CHAPTER 1

DESIGNING ASSEMBLY LINE

A task worth undertaking is one that adds value to the customer…

Thomas G. Schmitt
University of Washington

Keywords: assembly line design, logical line layout.

1. Introduction

The human *design* process is traditionally a time consuming, iterative approach. First, a preliminary design must be created, which is then analysed, experimented or tested in use, to determine its quality. The process of search and evaluation is repeated until the design is viewed as being acceptable. Computer aided-design (CAD) software and various computer simulation and analysis tools are widely used today. In contrast, automatic design or redesign processes, have been, until now less common. Many attempts have been made in the last few years to investigate the use of *semi-automatic* methods–the tedious part of the job is done by the computer, while the human tends to evaluate and select the best design. The recent design's success is due to the adaptive search techniques, in particular evolutionary search techniques such as genetic algorithms, evolution strategies, etc.

Assembly line are production systems composed of a succession of stations performing a set of tasks on products passing through them. They are the most commonly used method in a mass production environment, because they allow the assembly of complex products by workers with limited training, by dedicated machines and/or by robots. The main objective of assembly systems *designers* is to increase the efficiency of the line by maximising the ratio between throughput and the required costs. Assembly line design (ALD) involves the design of products, processes and plant layout before the construction of the line itself. These different modules interact at the different stages of ALD as shown in (De Lit, 1999). The current work is the result of a research project called CISAL carried out in collaboration with three universities, each one specialised in a particular step of the design of products and their assembly line. The FPMs (Faculté Polytechnique de Mons) specialised in the field of product analysis and design for assembly, while the UCL (Université Catholique de Louvain) was more focused on the field of the selection of the operating modes and techniques and the ULB (Université Libre de Bruxelles) was concentrated on the assembly line layout.

The product analysis proposes a first product design review, based on the classical DFA rules and precedence constraints between assembly tasks. The operating modes and techniques module proposes an assembly technique and the possible modes (manual, automated, robotic) for each task. The line layout (LL) module, assigns tasks to a set of stations, and decides on the position of stations and the resources on the plant floor (Figure 1.1).

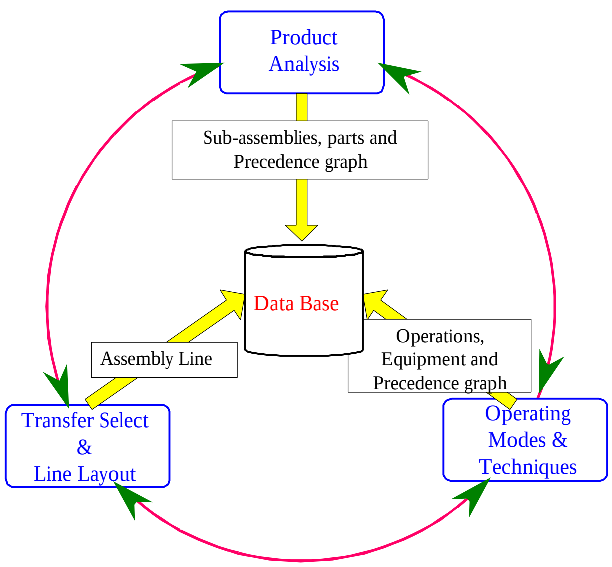

Figure 1.1. Methodology and information flow of the ALD (De Lit, 1999).

Before continuing the discussion about the assembly line design it is perhaps more desirable to define the assembly line concept. Indeed, the design of efficient assembly (or production) workshops is a problem of considerable industrial importance. A production workshop can be set up according to various topologies like cells (islands), combination of several lines, process layouts (isolated stations dedicated to a given process). As mentioned above, assembly lines are production systems composed of a succession of stations performing a set of tasks on the product passing through them. The assembled product takes shape gradually, starting with one part (usually called the base part), the remaining parts being attached at the various stations the product visits. A paced assembly line is a usual topology for medium and high production volume (cycle time varying from several seconds to several minutes) (Delchambre, 1996). In general, for simple products a unique linear assembly line, with possibly parallel stations can do the job. For complex products, the assembly system is most of the time decomposed into sub-systems which are easier to manage than the entire one. The line is decomposed into several linked sub-lines (called workcenters), with their own cycle time, reliability, and stations requirements.

The success of many companies during 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 their products development are known collectively as concurrent engineering (CE). Designing a manufacturing system is a hard task that necessitates many decisions. In broad generalities, we must select a product, design it, produce it, sell it, etc. Numerous decisions that affect the time and cost of the product must be made at each step. Managing the whole concept is difficult for human beings. The CE is a network of involved organisations 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. The main aim of the CE is to integrate product and process development in order to reduce the design lead time and to improve its quality and cost.

The line layout problem is known in the literature as logical and physical layout (Delchambre, 1996). The elaboration of the *logical layout* of the line consists in the distribution of tasks among stations along the line, while the *physical layout* of the line decides on the disposition of the stations, resources, conveyors, buffers, etc. on the shop floor. There is an serious interaction between the logical and physical layout.

In this study our emphasis is on the 'logical layout' where the aim is to assign tasks to a set of stations and selects assembly equipment for each of them. We are not concerned with fine tuning such as the specific position and angular orientation of worker's bench or location of the power outlets.

In this book the expression 'design of assembly line' or 'assembly line design' means, the proposition of a logical layout of the assembly line.

The logical line layout is composed of *assembly line balancing* (ALB) and *resource planning* (RP) problems (see Figure 1.2). In this study, the balancing used especially for manual assembly line aims to balance the stations workloads. For hybrid assembly line (where the operations can be executed either manually, by robots or by automated equipments) the RP helps designers to find an assignment of tasks to stations and an assignment of resources to each task.

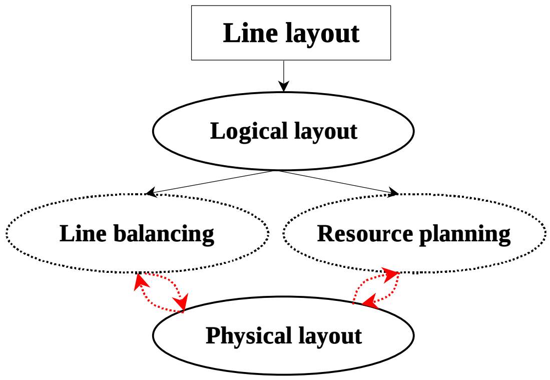

Figure 1.2. Line layout problem.

The classical common objectives of logical line layout are to equalise the station's process time to the cycle time and to minimise the number of stations, whereas other factors may also heavily affect the system performances. The objective is to minimise the total cost of the line by integrating design (station space, cost, etc.), operation issues (cycle time, precedence constraints, availability, etc.) and human desires (tasks complexity, etc.). Figure 1.3 shows the main features (blocks) of our concurrent assembly line design approach. The different blocks will be explained in detail along the chapters of this dissertation. The integrated approach to ALD will be detailed in Chapter 10. The ALD is a difficult task, since some information may be missing at the early stages of the design, and some available information may be subject to changes during the design or operation phase of the assembly line. None of these problems has a simple solution. An approach based on objectives and constraints was proposed to deal with such problems.

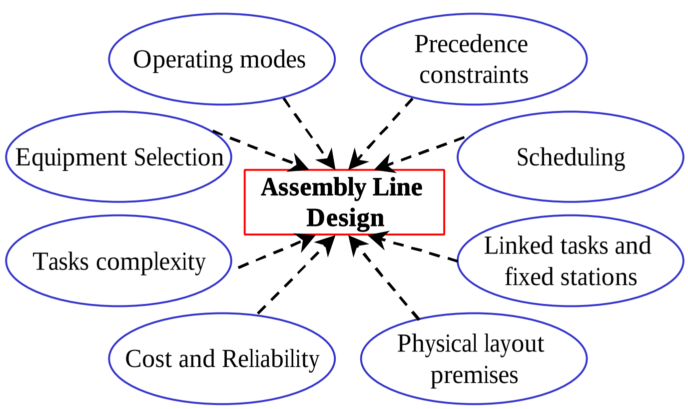

Figure 1.3. Concurrent design of assembly line.

A line design problem often has a complex structure due to multiple components, e.g. tooling, operators, material handling facilities, and so on. For a single product, 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 structured approach. For a given product and a given manufacturing environment, the design objective and constraints should be defined.

In this book, the author presents a computer system which imitates the human designer (it is a result of many collaborations with industrials) and is inspired by nature (Darwinian's evolution). It aims to design or redesign new assembly line starting from a set of specifications. Since the system is based on evolution, it will not be limited by the conventional *wisdom* of humans, and could create designs totally different from those produced by humans. Throughout this study we will talk about the *evolutionary design of the assembly line*. The system is aimed at being *generic*, i.e. it has to be capable of evolving a wide range of different line designs with minimal reconfiguration by a designer.

The proposed evolutionary design system has the following features :

1. It creates the assembly line logical layout from a combination of user-specified initial values. The system is given a complete freedom to evolve any solution that will fulfil the design specification. This allows the system to propose new and potentially unconventional designs. The system can also use *sub-designs* to create the whole design.
2. It allows an easy specification of the designer's desiderata. The main task is to find a compromise between the amount of data to be introduced in the system and the limitation or the validity of the specification.
3. It limits the interaction between the system and the designer to the specification stage. Designers have to communicate their desiderata to the system in the shape of a set of specifications, preferences or constraints (input data). Thus, a space of unconventional designs can be explored–evolutionary methods can help to propose solutions differing from conventional solutions.

2. Designing or optimising?

The *design* process is carried out using many different sources of knowledge and many different types of reasoning. Design is also an intelligent human information processing activity requiring many skills and knowledge. Design problems can be solved by individuals or by teams. Although design problems in different domains require different domain knowledge (such as mathematical modelling, evaluation, and analysis techniques) there are underlying similarities in the form of that knowledge and in the way it is used. It is obvious that different domains and different design problems will require a different knowledge and reasoning. This is why only a constructive analysis of a given design problem allows us to build useful *intelligent computer-aided design systems*. Systems can range from autonomous design tools, that will produce designs to the given requirements, or systems that support the designer in his design activity.

In general terms design is the process of specifying a description of an object (product, program, etc.) that satisfies a collection of *constraints*. These constraints may arise from a variety of sources. The constraints may be imposed by the problem itself, the designer, the end-user, or by natural laws. The term 'constraint' usually means something which is either satisfied or not (preferences are cited in general as *soft constraints*). In addition, there are special objectives to be met, such as minimising a cost. These act throughout the design process, and can be used to evaluate the quality of a given design. Not all of these requirements must be specified initially. It is a characteristic of many design problems that new constraints emerge as *decisions* are made (Gen, 1997).

Making a decision means choosing one or more alternatives from a list of options. The list of options would normally be more-or-less acceptable solutions for the problem. The performances of the chosen solution can be good or bad. The aim of the decision making is to maximise the positive performances and minimise the negative ones. The decision problem is related to a matter of considering the performance of all the options available simultaneously so that the *decision maker* (DM) can exercise the choice. The decision maker can be a human or a computer program. The different options may either be available and finite in number, as in a list (a catalogue), or they may be synthesised, as in engineering design. In general, the performance landscape is not single-peaked (*uni-modal*), but *multi-modal* one. This means that one needs robust *optimisation (search)* methods that can cope with a noisy, multi-peaked and discontinuous landscape. When there is no list of solutions to choose among but only a list of requirements to meet, it is appropriate to think in terms of *objectives*. An *attribute* (a characteristic of a given solution) with an associated direction is an objective. Thus, cost is an attribute but the aim of minimising cost is an objective. Real-world problems usually have a set of objectives associated to some *preferences* (the relative importance) set by the decision maker. Designs and their related problems can then be viewed (or classified) as multiple objective decision making problems (Sen, 1998).

The *combinatorial optimisation* deals with problems which are characterised by a finite number of feasible solutions. Although the optimal solution of such finite problems can be found by an enumeration, it is frequently impossible in practice, especially for practical problems of realistic size (the number of feasible solutions can be extremely high). The most challenging issue in combinatorial optimisation is to deal with the combinatorial explosion of problems. We can observe an important tendency to use *heuristics* rather than exact methods. Search methods are characterised by a search space (*domain*), on which they perform the search. A *metric* is needed to identify a successful search, i.e. indicate if the *goal* looking for was reached or not. This metric could be *binary* ('found', 'not found yet') or be an *information* on the proximity of the current solution in relation to the *best* solution (most of time unknown in advance). In many discrete-space problems, there is no *better* or *worse* solution, the solution is either wrong or right. The aim is rather to find a solution that satisfies the different constraints of the problem, the optimisation phase starts only once a solution satisfying all the constraints is found (Gen, 1997).

When designing an artifact, the question we ask is: are we looking for optimal solutions or only for satisfactory ones? In its pure definition, optimisation is '*to seek an ideal that is never realised*,' while the practical definition '*it is optimised if it gets reasonably close to optimum*'. Optimisation (loosely speaking) is nothing more than an evaluation function, hopefully, bringing its output closer to some ideal output (a scalar). Thus, optimisation implies the existence of an optimum. Simon (Simon, 1981) described the *satisfying search*, as a search that returns an element of the domain that *satisfies* all the constraints of the problem and the value returned by the evaluation function is *sufficient* for the current situation. Many authors in cognitive psychology argue that *people satisfy rather than optimise* (Mainzer, 1994). Most of the time, the design requirements are too inaccurate to provide clear evaluation criteria that can be used to determine the success or failure of the resulting design. The question is:

What does optimal solution mean in design problems?

There are two classification of design from the practical point of view :

Optimisation It tends to search for an optimum (regarding a *certain* goal) in the range of a given search space assigned to the problem being solved. For problems where a measure can be defined on the range, optimisation can be an appropriate method.

Search and decision The aim is to find a solution that satisfies a set of objectives under a set of constraints. In case there exist many solutions (having different *attributes*), a decision must be taken to select the best one.

Thus, assuming that the problem is of a reasonable difficulty, what we usually do with any computational method is to satisfy some needs (goals) rather than optimise. Indeed, the optimisation is an *objective* task, while search methods are *subjective*. Design methods which are generally subjective are more close to search and decision than to optimisation.

We think that design is an intellectual, cognitive activity rather than a procedural one. Indeed, the complexity of design is not due to the physical, material or procedural acts, but to the cognitive acts of understanding a problem and making well-founded decisions. There are some general steps that are good to follow when designing (Pahl, 1996). The designer has to begin by (1) formulating the problem to be solved, (2) then breaking down the problem into sub-problems, (3) grouping ideas that must be discussed, (4) evaluating and redesigning (if needed) the current design, and finally (5) implementing the proposed model.

3. Assembly line design

Our study does not proclaim newly discovered truths valid for all decades; it is done only for some particular people, and depends on its time and place and the socioeconomic level. It documents the *discoveries* and *realisations* and *expertise* gained from learning over the time (several years) it has taken to come to fruition. We had many discussions (with *academicians* and *industrials*) and shared many ideas with others a few years ago, now it is a pleasure to explore them again. With this in mind, this work is directed towards two main groups. The first is the *academic* (the *optimisation [search and decision]*, especially the *evolutionary computation*) community; the second, is the *designers* (especially *assembly line designers*) community. This book proposes a methodology for designing manual and hybrid assembly line, using *metaheuristics* (evolutionary approaches). Hence this study is based on ideas from these two communities, and I hope it contributes to both of them.

All members of the genetic algorithms (GAs) community are persuaded that it may be helpful to borrow ideas from nature and apply them in algorithms (methods) to achieve particular ends. Unfortunately, most research was concentrated on function optimisation, despite advice done by both John Holland (Holland, 1975), the father of GAs, and Emanuel Falkenauer (Falkenauer, 1998). Our emphasis will be more on a *solution construction*, a *function evaluation* and *selection* in the case of real-world optimisation, seeking to escape from the pitfalls of the classical methods. A kind of interactivity with evolutionary methods is introduced to deal with many, most of the time, conflicting *objectives*.

Do not stay in one place, move towards a goal! Assembly line design is a particular domain where such ideas can be applied. Different levels of complexity (many constraints) may be treated by the designers community, the computer is used to do the routine job of the design process. The designers prepare the data and do the last part of the job (visualisation, interpretation, decisions making, etc.). The 'virtual' limit between the two domains composing the whole job (computer and human) is not fixed and evolves in time. Whenever a success is achieved in a restricted domain, humans try to extend this domain, 'move the limits afterwards and/or backwards', so that one could talk here of design methodologies *co-evolving* with humans and their environments. People within optimisation can be broadly divided into two categories. First, those who are doing science. They deal more with computational models, algorithms behaviour with instances, convergence speed, search space architecture– they are more interested in theory (*cognition*) (Sedgewick, 1996) (Holland, 1975). The second category is those who are doing *engineering*–they are interested in creating useful or intelligent creatures, and treat ideas from nature, biology, evolution, etc. as means to a desired end (Falkenauer, 1998) (Gen, 1997). In this work we merely stand between the two camps (we are closer to the second community). Our stance is often modified and evolves by the inevitable constant interchange of useful and interesting ideas between engineers and scientists. Optimisation practitioners acting as engineers first need to be convinced that evolutionary approaches applied to design have advantages. It is still a big job to introduce such techniques into real-world industry.

It can be argued that during iterative and interactive design of complex systems approximate methods are the only possible ones. Evolutionary methods can do a good job to resolve such problems. Further, one needs to establish the different limits among all areas of systems design that have to be tackled by a given method.

This inevitably leads into some philosophical issues. We believe in the fact that a kind of relativity exists in design, and that there is a great interdependence between the objectively real-world and any observer (decision maker). The topic of our work is a miscellany of philosophy and engineering of a specific kind. Our approach is 'perhaps' still a minority one, yet fundamental to the directions pursued here, we will make it as clear as possible each time it seems appropriate and necessary.

Almost all the (*scientific*) historians agree that the improvements made in manufacturing were the engine that drove the *industrial revolution*. A remarkable increase in productivity was observed due to the improvements made in many types of technology—making a 'larger slice of the pie' for everyone. From Ransom E. Olds, who introduced the assembly line to car production, passing by Henry Ford's 20th century assembly line, through Eli Whitney's concept of interchangeable parts, manufacturing has been at the centre of industrial growth (Kirton, 1994).

Modern engineering is concerned with the design, improvement, and installation of integrated systems, equipment, and materials. It is based on specialised knowledge and skills in mathematical, and social sciences together with engineering methods of analysis and design, to specify, predict, and evaluate the design results. Many fields use the term *design* as part of their title; instances include, architectural design, product design, assembly line design, etc.

The term design implies a systematic planning processes prior to the execution of a plan in order to solve problems or development of things. Design is distinguished from other forms of planning by the level of precision, expertise and care used in the planning process. Design involves the consideration of many factors that may affect or be affected by the execution of a given plan. Many novice designers have the impression that doing design work is a 'cut-and-paste from old design' activities. This is not the case: *creativity* has a major role in design. Research is still going on in this field... The question is *where are we now?* and *what are we still lacking?*

4. Layout of the book

This book is divided into eleven chapters, including this one. The contents of the remaining chapters are summarised below. Broadly the book can be divided into four parts: the first describes the assembly line design problem, the second part introduces the chosen search method (GA). The third part is dedicated to the line layout problem. We finish by a fourth part which presents the integrated method.

The first part is for any public. The second part is dedicated to the evolutionary computation method used in this work and is more dedicated to the optimisation community. Parts three and four represent our main problems and is addressed to the assembly line designers community. Thus, the readers are invited to chose the chapters they will read depending on their interests.

Part 1: Assembly Line Design Problems: History

This part considers the principal aim of the given work which is the assembly line design. Chapter 2 introduces the design problems. Chapter 3 recalls the history and the evolution of assembly line and summarises the principal concepts of assembly.

Part 2: Evolutionary Combinatorial Optimisation

The second part deals more with the search method used in multiple objective design problems. The Fourth chapter gives an overview on GAs. The Fifth chapter considers the multiple objective design problem and introduces the improvements made to the traditional GA.

Part 3: Assembly Line Layout

Chapter 6 is dedicated to the manual assembly line balancing problem and explains the new balancing method 'the equal piles for assembly line', while chapter 7 is dedicated to the resource planning for hybrid assembly line and show how to tight the gap existing between the academic and real-world design methods.

Part 4: The Integrated Method

This last part is dedicated to the integrated method to design assembly line using evolutionary methods. The new concept of balance for operation (BFO) is described in chapter 8. It shows the interaction between the design and operation phases of assembly line. Most line layout approaches consider the physical layout problem after the logical layout. By separating the two problems, sub-optimal solutions are often obtained. In chapter 9 we introduce a new approach which allows us to use the premises of the physical layout as input data for the logical layout. The concurrent approach to assembly line design is presented in chapter 10, it shows the way to iteratively use and interactively the different modules.

A final summary chapter will attempt to assess the significance of what has been covered in this book. Conclusions and proposals for future works form the closing chapter of this book.

5. References

(De Lit, 1999) De Lit P., Rekiek B., Pellichero F., Delchambre A., Danloy J., Petit F., Leroy A., Marée J.-F., Spineux A. and Raucent B., 'A new philosophy of design of a product and its assembly line'. Proceedings of ISATP'99, Porto, Portugal, pp. 381-386, 1999.

(Delchambre, 1996) Delchambre A., 'CAD method for industrial assembly concurrent design of products, equipment and control systems', John Wiley & Sons Inc., Chichester, England, 1996.

(Falkenauer, 1998) Falkenauer E., 'Genetic algorithms and grouping problems', John Wiley & Sons Inc, Chichester, First Edition, 1998.

(Gen, 1997) Gen M. and Cheng R., 'Genetic algorithms & engineering design', John Wiley & Sons Inc, First Edition, Canada, 1997.

(Holland, 1975) Holland J.H., 'Adaptation in natural and artificial systems', University of Michigan Press, Ann Arbor, 1975.

(Kirton, 1994) Kirton J. and Brooks E., 'Cells in industry: managing teams for profit', McGraw-Hill, Berkshire, England, 1994.

(Mainzer, 1994) Mainzer K., 'Thinking in complexity: the complex of dynamics of matter, mind, and mankind', Springer-Verlag, Berlin Heidelberg, 1994.

(Pahl, 1996) Pahl G. and Beitz W., 'Engineering design: a systematic approach', Springer-Verlag, Berlin Heidelberg, 1996.

(Sedgewick, 1996) Sedgewick R. and Flajolet P., 'An introduction to the analysis of algorithms', Addison-Wesley, England, 1996.

(Sen, 1998) Sen P. and Yang J-B., 'Multiple criteria decision support in engineering design', Springer-Verlag, 1998.

(Simon, 1981) Simon H.A., 'The sciences of the artificial', The MIT Press, Cambridge, MA, 3rd ed., 1981.