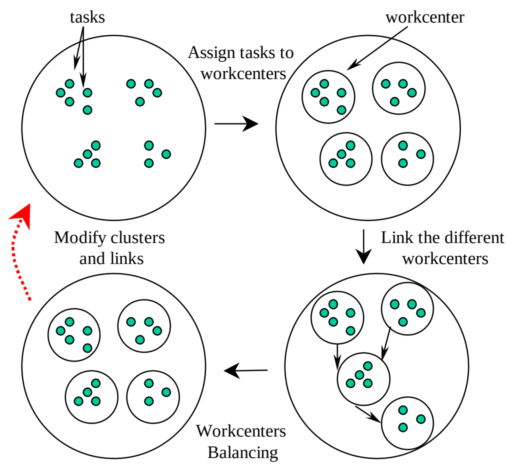

Fig. 3. Overall architecture of the balancing module.

4 The algorithm

4.1 Input Data

Our Logical Layout module needs the following input as illustrated on Fig. 4:

- for each workcenter
    - the desired number of workstations,
    - the desired cycle time,
    - the duration of each operation,
    - a precedence graph,
    - the user's preference constraints;
- possible links between work centers.

The different types of links between workcenters are (see Fig. 5):

- begin of WC1 linked to begin of WC2,
- begin of WC1 linked to end of WC2,
- end of WC1 linked to begin of WC2,
- end of WC1 linked to end of WC2.

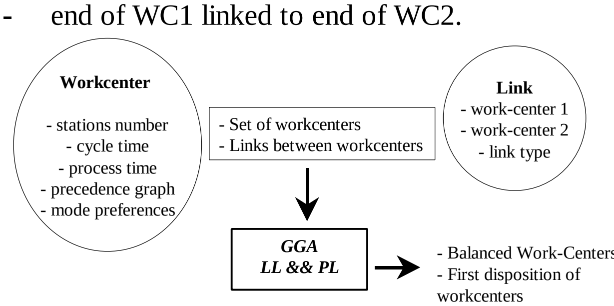

Fig. 4. Input data of the problem.

Note that the links are not mandatory: a workcenter may be isolated from the remainder of the line.

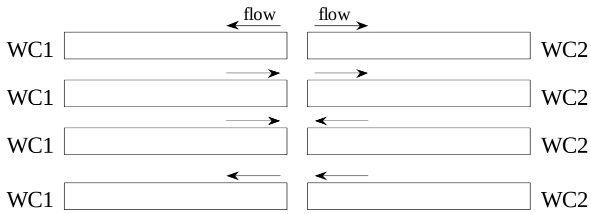

Fig 5. Possible links between workcenters.

The first set of data lets us balance locally a given workcenter (using only the tasks belonging to this workcenter) [12]. The second set (if it exists) globally balances the whole plant.
One of the industrial problems is the recovery of existing stations (special machines or robots) during layout changes. Two types of operations are introduced to deal with this kind of user's preferences:

- fixed operations on stations: some operations have to be fixed on a given workstation (control station, paint station,...), and no additional operation can be added to this station,
- linked operations: a set of operations must be grouped on the same workstation (designer's preferences), but additional operations can be added in order to balance the given assembly line.

4.2 The Grouping Genetic Algorithm

The line balancing of the line is done thanks to a Grouping Genetic Algorithm (GGA). The GGA [5] differs from the classical GA [8], [7] in two ways. Firstly, a specific encoding scheme is used so that the relevant structures of grouping problems become genes in chromosomes. Secondly, special genetic operators are used to suit the new encoding scheme. Both of the aspects avoid the weakness of the standard GAs applied to grouping problems.
In order to assign operations to workstations, we use our EPAL (Equal Piles in Assembly Lines) heuristic [12]. The hard constraint is the fixed number of workstations (piles). The approach to solve the problem is based on the so-called 'boundary-stones'. The main steps of this randomized heuristic can be summarized as follows:

1) the operations are ordered according to their number of predecessors and successors;
2) boundary stones (or workstation seeds) are chosen using the sequence obtained at step 1;
3) operations are grouped into as many clusters as stations;
4) a heuristic assigns operations to workstations, using the different clusters;
5) a multiple and simple wheel heuristic are used to equalize station loads by moving operations along the line or exchanging operations between workstations.

Two heuristics (applied separately to each workcenter) are used alternatively to improve the solutions obtained by the Boundary Stones algorithm: the simple wheel and the multiple wheels heuristic. Both of them will be executed on a solution until no improvement is obtained anymore, or a maximum number of trials is reached.

The simple wheel:
This heuristic moves sets of operations along the line (workcenter). Firstly we try to move a set of operations from the first station to the second one, then from the second one to the third one and so on. Next, the move begins with moves from the last station to the last but one and so on (Fig.5). This leads to move operations along the line to stop gaps in stations (precedence constraints are still respected).

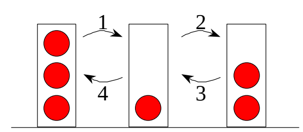

Fig.5 Simple wheel heuristic.

The multiple wheel:
The second idea is to exchange operations between stations. Two adjacent workstations are taken each time (Fig.6). All possible exchanges (which do not violate precedence constraints and cycle time) are executed. This kind of moves permits to escape from local optima due to bad 'boundary-stones'.

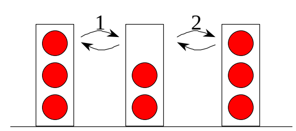

Fig.6 The multiple wheel heuristic.

The Linked wheels:
In order to take advantage of the links between workstations, a new heuristic has been developed. We call 'link node' the set of workstations by which a set of workcenters are linked. For instance suppose the link (end(WC1), end(WC2)), which means that the end of the workcenter 1 is linked to the end of the workcenter 2. The link node will be the last workstation of each workcenter.
Two stations in the link node are chosen (Fig.7) and all possible exchanges between them (which do not violate precedence constraints and cycle time) are executed. This kind of moves permits to balance two

adjacent workcenters by exchanging tasks between them.

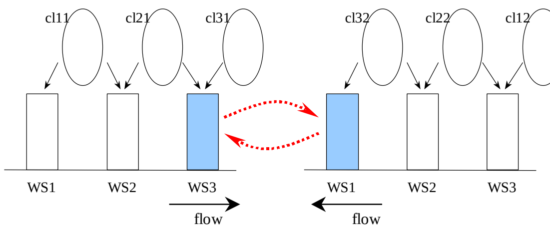

Fig.7 Linked wheels heuristic.

4.4 The cost function

The objective is to equalize workstations duration, under a fixed number of workstation constraints. Simply summing the differences (positive or negative) between the workstations operating time and the desired cycle time lacks any capacity of guiding the algorithm in its search. We thus settled for the following cost function: minimize

$$f_{EP} = \sum_{j=1..W} \left(\sum_{i=1..N_j} \left(\left| fill_i - cycletime_j \right| \right)^{1/2} \right) \qquad (1)$$

where W is the number of workcenters, N_j the number of workstations of each workcenter, $fill_i$ the sum of working times on workstation i, and $cycletime_j$ the desired cycle time of workcenter j, defined as follows (2):

$$cycletime_j = \frac{\sum_{j=1..nbop_j} time_i}{N_j} \qquad (2)$$

5 Case study

The case study we propose is adapted from one of the problems proposed in the Line Balancing Benchmark suite of Scholl [21]. The benchmark considers 29 tasks with precedence constraints and operating times illustrated at Fig. 8. We chose to create two workcenters, with the link (end(WCB), begin(WCA)).

We first balanced the workcenters without link between them. The results, presented at Table°1, represent the best solution for a given number of stations without cycle time restriction, according to an Equal Piles strategy. WC is the workcenter taken into account, and N the number of stations. As can be seen, if the cycle time of the line is not close to 50 (yielding respectively 2 and 5°stations), the line will poorly be balanced. The last three lines present the results for linked workcenters.

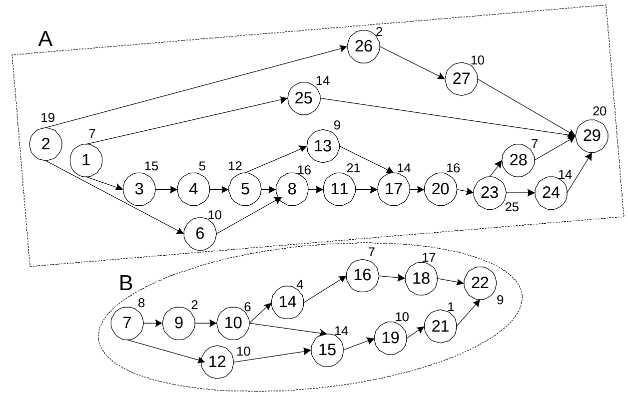

Fig. 8 : Precedence graph of the problem.

WC, N	Stations loads
A, 4	56, 58, 62, 60
A, 5	46, 49, 45, 48, 45
A, 6	41, 38, 40, 35, 41, 41
B, 2	44, 44
B, 3	30, 31, 27
B, 4,	20, 24, 24, 20
A, B, 6	55, 55, 55, 53, 55, 51
A, B, 7	44, 44, 46, 49, 45, 48, 45
A, B, 8	37, 41, 37, 40, 42, 45, 41, 41

Table 1. Results of the algorithm, with and wihout link between workcenters.

By adding the link between the workcenters, the whole line may be well balanced. The composition of the linked stations is given at Table 2. Note that the operation exchange between workcenters is only allowed at the node. Operations from workcenter A mixed with some of workcenter B were written in bold font in the table.

N	Stations description
6	{7,9,10,12,14,15,19,21}, {1,3,**16**,**18**,**22**}, {2,4,5,6,13}, {8,11,25,26}, {17,20,23}, {24,27,28,29}
7	{7,9,10,12,14,15}, {16,18,19,21,22}, {1,3,4,5,13}, {2,6,8}, {11,17,26,27}, {20,23,28} {24,25,29}
8	{7,9,10,12,14,16}, {15,18,19}, {**21**,**22**,1,3,4}, {2,5,13}, {6,8,25,26}, {11,17,27}, {20,23}, {24,28,29}

Table 2. Station compositions for linked workstations.

6 Conclusion

The balancing of assembly lines is most of the time uncoupled from the facility layout problem. This yields suboptimal line layouts. We proposed in this paper an iterative procedure partially treating the two problems simultaneously. We first split tasks between the desired workcenters (we set a number of links between them). We then balance the given workcenters. The designer will choose a well balanced architecture having the well balanced workcenters, and the manageable transportation network needed to satisfy all the material flow requirements.

The presented method is based on the Grouping Genetic Algorithm (GGA). The bin filling philosophy being very different from classical line balancing problem one, special heuristics were developed to tackle the problem: the 'boundary stones' and 'wheel' and 'linked wheels' heuristics.

Further research would be to develop an integrated method to tackle the physical layout of assembly lines problem. The influence of the workcenters clustering method will be analyzed and tested on industrial cases.

Our ALD method presented in this paper will be installed at the Resource Planning Department of the Marine Power company Liege, Belgium.

7 Acknowledgement

This paper is based on results of the project "Outils d'aide à la conception interactive des produits et de leur ligne d'assemblage". This project is made in collaboration with the Université Catholique de Louvain (UCL), the Faculté Polytechnique de Mons (FPMs), and the Belgian Research Center for the metalworking industry (CRIF). We particularly thank the "Région Wallonne" which has funded this project. We also would like to thank MARINE POWER Company, Liege (Belgium), for its collaboration in the computational validation of the tool.

References:

[1] Askin, R.G., and Standridge, C.R., *Modeling and Analysis of Manufacturing Systems*, John Wiley & sons Inc., 1993

[2] Baybars, I., "A survey of exact algorithms for the simple assembly line balancing," Management Science, No. 32, 1986, pp. 909-932.

[3] Delchambre, A., (ed), *CAD Method for industrial Assembly: Concurrent Design of Products Equipment and Control Systems*, Wiley & Sons Inc, 1996.

[4] De Lit, P., Delchambre, A., and Falkenauer, E., "Grouping Genetic Algorithms: an efficient method to solve the cell formation problem," to be published in *Special Issue of the IMACS International Journal Mathematics and Computers in Simulation*, 1999.

[5] Falkenauer, E., *Genetic Algorithms and Grouping Problems,* John Wiley & sons Inc., 1998.

[6] Francis, R. L., McGinnis, L. F., White, J. A., *Facilities Layout and Location: an analytical approach*, Prentice Hall International Series in Industrial and Systems Engineering, 1996.

[7] Goldberg, D., E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[8] Holland, J., H., *Adaptation in Natural and Artificial systems*, University of Michigan Press, Ann Arbor, 1975.

[9] Kusiak, A., and Chow, W. S., "Decomposition of manufacturing systems," *IEEE Journal of Robotics and Automation*, vol. 4, no. 5, 1988, pp. 457-471.

[10] Lucertini, M., Pacciarelli, D., and Pacifici, A., "Modeling an assembly line for configuration and flow management," *Computer Integrated Manufacturing systems*, Vol. 11, No.1-2, 1998, pp. 15-24.

[11] Miltenburg, J., and Zhang, W., "A comparative evaluation of nine well-known algorithms for solving the cell formation problem in group technology," *Journ. Op. Management*, vol. 10, no. 1, 1991, pp. 44-72.

[12] Rekiek, B., De Lit, P., Pellichero, F., Falkenauer, E. and Delchambre, A., "Applying the equal piles problem to balance assembly lines," submitted to the 1999 *submitted to the IEEE Int. Symposium on Assembly and Task Planning*, 1999.

[13] Rekiek B., Falkenauer E. and Delchambre A., Two problems in design and operation of assembly lines: line balancing and ordering variants, *Proceeding of the 14th Conference of ISPE, Narosa Publishing House, India*, 1998, pp. 243-250.

[14] Rekiek, B., Pellichero, F., De Lit, P., Falkenauer, E., and Delchambre, A., "A resource planner for hybrid assembly lines," *submitted to the 15th Conference of ISPE, Brazil, 1999.*

[15] Sule, D.R., *Manufacturing Facilities Location, Planning and Design*, PSW Publishing Company, 1994.

[16] Tompkins, J. A., White, J. A., Bozer, Y.A., Frazelle, E.H., Tanchoco, J.M.A, Trevino, J., *Facilities Planning*, John Wiley & sons Inc., 1996.