DEALING WITH USERS'S PREFERENCES IN HYBRID ASSEMBLY LINES DESIGN

Brahim Rekiek, Pierre De Lit, Pellichero Fabrice[1], Thomas L'Eglise[2],
Falkenauer Emanuel[1], Alain Delchambre

CAD/CAM Department
Université Libre de Bruxelles (ULB)
Av. F. D. Roosevelt 50, CP 165/14, B-1050 Brussels, Belgium
Tel. + 32-2-650.47.66 Fax. + 32-2-650.27.10
E-mail: brekiek@ulb.ac.be

Abstract: This work presents a new method to address the Hybrid Assembly Line Design Problem with multiple objectives. The aim is to assign a set of tasks to stations and select the resources 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, availability...). We used a grouping genetic algorithm to tackle the problem, hybridized with a branch-and-cut algorithm and the multi-criteria decision-aid method Promethee II. We present the method that assigns tasks to stations and selects assembly equipment for each station. We introduce the way to deal with user's preferences in design problems. The essential concepts adopted by the method are described. The application of our algorithm to an industrial case study is presented.

Keywords: design, assembly lines, multiple objectives, grouping genetic algorithm, multi-criteria decision-aid, interactive.

1. INTRODUCTION

The success of many companies during the recent years can be attributed to the way they have managed the design of their systems. The working practices and tools adopted by companies to improve product development are known collectively as Concurrent Engineering (CE) (Delchambre, 1996). Designing a manufacturing system is a difficult mission that necessitates many decisions. In broad generalities, we must select a product, design it, produce it, and sell it. Numerous decisions must be made at each step, that affect the time and cost of product manufacturing. Managing the whole concept is hard to human beings.

CE is a network of involved organizations through upstream and downstream linkages. The different processes and activities produce a value in the form of services that are added to the whole process. Design of manufacturing systems involves the design of products, processes and plant layout before physical construction. The 'line layout' (LL) problem is known in the literature as logical and physical layout (Delchambre, 1996). In this work our emphasis is on the 'logical layout' where the aim is to assign tasks to a set of stations. It is decomposed in the literature as the Assembly Line Balancing (ALB) and the Resource Planning (RP) problems. The balancing, used especially for manual assembly lines, aims to balance loads of stations. For hybrid assembly lines the RP helps designers to find an

[1] Optimal Design, Spin Off of Université Libre de Bruxelles, 20 Rue de l'Industrie, B-1400 Nivelles, Belgium.
[2] Department of Mechanical Engineering Université Catholique de Louvain, 2 Place du Levant, B-1348 Louvain-la-Neuve, Belgium.

assignment of tasks to stations and an assignment of resources to each task. The main objective is to minimize the total cost of the line by integrating design (cost, congestion...) and tasks issues (cycle time, precedence constraints, availability...).

A line design problem often has a complex structure due to multiple components, e.g. tooling, operators, material handling facility, and so on. For a single component, a number of design alternatives may exist. The problem can easily become unmanageable if the designer has to consider all the possible combinations of these alternatives. Therefore, the problem must be handled with a structural approach. For a given product and a given manufacturing environment, the design objective and constraints should be defined. Many practical search and optimization problems are better posed as multiple objective optimization problems and ask for a compromise among conflicting objectives. Since it is difficult to replace designer's intelligence, experience and creativity, it is more important to provide him with a set of assistance tools (computer programs). These tools just do the tedious part of the job, i.e. investigate and propose several solutions and make the necessary evaluations. Based on this information, the designer tests some alternatives and makes his decision. Due to their difficulty (industrial problems are often NP-Hard ones), meta-heuristics methods are often used to solve this kind of problems. The resulting solutions are adjusted to create an acceptable design and the best resulting solution is implemented. Thus, the design process must be viewed as an iterative, generate and test process.

This paper is organized as follows. Background to our work is briefly described in section 2. In section 3 we present the integrated method to design assembly lines. Our new approach to tackle the DAL problem (dealing with user's preferences) which is based on the grouping genetic algorithm is described in section 4. An industrial case study is presented in section 5. We draw conclusions in section 6.

2. STATE OF THE ART

In related work on design problem solving we often find many terms for types of design. (Parsaei, 1993) provides an excellent set of articles, which address a number of important issues within CE. The various definitions of design agree, however, that it is concerned with the mapping of a specified function onto a structure or description of a structure. Usually, the designed structure also satisfies performance, resource, and other pragmatic constraints.

Real-world problems are, in general, multi-criteria ones. That is, the problems involve multiple often conflicting objectives to be met. In assembly line design problems, designers deal with objectives like line efficiency, line imbalance, cost, reliability, buffers, stations space (Chow, 1990). Several applications of the evolutionary algorithms in the field of multiple objective optimization problems (MOP) have been reported in the literature. Numbers of authors use the popular weighted-sum approach. In the late eighties, Goldberg 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 (Goldberg, 1989). (Schaffer, 1985) was probably the first to recognize the possibility of exploiting evolutionary algorithms to treat multiple objective problems. Since then, numerous approaches to solve MOP have appeared in the literature. For a good introduction to and historical overview of relevant MOP concepts, the authors suggest to read the work done by (Fonseca, 1995) and by (Veldhuizen, 1999).

More and more DAL works deal with several objectives: minimize the idle time and the cost of the assembly line. A good survey on the subject can be found in (Baybars, 1986) and (Scholl, 1999). (Malakooti, 1994) uses multi-criteria decision making for ALB problems where objectives are the number of stations, cycle time, buffer size. He uses an additive utility function based on decision maker's preferences (weights). (Holmes, 1987) proposed an enumerative optimization procedure to solve a multi-equipment selection problem for DAL. The method seeks to assign tasks to stations and selects assembly equipment for each of them. (Lee, 1991) proposed an iterative method based on integer programming, depth-first branch-and-bound and queuing network analysis. The method minimizes the cost of work-in-process, machine investment and maintenance, and material handling. The proposed method allows dealing with assembly systems with single machine or identical parallel machines on each station.
(Falkenauer, 1997) proposed a Resource Planning tool based on a Grouping Genetic Algorithm and a branch-and-bound algorithm to balance assembly lines 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.

3. THE MULTI-OBJECTIVE GROUPING GENETIC ALGORITHM

The genetic algorithms (Holland, 1975) are inspired from the evolution of species in Nature. They have been proved a successful optimization method for three main reasons (Goldberg, 1989):

- their flexibility to conjugate themselves with specific heuristics adapted to the given problem;
- the power of their genetic operations on the chromosomes to perform a global search rather than a local one in the solution space;

- their ability to be adapted to many kind of constraints, linear or not, and any kind of cost functions, continuous, discrete, single criterion or multiple objectives.

3.1. The Grouping Genetic Algorithm

The Grouping Genetic Algorithm (GGA) 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 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 (Falkenauer, 1998).

3.2. Fitness and Multiple Objectives Problems

In classical genetic algorithms, the individual's fitness is computed according to a cost function that leads to a scalar fitness value. When dealing with multiple objective problems, the aggregation of the several criteria into a unique value is often used. This leads to rather artificial cost functions and to a difficult tuning of the weight the designer wishes to associate to each criterion.

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. The authors used the multi-criteria decision-aid method Promethee II to deal with such problems (see section 4).

3.3. Construction Heuristic: The Equal Piles Method

In this work, design of assembly lines is our concern. A solution to a given instance of the problem is a set of grouped tasks in a set of stations. In order to assign tasks to stations, we use an EPAL (equal piles for assembly lines) heuristic. The hard constraint is the fixed number of stations (piles). The approach is based on the so-called ‘boundary-stones’. The main steps of this randomized heuristic can be summarized as follows:

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

More detail on the ‘Equal Piles for Assembly Lines’ method can be found in (Rekiek, 1999a).

4. HOW AND WHERE DOES THE USER INTERVENE?

In a more general setting and especially when we have to design an artifact, one has to deal with *user’s preferences*. Two kinds of preferences can be found in design problems. Preferences said to be on the contents of the obtained design, and preferences among a set of solutions. The first kind of preferences may be hard (they cannot be violated) or soft (the solution to the problem has to be as close as possible to the designer’s desire, but it is not a *sine qua non* condition). In contrast, the second preference deals with a set of objectives (goals) and arises each time we have to decide about the best solution among a set of valid ones, all of them more-or-less satisfying the first kind of preferences. This second problem has more to deal with how the decision-maker judges a set of solutions. This might involve assigning different utilities (or preferences) to different objectives and combining them into some figure of merit. The difficulty with the specification of one compromise decision lies in the assessment of weights of the aggregate utility function that reflects the parties’ power, intensity.

Applying GA to multi-objective problems addresses two difficult problems: (1) searching large and complex spaces and (2) deciding among multiple objectives. Little work has been done on the combined problem of searching large spaces to meet multiple objectives. Recently, many studies have implemented Pareto-based GA search to sample the entire Pareto-optimal set of non-dominated solutions (Fonseca, 1995). Only few researchers have suggested ways of integrating multi-criteria decision making and the GA search. The GA iteratively samples the tradeoff surface (Pareto solutions) while the multi-criteria decision making successively narrows the search.

Two pragmatic and classic strategies were applied with the traditional separation of search and multi-criteria decisions:

- first, make multi-criteria decisions to aggregate objectives, then apply GAs search to optimize the resulting figure of merit,
- conduct the GA search using different aggregations of the objectives in order to obtain a range of alternative solutions and then make a multi-criteria decision to choose among the reduced set.

The first method consists of adding all the objective values together using different weighting coefficients for each one of them. The weighting coefficients represent the relative importance of the objectives. This means that our multiple objectives optimization problem is transformed into a scalar optimization problem. The ‘weighting objectives’ method was the first technique developed for the generation of non-inferior solutions for multiple objective optimization.

The main drawback of this approach is the fact that it can use the sum of values of two totally different objectives (in our example it could sum the imbalance value with the reliability value), which makes no sense.

The second approach yields the Pareto frontier – a pareto surface in case two objectives. The idea of pareto optimization is to provide the DM with a representative set of solutions from (or near to) the pareto optimal front, so that he can see the actual trade-offs that have to be made in choosing a solution, rather than asking these to be fixed through assignment of weights beforehand. Such non-aggregated decision making is generally considered to represent 'best practice'. The problem is the number of solutions the DM has to choose among them. The human cannot easily decide among more than a few solution, and the pareto frontier most of the time is composed by many non-dominated ones.

Even if it is difficult to analyze the convergence of exact methods on well-defined problems, it is quite common to talk about it. Convergence then means the time needed to reach a best solution of a given problem on a given kind of machines. In contrast, while dealing with multiple objectives problems – it seems a rather difficult task to talk about convergence, since there is no common agreement on what the optimum really is. Indeed, to speak about an optimal solution one needs to define a neighbor solution and its distance from the optimum. The question is how to define the closeness of two points (solutions) in the case of multiple objective problem search space. Our approach is situated in the middle of the two cited approaches, a merge of a search and multi-criteria decisions is used. Indeed, in order to come out of the multiple objectives problem stated by the cost function, we use the multi-criteria decision-aid method called Promethee II. For more detail about it, the reader is invited to refer to (Brans, 1994). 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 in the population. The weights (associated with each criterion) are involved in the computation of the f number and represent the relative influence of each criterion. Thus, the solutions are not compared according to a cost function yielding an absolute fitness of the individuals as in a classical GA, but are compared to each other thanks to flows, depending on the current population. In order to avoid a drift towards locally optimal solutions, elitism is used, i.e. the best-ever solution takes part in the evaluation of the f flows.

The choice of one solution over the others requires problem knowledge. It is the DM's task to adjust the weights to help the algorithm to find good solutions. Optimizing a combination of the objectives has the advantage of producing a single solution, requiring no further interaction with the DM. If the solution proposed by the GGA cannot be accepted, because of inappropriate setting of the weights, new runs may be required to adjust them until a suitable solution is found. For given user's preferences and a given design problem we run the following multiple objective GGA:

```
Generate an initial population with the ICA[3];
Order individuals using Promethee II;
repeat
   Select parents;
   Recombine best parents from the population;
   Mutate children;
   Reconstruct individuals using the ICA;
   Use Promethee II to order the new population;
   Replace some or all of the population by children;
until a satisfactory solution has been found.
```

5. APPLICATION OF THE METHOD

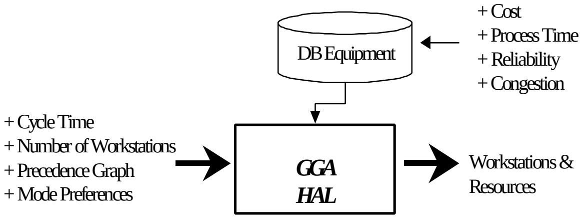

Fig. 1. Data flow for the hybrid assembly lines grouping genetic algorithm.

The presented design approach was applied to the resource planning of hybrid assembly lines. We used the following input as illustrated on Fig. 1:

- the desired number of stations,
- the desired cycle time,
- the durations for each task,
- the precedence constraints between tasks,
- the user's mode preferences for each task (manual, automated and robotized),
- an equipment database which yields the features of the different resources (cost, reliability, process 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.,
    - combination of operator and equipment cost.

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

- imbalance: the differences between the workloads on each station must be minimized.
- cost: the total price of the resources allocated to the stations must be minimized,
- availability: must be maximized on each station.

[3] Individual (problem solution) Construction Algorithm, which is problem dependent.

One of our goals is to minimize the whole cost of the line. Given the fact that faster resources are generally more expensive, the cheapest line can present fast and slow equipment together and can feature a small or a high number of stations. Thus we have to decide which task will be performed on which station 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 Individuals Construction Algorithm (for the GGA) to generate possible solutions of the problem (Rekiek, 1999b):

1) Assign tasks to the stations (using the operating time corresponding to the fastest equipment) according to the Equal Piles strategy (see section 3.3).
2) Generate all possible resource combinations for each station thanks to a branch-and-cut algorithm. The process time on a station should not exceed the cycle time, except if it is impossible to respect it, even when the fastest resources are selected to perform the tasks attributed to the station.
3) Select the best equipment combination for each station using the Promethee II method.

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 station. More details on the method can be found in (Rekiek, 1999b).

We settled for the given cost function:

$$\text{Cost function} = \text{Promethee}(\text{Cost}, \text{Imbalance}, \text{Reliability}, \text{Congestion})$$

where Promethee(x1,x2,…) means the multi-criteria decision-aid method among the objectives x1,x2… The imbalance *M* between the stations is defined as

$$M = Round\left[\sqrt{\sum_i \left(\frac{100*\left(FillWS_i - T_c\right)}{T_c}\right)^2}\,\right],$$

where $FillWS_i$ is the operating time of Workstation i and T_c is the cycle time.

The industrial case study we have chosen is a car alternator, which assembly line was implemented by FABRICOM some years ago. The cycle time is T_c =15 (arbitrary units, actually seconds).

Table 1 presents the data of this case study. The kind of operation (manual, automated, robotized) may be a user preference, but was here proposed using the approach presented in (Pellichero, 1997). This operating mode has an influence on the possible groupings, because manual operations will not be performed on the same station as robotized or automated ones. For each operation, a set of possible resources was determined; using the method described by (Pellichero, 1999). An indicative operating time is presented in Table 1. Of course the real operating time will be determined by the resource affected to a task. The predecessors of each operation are reported in the Preds column.

We applied the GGA for several user's preferences regarding the cost function. Fig. 2 presents the imbalance, the total cost of the line and the lower bound of the line availability according to the optimization strategy. Four cases were studied: a multi-criteria optimization, where all criteria are given the same importance, and three single criterion optimizations (respectively minimize the imbalance, minimize the cost, and maximize the availability). The results show that the proposed method respects the user's preferences regarding the optimization objective.

Op	Kind	Time	Preds	Op	Kind	Preds	Time
1	M	8	4	25	D	18	4
2	M	4	1	26	D	16	15
3	M	8	3	27	D	44	9
4	M	3	—	28	M	45	15
5	M	3	3	29	D	28	7
6	M	3	10	30	D	28	8
7	M	3	—	31	M	27	6
8	M	3	—	32	M	31	5
9	M	3	—	33	D	32	7
10	D	15	5,7,8,9	34	D	31	4
11	D	0	10	35	D	34	15
12	D	0	10	36	D	35	3
13	R	7	11,12	37	D	22	3
14	R	7	13	38	D	31	14
15	D	3	14	39	M	36,38, 46,47,48	3
16	D	9	15	40	M	39	10
17	R	16	7	41	D	26,37,40	14
18	M	6	4	42	M	41	15
19	M	17	8	43	M	42	15
20	D	19	8	44	M	29,30	3
21	D	20	5	45	D	23,24	14
22	M	21	5	46	D	35	4
23	M	17	5	47	D	35	4
24	M	17	9	48	D	35	4

Table 1: Operating mode, indicative operating time, and associated prececessors for each operation.

6. CONCLUSIONS 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 multiple objective grouping genetic algorithm (MOGGA), the branch-and-cut method the multi-criteria decision-aid method. The aim is to select equipment to carry out the assembly tasks. The accent is put on how to deal with the user's preferences in design problems. We show how the method can deal with the preferences, simply by adjusting the weight of the different objectives. We thus introduce a new paradigm to deal with multiple objectives using evolutionary computation methods.

In the future, further research will be undertaken on multi-products resource planner for hybrid assembly lines.

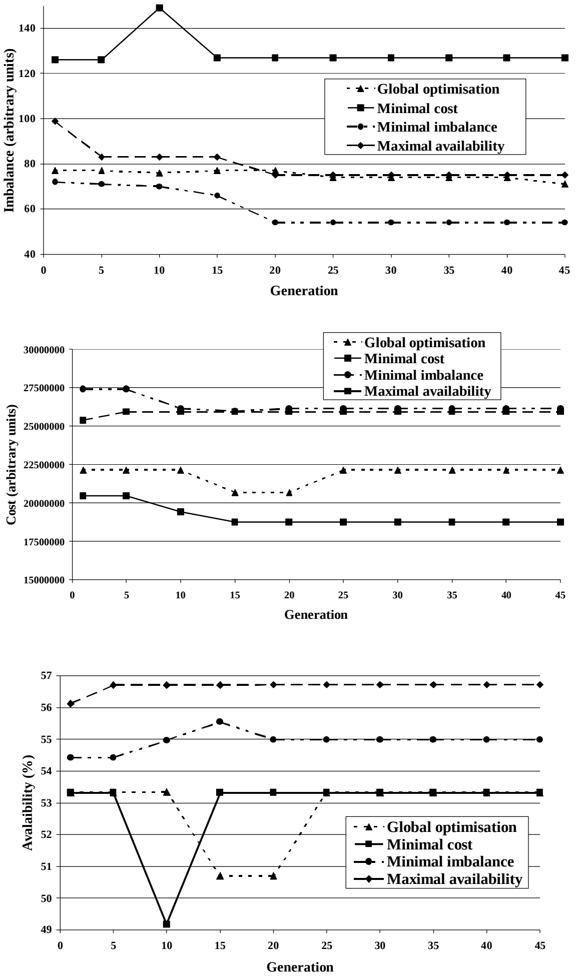

Fig. 2. Values of the several criteria according to the optimization strategy. For the global optimization each criterion is given the same importance.

REFERENCES

Baybars I. (1986). A survey of exact algorithms for the simple assembly line balancing. Management Science, vol. 32, pp. 909-932.

Brans J.-P. and Mareschal B. (1994). The PROMCALC & GAIA decision support system for multicriteria decision aid. Decision Support Systems, vol. 12, pp. 297-310, North-Holland.

Chow W-M. (1990). Assembly Line Design - Methodology and Applications. Marcel Dekker, New York.

Delchambre A. (1996). CAD Method for Industrial Assembly Concurrent Design of Products Equipment and Control Systems. John Wiley.

Falkenauer E., (1997). A grouping genetic algorithm for line balancing with resource dependent task times, in Proceedings of ICONIP'97, Dunedin, New Zealand, pp. 464-468, November 24-28.

Falkenauer E., (1998). Genetic Algorithms and Grouping Problems. John Wiley & Sons Inc.

Fonseca C.M. and Fleming P.J. (1995). An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation, Vol. 3, No 1, pp. 1-16.

Goldberg D.E., (1989). Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wessley.

Holland J.H., (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor.

Holmes C.A., (1987). Equipment selection and task assignment for multiproduct assembly system. MIT Operations Research Center, SM Thesis, Cambridge, MA.

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

Malakooti B., Kumar A., (1996). A Knowledge-Based System for Solving Multi-Objective Assembly Line-Balancing Problems. Int. J. Prod. Res., Vol. 34, n°9, pp. 2533-2552.

McMullen P.R., and Frazier G.V., (1998) 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.

Parsaei H.R. and Sullivan W.G., (1993). Concurrent Engineering: Contemporary Issues and Modern Design Tools, Chapman and Hall, England.

Pellichero, F. , Delchambre, A. and Mareschal, B. (1997). Interactive method for grouping operations in assembly line design. In *Proceedings of the FAIM'97 Conference*., pp. 111-122.

Pellichero F., Rekiek B., Falkenauer E. and Delchambre A. (1999). ‘Interactive tool for an optimal equipment selection’. in Proceedings of FAIM’99, pp. 313-323, The Netherlands.

Rekiek B., De Lit P., Pellichero F., Falkenauer E., Delchambre A., (1999a). ‘Applying the Equal Piles Problem to Balance Assembly Lines’, In Proceedings of ISATP, Portugal, pp. 399-404.

Rekiek B., Pellichero F., De Lit P., Falkenauer E., Delchambre A. (1999b). A Resource Planner for Hybrid Assembly Lines. In Proceedings of CARS&FOF'99, pp. MW6-18-MW6-23, Brazil.

Schaffer J.D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. Proceedings of the First ICGA, pp. 93-100. Lawrence Erlbaum.

Scholl A. (1999). Balancing and Sequencing of Assembly Lines. Heidelberg Physica.

Van Veldhuizen D.A. and Lamont G.B. (1999). Multiobjective Evolutionary Algorithm Research: A History and Analysis, Ph.D. Thesis.