RAIRO. Recherche opérationnelle

D. de Werra

Design and operation of flexible manufacturing systems : the kingdom of heuristic methods

RAIRO. Recherche opérationnelle, tome 21, n° 4 (1987), p. 365-382

<http://www.numdam.org/item?id=RO_1987__21_4_365_0>

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 21, n° 4, novembre 1987, p. 365 à 382)

DESIGN AND OPERATION OF FLEXIBLE MANUFACTURING SYSTEMS: THE KINGDOM OF HEURISTIC METHODS (*)

by D. DE WERRA (1)

Abstract. — *Although most of the decision problems related to the operation and to the design of flexible manufacturing systems (FMS) may be clearly formulated, the use of exact mathematical techniques is generally not possible for various reasons. One is then led to apply so-called heuristic methods. Since many new types of optimization problems arise in FMS, there is a need for developing new heuristics. A framework for designing such procedures is presented; basic principles for developing and testing heuristic methods are presented. References are made to decision problems occuring in FMS.*

Keywords : Flexible Manufacturing Systems; Heuristics, Scheduling; Optimization; Production; Models.

Résumé. — *La mise en place et la conduite de systèmes flexibles de production posent de nombreux problèmes qui ne peuvent en général pas être traités par des méthodes mathématiques exactes. On doit recourir à des méthodes heuristiques. Cet article passe en revue quelques principes qui peuvent être utiles pour élaborer, analyser et classifier ce type de méthodes.*

Mots clés : Ateliers flexibles; Heuristiques; Optimisation; Production; Modélisation; Ordonnancement.

1. INTRODUCTION

Among the many decision problems related to flexible manufacturing systems (FMS), we shall essentially consider the following ones: (*a*) design problems and (*b*) operational problems (*see* [18]). As soon as one tries to formulate and solve these problems, one is led to apply heuristic methods, just because in general no exact mathematical technique can bring up "the" solution in a reasonable amount of time.

(*) Received February 1987.
(1) École Polytechnique Fédérale de Lausanne, Département de Mathématiques, MA (Ecublens), CH-1015 Lausanne, Suisse.

But what is a heuristic method ? Although the word may not seem familiar, practitioners have been using such methods for a long time. Roughly speaking a heuristic method is a simple procedure mostly based on common sense which is designed for getting quickly good but not necessarily "optimal" solutions to difficult real-life decision problems. Such a definition raises many questions. For instance what is an "optimal" solution ? How good is a given solution ? How does one design a heuristic method ? Why does one have to use such procedures ? We shall try to discuss some of these issues in the next sections.

Let us first recall that the term *heuristic methods* or shortly *heuristics* has been initially introduced by specialists in Operations Research for describing procedures which could reduce the search for a solution in a problem-solving activity. Such techniques used to be considered as "quick and dirty" tricks and hence of no interest to the specialists of pure Operations Research. Since these times a huge amount of research has been carried out in the area of heuristics; such efforts were motivated by the numerous real-life situations where a problem had been identified and could not be solved by using well-known mathematical techniques.

Besides this the recent development of artificial intelligence (AI) has shed a new light on heuristic methods. In fact one purpose of artificial intelligence is precisely to guide the search for a good solution. This is exactly what a heuristic method does. The connection between AI and heuristics goes the other way too: some techniques of AI will be useful in designing and using heuristics. We shall discuss later some scheduling techniques based on AI (*see* [20, 22]).

Several authors have tried to formalize the concept of a heuristic and to describe the process of designing such a method (*see* for instance Müller-Merbach [11], Eglese [5 *a*] and Silver *et al* [14]). Besides this a wide collection of papers describes applications and design of special purpose heuristics in various domains like scheduling, timetabling, inventory control, production planning, etc.

In this text we shall give a short introduction to these methods; we shall insist on some points which seem important for the practitioner, but we shall have to omit many aspects of these procedures. The interested reader is referred to the short list of references included in this paper: Müller-Merbach [11] and Eglese [5 *a*] give a rather complete list of papers on heuristics.

2. SOME PROBLEMS RELATED TO FMS

A flexible manufacturing system (FMS) is a fully automated, computer controlled production system. It consists of a set of numerically controlled

(NC) machines linked by a material handling system (MHS) and the whole is under central computer control. In spite of its relative youth, this research area has already produced a huge amount of contributions (*see* for instance the proceedings [8, 16, 17] which present various optimization problems occurring in FMS and which contain many references).

We shall just mention some of these problems in order to point out where and how heuristic methods have to be used.

From a very broad point of view we can distinguish, as mentioned above, two main classes of problems: (*a*) design problems (*b*) operational problems. In the first class we consider the situation where an FMS has to be installed: NC machines have to be chosen in view of a future production; also an MHS has to be bought. Furthermore a central computer together with a network of mini- or microcomputers must be installed.

Mathematical programming models have been proposed for solving these problems (*see* [8] for instance where several such models are presented).

In this same class are included as well the layout design problems: knowing the operations needed to produce the various part types, knowing which machines can perform these operations, one has to decide the locations of the machines and especially the connections to be built between the machines for the MHS.

A graph-theoretical model has been described by Afentakis [2]: starting from a transition graph representing all transitions of parts between machines, one has to find an "optimum" layout graph which summarizes the MHS connections between machines. Several heuristic methods are proposed for solving this design problem which are all based on basic graph-theoretical properties.

For evaluating the efficiency of a given collection of NC machines with an MHS, many models based on queuing networks have been proposed; an analytic review is given by Seidmann and Shalev-Oren (paper 31 in [16]). We shall not discuss this here; we just want to mention that these are stochastic models. One should remind that most FMS problems are quite deterministic (especially for medium term planning and scheduling). It is only because the analysis is easier that the problem is considered as a stochastic one. Most of the scheduling problems will generally be modelled without introducing probabilities.

Let us now turn to the second class of problems, the so-called operational problems. Due to the inherent flexibility of the production systems we are interested in, the situation is very different from the classical production scheduling problems.

We may, as several authors do, consider 3 levels of operational problems:

(A) Long range production scheduling.

(B) System set-up.

(C) Dynamic Operations Control.

In level *A* production of the various part types is determined for the long range. Mathematical programming models have been designed for this purpose.

Level *B* deals with the issues which have to be settled (by using the flexibility of the system) before production can start. In an FMS production occurs in batches: each batch consists of a mix of several products which are produced simultaneously during a certain time shift (several hours to one day). K. Stecke and I. Kim (paper 24 in [16]) consider the following phases:

(1) part type selection problem: choose the parts to be produced in the next shift;

(2) machine grouping problem: partition the machines of similar types into identically tooled machine groups (since the tool magazines of the NC machines have a finite capacity, we may have to partition the machines of a given type into several groups of machines in order to be able to perform in the coming shift all different operations which require this type of machine);

(3) production ratio problem: determine the relative part type mix ratios at which the selected part types should be produced over time. Here one should try to use the various machines as efficiently as possible in the various time shifts;

(4) resource allocation problem: allocate the minimum number of pallets and fixtures of different types required to maintain the production ratios;

(5) loading problem: allocate the operations (and hence the required tools) to the various machine groups.

Similar decompositions are given in Kalkunte *et al* [7]; *see* also [9]. Now these problems are not independent, but they are so intricate that one cannot hope to build a general model which handles all of them together. This is why most authors suggest to solve these problems consecutively, taking the output of a problem as an input for the next one. This will lead to an approximate method independently of how exactly each one of the five problems is solved. In some sense, this is a heuristic procedure. In practice what will happen is that the whole collection of problems will be re-solved several times with adjustments until a satisfactory solution is found. Such an iterative method could be called a hierarchy-cyclical procedure.

In [13] Rajagopalan combines some of the problems (1) to (5) to improve the solution one can hope to get. Parts grouping and tool loading give rise to integer programming models. Their size and complexity compells him to use heuristic methods; various such procedures are compared in [13]. Bastos describes in [2 *a*] a heuristic method based on linear programming for dealing with the "batching" problems.

Problem (4) is sometimes solved by using closed queuing networks: this allows to compute the throughput of the system in an equilibrium state as a function of the number of pallets available. This is also a heuristic procedure.

Problem (4) is sometimes solved by using closed queuing networks: this allows to compute the throughput of the system in an equilibrium state as a function of the number of pallets available. This is also a heuristic procedure.

For problem (1) one should not forget that there are many constraints which are to be taken into account: one should try to put in the same batch parts which require as much as possible the same tools. The use of group technology is now widespread and data bases are available in many instances: parts are grouped according to the types of operations they require.

Among the various models proposed for partitioning a collection of part types into a few "homogeneous" batches, graph-theoretical models can be designed. The problem reduces essentially to graph coloring; for this some techniques based on simulated annealing have been applied; good solutions can be obtained for large size problems (*see* Chams *et al.* [4]). The technique is once more a heuristic one; it has recently been improved by using some simple ideas of artificial intelligence. Results will be reported elsewhere.

For some of the problems (1) to (5) integer programming models (linear and nonlinear) have been suggested. When their size is not too large, the exact "optimal" solution can be computed with an implicit enumeration technique. When the size is too large a heuristic method will give a good (but possibly not optimal) solution. Notice that even in the first case, heuristics are needed for reducing as much as possible the enumeration.

Finally we have to examine the third level which consists of the dynamic Operations Control. Here one has to decide how the production of the next time shift should be run through the FMS. One also has to prepare the reactions of the system (which is fully automated) to machine breakdowns for instance. Essentially one has to determine dispatching rules for introducing the parts and priority rules for choosing in a queue the parts to be processed next on a given machine. These rules are sometimes dynamic in the sense that they depend upon the load of the machine or of the FMS. They are adapted from optimal rules determined for much simpler problems. In the

FMS case, they become heuristic rules. In that context simulation is a widely used tool for evaluating the efficiency of such rules.

So simulation also can be considered as a heuristic method for developing scheduling rules: several runs will help to determine by adjustments a scheduling technique which will lead to a reasonable solution.

For the scheduling problems arising in FMS, the use of artificial intelligence techniques seems to be a promising research area. Some authors discuss the use of an expert system for solving the scheduling problems (*see* [20, 22]). In such cases, the role of an expert system would be to recognize the state (of load) of an FMS and to deduce from this state the best scheduling rules to use. Due to the flexibility of the production system, these rules are generally more intricate than the ones which would be used in a classical jobshop.

In order to be successful an expert system must indeed include the knowledge of an expert. For FMS this knowledge may not even exist. This is why a different type of scheduling system may have to be developed. As a human being, the scheduling system may start with a collection of rules and it may make some experiments with various rules applied on-line to real scheduling problems in an FMS. Then the experiments and their results (small throughput of the system or long waiting lines or perfect balancing of the loads of the machines) are recorded and the system learns step by step what are the good scheduling rules for such and such cases. So the system can try to use the best rule it knows when a scheduling problem is encountered.

Research is presently carried out with this type of system which is again based on some artificial intelligence principles. Such an approach can also be considered as a heuristic scheduling method.

We shall not go into more details about the various problems occurring in FMS. We have just shown that in most cases where an optimization problem is present, the use of heuristic methods is the only way out. In the next sections we shall concentrate on the properties of heuristic methods; the above situations will provide examples in the discussion which follows.

3. WHEN TO USE HEURISTICS?

Without expanding too much on the process of problem identification, we shall consider the case of a manager who perceives that there is something to be done to improve the efficiency of an existing production system. This is usually the first step of an intricate process which will finally lead to solving a well-defined (mathematical) problem.

More formally, starting from a real system (here the production system), a *model* will be constructed. It should be a simplified representation of the system keeping its main features and dependencies and leaving out all unimportant characteristics. Particularly one should exhibit the decision variables (describing the decisions to be taken) and the state variables (describing the state of the system). Quite often such a model consists of mathematical relations, this is why one calls it a *mathematical model.* In some instances there are things which cannot be quantified in the system or even not be translated into mathematical terms.

Constructing a model, i. e. a simplified image of the real situation, is already making an approximation in the process which should lead us to "the solution". It should be pointed out that we are here interested in the approximation which will be made at a later stage: having constructed a model, we shall have to use an "approximate procedure" for getting a solution through the model. This is essentially where heuristics come up.

At this stage a distinction is made between *generative* (or *prescriptive*) models and *evaluative* (or *descriptive*) *models.* The first type may be considered as a black box which produces a collection of best decisions when the objectives which we try to reach have been given. In descriptive models, the decisions regarding the design and the operations in a FMS are given as input; the model simply helps the manager to increase her (or his) understanding of the system. It provides insight but not decisions.

The process of building a model is a complex phenomenon where problem owners and managers have to cooperate very closely. At this stage of the solution process we are in fact already involved in the design of a heuristic method which will lead us to a good solution.

In a generative model, one should also include the objectives which are aimed at; this means that various solutions can be compared according to one or several criteria. So in principle when such criteria have been expressed, it should be possible to use sophisticated mathematical techniques to get an optimal solution in a very short time.

Unfortunately this situation almost never occurs, and one is led to using a heuristic method. There are several reasons for that:

(*a*) the mathematical problem is so difficult that no analytic or iterative solution procedure is known. This is the case of most scheduling problems among the ones we have sketched.

(*b*) Although such a procedure exists, it is computationaly prohibitive to use or unrealistic in its data requirements.

(*c*) the exact methods may be too complicated to understand, so one prefers simpler and more transparent procedures. In many scheduling problems instead of implementing a complicated method providing an exact solution, the manager rather wishes to implement some simple decision rules for dispatching parts in an FMS. In building an expert system, only a few simple rules will be kept as candidates. Also in inventory control, a solution of a simple form (easy to apply for the users) is much preferred to a mathematically optimal solution.

(*d*) Resource limitations (project time, budget, manpower availability, etc.) may force the use of heuristics.

(*e*) A frequent case where heuristics have to be used is dynamic operations control (level C of operational problems): a collection of operations with yet unknown arrival times have to be scheduled on some machines. In this situation decisions may have to be made before the complete data are available. No exact method will provide a solution, so one has to devise some rules which will provide a reasonable schedule.

It is worth observing that heuristic procedures appear also within some exact methods: for instance obtaining a good initial solution by a heuristic method may be the key to success in an implicit enumeration method.

Finally if an exact method can be used, then one should obviously apply it; in large-scale applications a difference of a few percents from optimality may mean very large amounts of money. However there are many situations where heuristics are the only available methods, so that it is worth studying such procedures and seeing how they can be designed.

4. WHAT IS A GOOD HEURISTIC?

There are several features which characterize good heuristics; here are some of the properties which they should have:

(*a*) *Simplicity:* this means that the method itself should be simple; this will make it easy to understand to the user and so the chances of implementing it will be increased. Furthermore, as mentioned earlier, it should also provide solutions of a simple form (for instance if simple scheduling rules are used in an automated manufacturing system, it will be easier to continue running the system by hand for a while in case of breakdown).

(*b*) *Speed:* the method should require a reasonable computing time. First of all the computing time should be easy to estimate from the size of the data collection and it should by no means grow exponentially with the size of the problem. This requirement is especially important if the problem has to be solved on-line.

(*c*) *Limited space requirement:* it is important that the core storage requirements of the method be not excessive. In fact when designing a heuristic method for a specific problem, one also has to take into account constraints related to the computing resources: the method will strongly depend upon the available computing facilities. In an FMS there is usually a hierarchy of computers; heuristics running on local processors will have to be more modest in space requirement than the ones used on the central computer.

(*d*) *Good performance:* the chance of getting a poor solution (i. e. far from an optimal solution) should be very low. Also the solution given by the method should be on the average close to an optimal solution. We shall see later how to evaluate the performance of a heuristic method.

(*e*) *Generation of solutions:* In general the criteria used for evaluating solutions do not reflect all objectives of the problem; an optimal (or a good) solution may not be *the* solution for the manager. It is therefore highly desirable that a heuristic method generates several good solutions (possibly in one shot). This will allow the user to choose the one he prefers with respect to criteria which may be only implicit in the model. Such a feature is almost a necessity when solving the design problem of a FMS with a mathematical model.

(*f*) *Interactivity:* For real-life problems one generally develops interactive heuristics; the idea behind this is to attain a high efficiency by combining the experience of the problem owner with the mathematical techniques for constructing solutions. When a solution is to be produced, decisions at some crucial steps are left to the user who can then influence the construction process if she (or he) so wishes. Such a situation is common in the heuristics used for constructing university timetables. An experienced scheduler can obtain excellent results in a short time with a good interactive heuristic method.

In an FMS, one wishes to reach the highest possible level of automation. This interactivity may however be a desirable feature in some cases: instead of a human controller, we may have an expert system which will at some crucial steps take a decision for the choice of a scheduling rule.

(*g*) *Learning systems:* Nowadays a good heuristic procedure should not only provide good solutions for all problems it has to solve, but it should also go through a learning process. Remembering the results obtained on previous data, it could memorize the rules which were the most successful for getting good solutions. This is already applied in operations control.

As previously discussed, heuristics form the basis of most tools in artificial intelligence. An important research field consists in developing systems which will generate heuristics for designing first and then for running an FMS.

5. HOW TO EVALUATE THE QUALITY OF HEURISTICS?

Basically the value of a heuristic lies in its ability to provide a good solution in a (mathematical) model. One should remember here that the aim is not to get a (mathematical) solution in a model but rather to have a way of reaching an acceptable level of performance for the system (in our case an FMS) where a problem of design or of operation was identified.

Assuming for the moment that the model which was built contains the essential components of the system under study, we shall briefly describe some ways of evaluating the quality of a heuristic method. Our goal is to obtain a measurement of how far solutions produced by heuristics are from optimal solutions. Naturally these optimal solutions are unknown in the situations where heuristics have to be applied.

(a) Comparison with optimum solutions

A first idea consists in resorting to simulation for estimating the best solution; this approach is computationally tractable in the case of scheduling in an FMS; several runs of a scheduling technique will give an estimate of an optimal solution with the help of statistical techniques.

In general one will have to concentrate on small scale problems for making this evaluation realistic.

(b) Relaxations

When an optimum solution to the model cannot be found, then we may transform the model into a "relaxed" model: some of the requirements between the variables of the problem may be temporarily dropped. For instance if a scheduling problem comprises various types of constraints (time disjoint processing of some jobs), we may keep only the precedence constraints; this gives a much simpler scheduling problem. Its solution will not satisfy the requirements of the initial model, but for instance the total completion time of this relaxed problem will be a lower bound for the total completion time in the initial model. We know that the minimum total completion time of the problem will not be smaller than the value obtained (easily) for the relaxed problem. Such relaxations are generally used in the design phase as well as in the short term planning phase of operations problems.

In general we manage to obtain a relaxed model which is easy to handle and we construct an optimum solution. Then we check how close the heuristic solution is from this bound. The value of the optimum solution lies between the relaxed solution and the heuristic solution. So if the value of the heuristic solution is very close to the bound, it must also be very close to the optimum solution. A classical example of relaxation occurs when a model consists of an integer linear programming problem; a relaxation is obtained by removing the integrality constraints and we now have a usual linear programming model.

Another way of getting bounds in the case where model consists in a mathematical programming problem is to use the Lagrangean relaxation. In various instances this relaxation problem is easier to optimize (at least heuristically) and it provides a good bound of the optimum solution of the model.

(*c*) *Statistical methods*

Many heuristic procedures for finding good solutions in a model are such that the computation starts from an initial feasible solution which is improved step by step until some stopping rule is met (*see* Müller-Merbach [11]). It is therefore a simple matter to generate a sample of starting solutions, to construct several solutions by running the heuristics several times and to keep the best solution as the final result. The theory of extreme value statistics can be used to compute estimates of the best solution (*see* Zanakis et *al.* [25], Silver *et al.* [14] for references on this topic).

(*d*) *Other comparisons*

Besides the previous criteria, one should also mention some other ways of measuring the efficiency of heuristics. First one can simply compare the solution obtained by the heuristics with the performance of the solution actually used by the manager. If significant improvements have been obtained, a proof of nearoptimality is perhaps not the most urgent thing to develop. It is also frequent to compare the results obtained by a heuristics with the solutions provided by existing heuristics.

(*e*) *Worst case analysis*

For various heuristic procedures used in combinatorial problems, one has been able to compute the worst possible error that can happen (*see* Fisher [6]).

Such information is certainly useful; however a decision maker or a manager is more interested in average performance or in the probability that the error (distance from optimal solution) is larger than a certain value. It is conceivable that a specific heuristic may have a very poor worst case behaviour (this occurs for some very rare cases which may anyway be very far from the normal data of the real-life cases) and have a very good behaviour for almost all cases.

6. HOW TO DESIGN A HEURISTICS?

In section 3 we have alluded to the fact that even identifying a problem may be a considerable task if the system we are considering is complex. Many authors have studied the process of model building in very general situations (*see* Ackoff *et al.* [1]).

We shall limit ourselves to a few considerations which, although general, are oriented towards scheduling problems in FMS. First designing a heuristics in an art, i. e. there is no fixed set of rules which will automatically lead to a successful procedure. Second this art needs scientific support (Müller-Merbach [11]) and this is where some general rules should be discussed.

Having recognized that there is some problem in a manufacturing system, the first thing consists in building a comprehensive model. It will have to represent the essential features of the problem; quite often this is simply a mathematical model, i. e. a representation in terms of mathematical relations, but in general we may have more general models (mathematics only form a partial model). The reason for using more general models is that quite often the objectives are not clearly formulated, the relations between various components of the system are not explicit to the manager. Also one should include in the representation the experience of the manager and this is generally hard to translate into equations. Furthermore the constraint stating that the method has to be simple may not be easy to translate into mathematical terms.

Besides the many scheduling problems which are present in an FMS, school timetabling is another domain where heuristic procedures have been successfully applied for a long time (*see* for instance [23] for an introduction to some models in timetabling). Here the objectives are not formulated in precise terms and there are various types of requirements which should be taken into account with more or less flexibility.

In timetabling, it is often extremely useful to define a partial mathematical model which may express in formal terms the strong requirements (those which have to be satisfied) and the partial objectives which are well defined. Such partial models may for instance reduce to some discrete optimization problems in networks.

Working only on this partial model would certainly give an unacceptable solution for the practitioner. So one has to include the remaining requirements and objectives in the general model and it is on this general model that the heuristic procedure will be applied.

In some other cases (*see* [15]) the mathematical part of the general model may be almost absent. Notice that a danger exists that the specialist in Operations Research will try to use his or her knowledge of mathematics before he or she has completely grasped the essence of the identified problem (*see* Stainton *et al.* [15]).

For a specific problem type a heuristic will be efficient if it combines the efforts of the manager (who knows the system and its reactions) and the knowledge of the problem solver (who has hopefully some experience with the mathematical techniques of optimization).

One should also remember that unlike most well-known mathematical techniques for optimization, heuristics are very sensitive to the special structure and to the numerical data of a problem. As an example scheduling problems can be rather naturally formulated as integer linear programming problems (by dropping some requirements in some cases), but a general heuristic procedure for integer linear programming will be much less efficient than a heuristic taking into account the special structure of the integer programming formulation of the scheduling problem. It is therefore not a good approach to try to formulate a problem with all constraints in terms of say a huge integer programming problem and to apply to it a general heuristic for integer programming. Advantage must be taken of its special structure.

On this line, it should also be observed that a given heuristic may have quite different performances on two similar problems who differ only in the volume or density of constraints but not in their form.

More specifically in school timetabling it has often been shown that a given heuristic procedure will perform quite well for a given school, while it will provide an unsatisfactory timetable for a school of different size having the same type of constraints. Similar observations have been made for other types of scheduling problems. This means that heuristics have to be adjusted not only to a problem type but also to a specific set of data.

It should finally be observed that in most of the FMS problems we have discussed, there is not a single objective function to optimize, like a cost. We are in fact facing a collection of conflicting objectives and so we cannot define an "optimal" solution; there may not exist a solution which is simultaneously optimal for all criteria : we have a multicriteria decision problem. In spite of this, heuristic techniques may still be the techniques to use: they can provide solutions which are "good" for all criteria.

The above remarks indeed do not tell us how to model a complex system in order to be able to improve its performance. We could not expect this to happen, since devising heuristics is considered as an art. However for moving towards a partial answer to this question we should attempt to review various types of heuristics which have been described in the Operations Research literature.

7. SOME TYPES OF HEURISTICS

Our purpose is now to give a few guidelines attempting to classify the various types of heuristics. The classes which are described below may not be disjoint : some specific methods may belong to several of them. The interested reader will find in [11] a collection of references where classifications of heuristics for combinatorial problems are proposed. In [3, 5] some heuristics are designed for solving operational problems in a FMS.

(a) Direct methods

These procedures construct solutions step by step (without any backtraking) by using some greedy algorithm. In scheduling for instance, one may construct a list of jobs to be scheduled in order of decreasing priorities; the method would consist in scheduling the jobs one after the other while taking some additional constraints into account.

(b) Improvement methods

Here one starts from a solution satisfying as many of the requirements as possible and one tries to improve it by examining neighbour solutions and jumping to a better solution if one is found in the neighbourhood (in scheduling one could try to permute some jobs). Such methods have been widely used in combinatorial problems (like scheduling). In [2] such an improvement method is given for the design problem. Simulated annealing [4] is also in this class.

(*c*) *Decomposition methods*

When a model is too big to be handled, then a natural idea consists in splitting it into smaller models of a more reasonable size. Every submodel is then handled separately and the subsolutions of all submodels are then put together with some adjustment if necessary. In a production problem, one may for instance separate a workshop into smaller and almost independent flexible cells for which schedules can be built (possibly with heuristics) easily. A global schedule is then constructed by combining these individual schedules. This is precisely what is done when an FMS is decomposed into FMC (flexible manufacturing cells) which are as independent as possible.

(*d*) *Slicing methods*

In some cases we get models for which another type of size reduction is natural. Instead of cutting the model into smaller submodels of the same type, we decompose the solution procedure into a sequence of steps (like in the technique called dynamic programming) where the output of one step is the input of the next one. As an example [19] a first step in a production model would consist in determining the part types to be produced in the next period. Then one would determine the ratios of these part types. The next problem would be to assign the operations to the various machines (taking into account the constraints due to the capacities of the tool magazines). Finally the routings of the various parts (we may use several routings for the same part type) should be calculated in order to use the machines as efficiently as possible (*see* paper 24 in [16] and also [2 *a*]). Remember that in the normal case, one has to go through these slices several times (hierarchy-cyclical procedure).

The way to define the various steps of such a procedure would normally be suggested by the problem itself or by the problem solver who will recognize what type of slicing will provide at each step a model which he or she can solve at low cost.

We shall not discuss classification in more details here; there are of course many other ways of exhibiting differences or common points of heuristics.

8. FINAL REMARKS

We have made a very brief review of some properties of heuristics. Such methods are used daily for handling practically all real-life problems where quantitative decisions have to be made.

As a conclusion one should keep in mind that developing a heuristic is a process which starts from the early stage where some problem has been identified in a system. It is not only a technique which deals with a mathematical model. To develop an efficient heuristic one needs a comprehensive information about the system under study, an extended experience with the present solution method and a lot of creativity (lateral thinking) and imagination for suggesting new ways of deriving solutions (*see* [12]).

Building a model is a crucial step; one should remember that a model is not necessarily a mathematical model; its construction should simply help to get a clear insight into the problem. This can be done only with the active participation of the manager and the problem solver; the manager should be lead to formulating what is wanted.

Notice that there may be more than two persons or groups of persons involved in this process. Then, as underlined in [12], the problem solver (specialist in Operations Research) must suggest ideas for getting solutions (this is where creativity is needed). These suggestions may be guided by his or her knowledge of other problems or models which may look similar in some aspects. But the starting problem should not be forgotten (For him who only knows a hammer the whole world looks like a nail).

One should also emphasize that the problem solver should get involved in the system deeply enough; he must try to get solutions himself by hand if necessary in the same way as the manager used to work. Müller-Merbach mentions in [12] the case of timetabling where before suggesting any heuristic, the specialist in Operations Research should try to construct a timetable himself or herself. Only at that price will he or she fully understand the problem. The same is of course true for scheduling in FMS: experiments with a simulation model (evaluative model) may be extremely useful.

At this point the question is raised to decide whether a heuristic method giving a solution with the help of a computer should be a faithful image of what was done manually by the manager. Or should one define quite different approaches? There is no general answer to this; but one important point is that a heuristic should be as simple as possible ("People would rather live with a problem they cannot solve than accept a solution they cannot understand", according to Woolsey *et al.* [24]) as mentioned in section 4.

Then starts a long iterative process where heuristics are proposed to the managers, discussed with them; according to their remarks the model may be changed and new heuristics derived. This will stop when managers and solvers will agree on a solution method which meets the requirements of the management.

In this process the model will be constantly refined; in fact getting an excellent model is more important than having a heuristic giving an almost optimum solution :

An approximate solution in a good model is always preferred to an optimal solution in a rough model.

REFERENCES

1. R. L. Ackoff and E. Vergara, *Creativity in Problem Solving and Planning : a review*, European J. of Operational Research, Vol. 7, 1981, pp. 1-13.
2. P. Afentakis, *A model for Layout Design in FMS*, in [8], pp. 127-139.
2 a. J. M. Bastos, *Batching and Routing : two Functions in the Operational Planning of FMS*, Report KLR 86-19 C, Brown Boveri S. A. (Baden, Switzerland).
3. J. C. Ammons, C. B. Lofgren and L. F. McGinnis, *A large Scale Machine Loading Problem in Flexible Assembly*, in [17], pp. 319-322.
4. M. Chams, A. Hertz and D. de Werra, *Some Experiments with Simulated Annealing for Coloring Graphs* (to appear in European Journal of Operational Research).
5. Y. L. Chang, R. S. Sullivan, U. Bagchi and R. Wilson, *Experimental Investigation of Real-time Scheduling in Flexible Manufacturing System*, in [17], pp. 355-377.
5. a. R. Eglese, *Heuristics in Operational Research*, in: Recent Developments in Operational Research, V. Belton and R. M. O'keefe eds., Pergamon Press, Oxford, 1986, pp. 49-67.
6. M. L. Fisher, *Worst-case Analysis of Heuristic Algorithms*, Management Science, Vol. 26, 1980, pp. 1-17.
7. M. V. Kalkunte, S. C. Sarin and W. E. Wilhelm, *Flexible Manufacturing Systems: A review of Modeling Approaches for Design*, Justification and Operation, in [8], pp. 3-25.
8. A. Kusiak ed., *Flexible Manufacturing Systems: Methods and Studies*, North-Holland, Amsterdam, 1986.
9. T. E. Morton and T. L. Smunt, *A Planning and Scheduling System for Flexible Manufacturing*, in [8], pp. 151-164.
10. H. Müller-Merbach, *The Use of Morphological Techniques for OR-approaches to problems*, Operational Research'75, K. B. Haley ed., North Holland, Amsterdam, 1976, pp. 127-139.
11. H. Müller-Merbach, *Heuristics and Their Design: a Survey*, European J. of Operational Research, Vol. 8, 1981, pp. 1-23.
12. H. Müller-Merbach, *A Five Facets Frame for the Design of Heuristics*, European J. of Operational Research, Vol. 17, 1984, pp. 313-316.
13. S. Rajagopalan, *Formulation and Heuristic Solutions for Parts Grouping and Tool Loading in Flexible Manufacturing Systems*, in [16], pp. 311-320.
14. E. A. Silver, R. V. Vidal and D. de Werra, *A Tutorial on Heuristic Methods*, European J. of Operational Research, Vol. 5, 1980, pp. 153-162.
15. R. S. Stainton and D. B. Papoulias, *Heuristics-the Relational Approach*, European J. of Operational Research, Vol. 17, 1984, pp. 16-20.

16. K. E. STECKE and R. SURI eds., *Flexible Manufacturing Systems: Operations Research Models and Applications*, Elsevier, New York, 1986.

17. K. E. STECKE and R. SURI, *Flexible Manufacturing Systems: Operations Research Models and Applications*, Annals of Operations Research, Vol. 3, 1985.

18. K. E. STECKE, *Design, Planning, Scheduling and Control Problems of Flexible Manufacturing Systems*, in [17], pp. 3-12.

19. K. E. STECKE and J. J. SOLBERG, *Loading and Control Policies for a Flexible Manufacturing System*, Internat. J. of Production Research, Vol. 19, 1981, pp. 481-490.

20. S. SUBRAMANYAM and R. G. ASKIN, *An Expert System Approach to Scheduling in Flexible Manufacturing System*, in [8], pp. 243-256.

21. R. SURI, *An Overview of Evaluative Models for Flexible Manufacturing Systems*, in [17], pp. 13-21.

22. A. THESEN and L. LEI, *An "Expert" System for Scheduling Robots in a Flexible Electroplating System with Dynamically Changing Workloads*, in [16], pp. 555-566.

23. D. DE WERRA, *An Introduction to Timetabling*, European J. of Operational Research, Vol. 19, 1985, pp. 151-162.

24. R. E. D. WOOLSEY and H. S. SWANSON, *Operations Research for Immediate Application — a Quick and Dirty Manual*, Harper and Row, New York, 1969.

25. S. H. ZANAKAKIS and J. R. EVANS, *Heuristic "Optimization": why, when, and how to use it*, Interfaces, Vol. 11, 1981, pp. 84-91.