A Resource Planner for Hybrid Assembly Lines

Brahim Rekiek, Fabrice Pellichero, Pierre De Lit,
Emanuel Falkenauer and Alain Delchambre

Applied Mechanics, CAD Unit, University of Brussels (ULB)
Av. F. D. Roosevelt 50, CP 165/41, B -1050 Brussels, Belgium
brekiek@ulb.ac.be

Abstract

This paper presents a new method to address the Hybrid Assembly Line Balancing Problem with multiple objectives. The aim is to assign a set of tasks to workstations and select the assembly equipment or resource to perform each of them. The goal is to minimize the total cost of the line by integrating design (congestion, machine real cost...) and operation issues (cycle time, precedence constraints and availability...). We used a Grouping Genetic Algorithm to tackle the problem, hybridized with an Branch & Cut and the multi-criteria decision-aid method Promethee. We present the optimization procedure that assigns tasks to workstations and selects assembly equipment for each workstation. The essential concepts adopted by the method are described. The first results of the algorithm are illustrated by a case study.

Keywords: Hybrid assembly line, multiple objectives, grouping genetic algorithm, equal piles, branch & cut, Promethee.

1 Introduction

The objective of modern assembly systems is to produce high quality and low cost products. One of the innovations in the history of assembly manufacturing is the division of the assembly process into tasks of relatively limited content. Thus the skills needed to accomplish it can be developed in a short time. Paced assembly lines are widely used for medium to high production rate.

A line design often has a complex structure due to multiple line components, such as tools, material handling facility and so on. For a single component number of design alternatives may exist. The problem can easily become unmanageable if the line designer has to consider all the possible combinations of these alternatives. The problem must therefore be handled with a structural approach. For a given product and a given manufacturing environment the design objectives and constraints should be defined. The problem may be then subdivided into a number of linked sub-problems. Since the scope of each sub-problem is limited, a complete analysis becomes possible. Results of the analysis should help to reduce the number of alternatives at the sub-problem level and consequently simplify the overall design problem.

The main task of the resource planner is to design a flexible assembly system that will be able to assemble a product at least cost. For manual assembly lines the global cost of the line is directly influenced by the number of workstations. So the main objective of the classical Line Balancing was to minimize the number of workstations used in the line.

We deal with the design of assembly lines where the operations can be executed either manually, by robots or by hard automated equipment. These lines are called Hybrid Assembly Lines (HAL). In this case, the above reasoning is not valid anymore. In general, the operating cost and time will depend from the resource used. Given a list of candidate equipment available to complete the operations, the design problem thus becomes to decide which resource types to select (operating mode) and which tasks to assign to each resource in order to meet the production requirements at minimum total system cost.

The remainder of this paper is structured as follows. We briefly review work related to ours in section 2. Section 3 is devoted to the explanation of the Resource Planning Problem we tackle and section 4 to the description of the algorithm we propose to solve it. A case study will be presented at section 5. We draw conclusions at section 6.

2 Related work

Related design problems and issues are characterized in the literature as the assembly line balancing problem. Most papers deal with a single objective: minimize the idle time of the assembly line. Surveys on the subject are found in Baybars (1986).

Graves and Withney (1979) presented one of the first resource planning approaches based on linear programming techniques. The method does not deal with precedence constraints. Pinto *et al* (1983) proposed an integer programming formulation hybridized with the Branch & Bound algorithm. Holmes (1987) proposed an optimization procedure to solve a multi-equipment selection problem for assembly design. The method seeks to assign tasks to workstations and selects assembly equipment for each of them. An interactive and iterative method is presented in Nevins *et al.* (1989), based on technical and economical considerations which lead to the cheapest technically feasible assembly line. Its weakness is that the assembly sequence is fixed before the application of the method. This could be too restrictive and lead the system to miss the most cost-effective solutions. Lee *et al.* (1991) proposed an iterative method based on integer programming, depth-first branch and bound and queuing network analysis. The method minimizes several objectives: the cost of work-in-process inventory, machine investment and maintenance, and material handling. The proposed method allows to deal with assembly systems with single machine or identical parallel machines on each workstation. Falkenauer (1997) proposed a Resource Planning tool based on a Grouping Genetic Algorithm and a Branch & Bound algorithm to balance assembly line at cheap cost.

McMullen (1998) presented a simulated annealing method to address the assembly line balancing with multiple objectives. Different weights are attributed to objectives to lay stress on the favorite criteria.

Several applications of the evolutionary algorithms in the field of multi objective optimization problems have been reported in the literature. To deal with these problems, a lot of mathematical programming techniques have been developed. Number of authors use the popular weighted-sum approach. In the late eighties, Goldberg (1989) published his method called non-dominated sorting, and search techniques started to use the concept of Pareto (non dominant) optimality through selection and ranking methods.

Because evolutionary algorithms (EA) require scalar fitness information on which to work, the objectives are often artificially combined or aggregated into a scalar function according to some understanding of the problem. Schaffer (1985) was probably the first to recognize the possibility of exploiting Evolutionary Algorithms to treat multi objective problems. His approach was to use an extension of the simple genetic algorithm (called Vector Evaluated Genetic Algorithm 'VEGA'), and that differed from the standard one in the way the selection was performed. At each generation a number of sub-populations were generated by performing proportional selection according to each objective function in turn. Hajela and Lin (1992) included the weights of each objective in the chromosome, and promoted their diversity in the population through fitness sharing. The goal was to be able to simultaneously generate a family of Pareto optimal designs corresponding to different weighting coefficients in a single run of the GA. Besides using sharing, they used a vector evaluated approach based on 'VEGA' to achieve their goal.

Fonseca (1995) proposed a method called multi objective ranking. The rank of a individual corresponds to the number of chromosomes in the current population by which it is dominated. All non-dominated individuals are assigned rank 1, while dominated ones are penalized according to the population density of the corresponding region of the trade-off surface (the surface produced from the solutions evaluated during the run). The population is sorted according to the multi objective rank.

In the next section we introduce our method to treat the Resource Planning for Assembly Line problem. We will introduce a Multi-Objectives Grouping Genetic Algorithm (MO-GGA), based on the Equal Piles approach. The concern is the quality of the resulting line in terms of balancing and cost.

3 Hybrid assembly line optimization

In formal terms we defined the Hybrid Assembly Line Problem as the following decision problem. Let $G = (T, P)$ be a directed non-cyclic graph, the nodes of T representing the tasks and the arrows of P the precedence constraints between the tasks. Each node T_i is characterized by a set of couples $\{L_{i,j}, C_{i,j}\}$ ($L_{i,j}$ is a possible duration of the task and $C_{i,j}$ the cost of the corresponding resource used). Let N (number of workstations), CT (cycle time), and C (cost) be three constants. We define the cost of a subset of T as the sum of $C_{i,j}$ of the operations belonging to this subset. Is it possible to find a partition of N subsets of the set of operations and for each of them select a couple ($L_{i,j}$, $C_{i,j}$) so that the sum of $L_{i,j}$s in a partition is less or equal to CT, and the sum of all subsets costs less or equal than C, and so that there exists an ordering of the subsets in such a way that whenever two nodes in distinct subsets are joined by an arrow in G, the arrow goes from a higher ordered (earlier one) to a lower ordered (later one) ?

HALP is a variation of the Simple Assembly Line Balancing (SALB). The only difference is the way to treat the problem, the aim here is to minimize the cost of the assembly line. As far as an available algorithm for the HALP is concerned, we are not aware of any polynomial approximation similar to those known for the Bin Packing Problem (BPP). In the remainder of this paper we will not make the difference between this problem and its optimization NP-hard pendent.

Line performance is determined by other parameters than its cost, such as idle time on stations, workstation availability which depends on its reliability, space required by the layout, congestion, and so on.

Our Resource Planning (RP) problem is defined as follows. Given a set of tasks, and for each task a set of possible resources each characterized by its price, reliability, surface penalty and speed in terms of the resulting duration of the task; given a cycle time and possible precedence constraints, we try to find:

– the resources to be allocated to each task among the possible ones for that task,

– an assignment of the tasks to workstations along the line;

so that:

– no workstation requires more time than the cycle time to perform all the tasks assigned to it using the resources allocated to each of the tasks,
– no precedence constraints are violated,
– the following objectives are met:
  – total price of resources allocated to tasks is as less as possible,
  – a maximal reliability of the line is attained,
  – the surface occupied by the equipment avoids congestion problems.

4 The algorithm

The algorithm we propose is a Grouping Genetic Algorithm (GGA). It is subdivided into two levels. The first one distributes the tasks among the workstations, the second one handles the selection of workstation equipment.

4.1 Input data

Our hybrid assembly line algorithm needs the following input as illustrated on Figure 1:

– the desired number of workstations,
– the desired cycle time,
– the durations for each operation,
– the precedence constraints between operations,
– the user's mode preferences for each operation (manual, automated and robotized),
– an equipment database which yields the features of the different resources (cost, reliability, operating time). The cost of a resource is calculated over the expected lifetime of the line. It can be any of the following:
  – purchase price plus exploitation costs,
  – cost of manpower, etc.,
  – a combination of manpower cost and equipment cost.

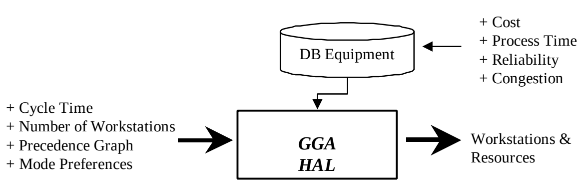

Figure 1: Data flow for hybrid assembly lines.

4.2 The Overall Method

The line balance efficiency is impacted by the number of workstations in the line and the idle time. Normally, the fewer the number of stations in the line and the less the idle time, the more efficient the line. One of our aims is to minimize the whole cost of the line. Given the fact that faster resources are more expensive (and cheaper resources are slower), the cheapest line can present fast and slow equipment together and can feature a small or a high number of workstations. Thus we have to decide which task will be performed on which workstation but we also must select the resource allocated to each of them. There is an important link between the two stages. We propose the following algorithm to generate possible solutions of the problem.

Individual Construction Algorithm (ICA):

1) assign tasks to the workstations (using the operating time corresponding to the fastest equipment) according to an Equal Piles strategy (see section 4.3.2),
2) generate all possible resource combinations for each station thanks to a Branch & Cut algorithm,
3) select the best equipment combination for each station using the decision-aid method Promethee;

This algorithm will be applied each time we have to construct or complete the construction (during crossover for example) of an individual in the GGA.
The GGA steps are the following:

1) Create a population of individuals using the Individual Construction Algorithm
2) Use the decision-aid method Promethee to order individuals in the population;
3) Recombine (mate) best individuals (parents) to produce children (with use of the ICA);
4) Mutate children (with use of the ICA);
5) Use Promethee to order the new population;
6) Replace the worst individuals of the population by the new children;
7) If a satisfactory solution has been found stop. Else go to 3).

4.3 Distributing tasks among workstations

4.3.1 The Grouping Genetic Algorithm (GGA)

The Grouping Genetic Algorithm (GGA) Falkenauer (1998) differs from the classical GA Holland (1975), Goldberg (1989) in two ways. Firstly, a specific encoding scheme is used so that the relevant structures of grouping problems became 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 Falkenauer (1998).

4.3.2 Equal Piles Algorithm

In order to assign operations to workstations, we use our EPAL (Equal Piles in Assembly Lines) heuristic Rekiek (1999). The hard constraint is the fixed number of workstations (piles). At the stage of operations assignment to workstations, we use a minimum cycle time *min_ct*. This *min_ct* is the ratio between the sum of minimum process times of the operations and the desired number of workstations. 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 workstation loads by moving operations along the line or exchanging operations between workstations.

4.3.3 Mode Preferences

The heuristics must deal with the mode preferences of operations (manual, robotized, or automated), yielding grouping constraints. The SAM software Pellichero (1999) proposes one or several possible modes for feeding, handling and insertion of each element (part or subassemblies) of the product. The resulting constraints impeach manual operations to be grouped with robotized or automated ones. Note that if several modes are allowed for an operation, the mode will be fixed by the GGA to yield the best logical layout.

4.4 Selecting the equipment

4.4.1 The Branch and Cut Algorithm

As its name implies, the Branch & Cut method consists in two fundamental procedures: branching and cutting. The search procedure may be represented as a tree, the root symbolizing the whole problem. Branching is used to divide a large problem into two or more sub-problems usually mutually exclusive. A branch is associated to each sub-problem. These can be partitioned in a similar way, yielding new branches and so on. Cutting permits to stops partitioning of non valid sub-problems. The associated branches are not further developed. The examination of the search tree associated to the partitioning process stops if it represents only one solution, or if it can be shown that the node does not contain an optimal solution.

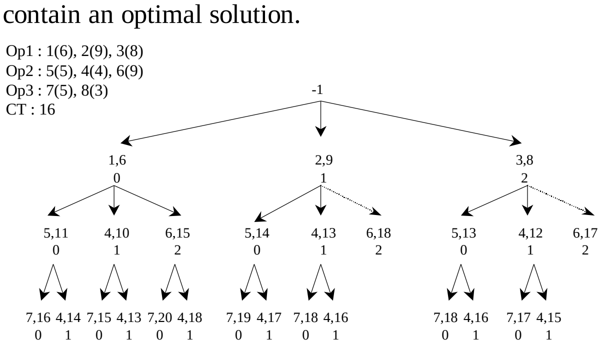

Figure 2: Tree generated by the B&C Algorithm.

We used a Branch & Cut algorithm to assign equipment to operations. Initially, the tree contains only one node, the first task (level 0). The branching is done by developing a set of possible resources for the given task. Each node corresponds to an equipment and each level to one task. On the graphic presented at Figure 2 each couple (a, b) corresponds to equipment index, and the sum of operating times in that branch at this level. For example, the couple (5,11) means that by selecting equipment 5 to realize operation 2, the total operating time on the station is 11. At each level, all possible equipment for the given operation are generated but we only develop further the valid branches respecting the constraints of the problem (cycle time). For example, selection of equipment 2 for operation 1 and equipment 6 for operation 2 yields an operating time of 18. So this branch will not be further developed. Once all the levels (valid branch) have been developed, only valid solutions (the sum of process time of operations for the selected equipment must not exceed the cycle time) are kept. If there is no possible solution (the sum of operating times of the fastest equipment exceeds the cycle time), the fastest equipment will be selected. This may happen if the desired cycle time is incompatible with the fixed number of stations (this constraint will never be violated).

Due to the difficulty of comparison of solutions two by two, we settled for a multi-criteria decision-aid method Promethee Brans (1994). The different solutions found by the B & C algorithm serve as input data for the Promethee algorithm to choose the best equipment taking into account the different criteria. Afterwards, resources are assigned to each task of the given workstation.

4.4.2 Promethee

Selection of a solution from a set of possible ones on the basis of several criteria can be considered as a difficult and intriguing problem. Selecting equipment is a process with a number of uncertainties. One of our main concerns during the definition of the structure of this algorithm was to keep all valid selections during the search steps (assignments respecting all the constraints even if they seem not very good). We considered it important to allow selections to stay in the search until all the knowledge about the target installation was complete, even if they seemed at first glance.

A complete description of the theory related to the Promethee method is out of the scope of this paper. For more details about it, the reader is invited to refer to Pellichero (1999). It is however important to know that it computes a “net flow” (f) associated to each solution. This flow gives us a ranking, called the Promethee II complete ranking, between the different solutions of the problem. Here are the rules defining this ranking:

$$\begin{cases} aP^{II}b \quad iff \quad f(a) > f(b) \\ aI^{II}b \quad iff \quad f(a) = f(b) \end{cases}$$

It means that solution a is preferred to solution b if and only if $f(a) > f(b)$, and that solution a and b are indifferent if and only if $f(a) = f(b)$.

It is also important to know that a weight is associated to each criterion. These weights are involved in the computation of the f number and represent the relative influence of one criterion with respect to the other ones.

4.4.3 The Performance of the Solutions

In typical multi-objective optimization problems, there exists a set of solutions that are superior to other ones in the search space when all objectives are considered, but are inferior to the other solutions in the search space according to one or more objectives. These solutions are known as Pareto-optimal solutions or non-dominated solutions. The rest of the solutions are known as dominated ones. Since none of the solutions in the non-dominated set is absolutely better than any other, any of them is an acceptable one. The choice of one solution over the other requires problem knowledge. We told that a weight was associated to each criterion. It is the designer's task to adjust them to help the algorithm to find solution he considers being good ones.

Optimizing a combination of the objectives has the advantage of producing a single solution, requiring no further interaction with the decision maker. If this "optimal" solution cannot be accepted, due to inappropriate setting of the coefficients of the combining function (the weights of the criterions), new runs of the optimizer may be required to adjust them until a suitable solution is found.

We here present the criteria of comparison used during the selection of equipment.:

- process time: workstations should not require more than a cycle time to perform all the tasks,
- cost : the total price of the resources allocated to the workstations must be minimized,
- reliability: must be maximized on each workstation,
- congestion: is proportional to the space occupied by the workstation, and must be reduced.

The same multi-criteria decision-aid method Promethee will be used to compare the potential solutions the genetic algorithms proposes. Note that the solutions are not compared thanks to a cost function yielding the fitness of the individuals. They are compared to each other thanks to flows in Promethee, depending on the current population. The value of these flows is context related and has no absolute meaning. So it becomes impossible to fix a stop criterion for the GGA. The optimization is stopped at user's request, or if no better solution has been found for a given number of generations.

5 Case study

The case study we propose is adapted from one of the problems proposed in the Line Balancing Benchmark suite[1] of Scholl (1995). The benchmark presented here is called 'Buxey'. It considers 29 tasks with precedence constraints illustrated at figure 3. We proposed three possible equipment (and operating times) for each operation. We considered that all equipment had the same reliability (99 %) and same congestion factor (1).

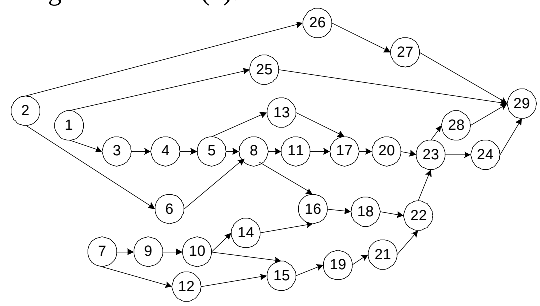

Figure 3 : Precedence graph of the problem.

We tested our algorithm for several numbers of stations (N) and several desired cycle times (C). The results of the GGA are presented at Table 1. We showed the total cost of the line (arbitrary units), and the loads of the stations (ratio between the sum of operating times and the cycle time). Note that the station load is superior to 1 in some cases, meaning that the desired cycle time cannot be held for the selected number of stations. As can be seen, the line will generally be less expensive as the cycle time constraint is relaxed (cycle time augments). But this is not always the case. For example, for seven stations, the cost raises as the cycle time augments. This is because the cost is not the only criterion taken into account. The line is better balanced for higher cycle times, for slight differences of the station cost. So the given solutions for cycle time fixed to 39 or 40 are considered to be better ones than those obtained by distributing the tasks and equipment the same way as for cycle time equal to 38 (which is less expensive).

N,C	Cost	Stations loads
6, 44	3340	1.00, 1.00, 1.05, 1.07, 1.05, 1.00
6, 45	3340	1.00, 1.00, 1.00, 1.00, 1.00, 1.00
6, 46	3280	1.00, 1.00, 1.00, 1.00, 1.00, 1.00
7, 38	3230	1.00, 1.05, 1.03, 0.97, 1.05, 1.08, 1.00
7, 39	3240	1.00, 1.03, 1.03, 1.00, 1.03, 1.00, 1.00
7, 40	3270	1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00
8, 34	3280	1.00, 1.00, 1.03, 1.00, 1.03, 1.00, 1.00, 1.00
8, 35	3240	1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00
8, 36	3030	1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00

Table 1: Results of the HAL balancing algorithm.

[1] The benchmark suite can be accessed via the Web at http://www.bwl.tu-darmstadt.de/bwl3/forsch/projekte/alb/index.htm

6 Conclusion and Further Work

In this paper, we presented a new method to treat the Resource Planning for Assembly Lines Problem. The method is based on a Grouping Genetic Algorithm (GGA), the multi-criteria decision-aid method and the Branch and Cut method. We developed a new approach to tackle multi-objective problems using the genetic algorithm. The aim is to select equipment to carry out the assembly operations. This method is integrated in the software package SELEQ (SELection of EQuipment), which is a user-friendly tool to design assembly lines at low cost. Special user preferences are taken into account by the proposed method. In the future, further research will be undertaken on multi-products resource planner for hybrid assembly lines.

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.

References

[1] Baybars, I., "A survey of exact algorithms for the simple assembly line balancing," *Management Science*, vol. 32, pp. 909-932, 1986.

[2] Brans, J.-P., Mareschal, B. "The PROMCALC & GAIA decision support system for multicriteria decision aid," *Decision Support Systems*, vol. 12, pp. 297-310, North-Holland, 1994.

[3] Delchambre, A., *CAD Method for Industrial Assembly Concurrent Design of Products Equipment and Control Systems*, John Wiley & Sons Inc., 1996.

[4] Delchambre, A., Pellichero, F., Falkenauer, E., and Rekiek, B., "Optimization of assembly line lay-out by computer-aided choice of assembly methods and line balancing," in *Proceedings of the FAIM 98 Conference*, pp. 691-702, 1998.

[5] Falkenauer, E., "A grouping genetic algorithm for line balancing with resource dependent task times," in *Proceedings of the Fourth International Conference on Neural Information Processing (ICONIP'97)*, University of Otago, Dunedin, New Zealand, pp. 464-468, November 24-28, 1997.

[6] Falkenauer, E., *Genetic Algorithms and Grouping Problems*, John Wiley & Sons Inc., First Edition, 1998.

[7] Fonseca, C. M., and Fleming, P. J., "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1-16, 1995.

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

[9] Graves, S.C., and Withney, D. E., "A mathematical programming procedure for equipment selection and system evaluation in programmable assembly," in *Proceedings of the 1979 IEEE Decision and Control Conference*, Ft. Lauderlale, FL, 1979.

[10] Hajela, P., and Lin, C. Y., "Genetic search strategies in multicriterion optimal design," *Structural Optimization*, vol. 4, pp. 99-107, 1992.

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

[12] Holmes, C. A., "Equipment selection and task assignment for multiproduct assembly system", MIT Operations Research Center SM Thesis, Cambridge, MA, 1987.

[13] Lee, H. F., Johnson, R. V., "A Line-Balancing Strategy for Designing Flexible Assembly Systems," *International Journal of Flexible Manufacturing Systems*, vol. 3, pp. 91-120, 1991.

[14] McMullen, P. R., and Frazier, G. V., "Using simulated annealing to solve a multiobjective assembly line balancing problem with parallel workstations", *Int. Jour. Prod. Res.*, vol. 36, no. 10, pp. 2717-2741, 1998.

[15] Nevins, J.L., and Withney, D.E., *Concurrent Design of Products and Processes*, Mc Graw-Hill, New York, 1989.

[16] Pellichero F., Rekiek B., Falkenauer E. and Delchambre A., "Interactive tool for an optimal equipment selection," To be published in *Proceedings of the 1999 International Conference on Flexible Automation & Intelligent Manufacturing (FAIM99)*, Tilburg, The Netherlands.

[17] Pellichero F., Delchambre A., Mareschal B., Falkenauer E. and Rekiek B. "An integrated tool for the choice of the assembly method and the design of the logical-layout of the line," to be published in the *Journal of Engineering Design and Automation.*

[18] Pinto, P. A, Dannenbring, D. G, and Khumalawa, B. M., "Assembly line balancing with processing alternatives: an application," M*anagement Science*, vol. 29, no. 7, pp. 817-830, 1983.

[19] 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 IEEE Int. Symposium on Assembly and Task Planning (ISATP99)*, 1999.

[20] Schaffer, J. D., "Multiple objective optimization with vector evaluated genetic algorithms". In *Genetic Algorithms and their Applications, Proceedings of the First International Conference on Genetic Algorithms*, pp. 93-100., 1985.

[21] Scholl, A., *Balancing and Sequencing of Assembly Lines*, Heidelberg: Physica, 1995.