

**COMBINATORIAL OPTIMIZATION MODELS
FOR PRODUCTION SCHEDULING
IN AUTOMATED MANUFACTURING SYSTEMS**

Yves CRAMA

Ecole d'Administration des Affaires
Université de Liège
Boulevard du Rectorat 7 (B31)
4000 Liège Belgium
Email: y.crama@ulg.ac.be

To appear in: *European Journal of Operational Research*

Abstract. Production planning and scheduling models arising in automated manufacturing environments exhibit several features not encountered in models developed for traditional production systems. For instance, models of automated facilities typically include tooling constraints which reflect the possibility for a machine to use different tools in order to perform successive operations, within limits imposed by the size of the tool magazine. Also, these models often account for the existence of flexible material handling systems whose activities must be synchronized with the machining operations in order to optimize system utilization. In this paper, we describe a few interesting combinatorial optimization problems proposed in this framework, we point to their relationships with models investigated in seemingly remote areas, and we identify a number of challenging open problems.

Keywords. Scheduling theory; Combinatorial optimization; Flexible manufacturing systems; Tool management; Robotic cells.

1 Introduction: What this paper is (and is not) about

In this paper, we discuss a few interesting combinatorial optimization models which we believe to play a central role in mathematical programming approaches to production planning and scheduling for highly automated (flexible) manufacturing systems.

This article is not a survey, in the sense that we have no claim to provide a systematic or exhaustive review of the many uses to which combinatorial optimization models have been put in order to formulate management issues arising in automated manufacturing settings. There are several reasons to this negative choice. First of all, the *abundance* of research contributions in this field and the *diversity* of the issues that they tackle have been so great over the last few years that any attempt at (near) comprehensiveness would result, at best, in a terse enumeration of references. Second, several excellent surveys organized by subtopics have recently appeared and continue to appear regularly in the literature. This compartmentalization into distinct, though interrelated, subtopics, is obviously imposed by the difficulties mentioned above. It makes it easier to apprehend the central ideas and the state-of-the-art of research concerning a particular theme. Thus, the reader anxious to gain a broad understanding of the rich interplay between management science and automated manufacturing is referred to all or any of the recent articles [1,4,16,17,40,41,63,87,94,98,108,110,111] or to the books [8,19,29,109].

Now that we have stated what this article is *not*, let us try to explain what it strives to be. Several authors have proposed general frameworks for modeling production planning and scheduling decisions in flexible manufacturing (see e.g. [66,102,110]). In spite of minor differences, there is some kind of general agreement among these authors regarding the nature of the decisions that have to be made in the planning process, and on how this process can be structured in a hierarchical fashion. What still appears to be missing at this point, however, is a clear identification of the *fundamental optimization models* that lie at the core of these decision problems. Of course, many optimization models have been proposed in the literature and/or have been successfully applied to the solution of real-world planning problems in automated environments. But, oftentimes, these models include numerous specific features which, for all their practical or conceptual relevance, may obscure some of the basic issues rather than illuminate them. As a result, little consensus exists as to what a suitable model should be for a given subproblem, and the formulations proposed by different authors tend to be difficult to compare.

From the viewpoint of the theoretician, this is a rather disturbing state of affairs. It is much like being confronted to the huge literature on production planning in conventional manufacturing systems without being able to refer to classical models like the lot-sizing model or the various multiprocessor scheduling models (flow shop, job shop, parallel machines,...). We are not going to argue here about the benefits gained by focusing on simplified (hence restrictive) models: better understanding and classification of various approaches, easier complexity analysis, development of efficient algorithms or approximation algorithms, investigation of intricate formulations through a reduction to simple ones, etc. We simply want to point out that this line of attack has proved fruitful in other settings, and we will postulate without further justification that it should be the case in the present framework as well.

Of course, we have no ambition to identify all fundamental models alluded to above. Rather, our aim in this paper will be twofold. First, we will describe a few *combinatorial optimization models* which, as far as formulating planning decisions for flexible production systems goes, appear to rank among the more basic ones. (Let us immediately add, however, that our selection of models has been strongly biased by our idiosyncrasies and personal research involvements.) We will try to convince the reader of the importance of these models by pointing to some of their occurrences in the research literature, either in their pure form or as substructures of more complex models. Interestingly, several of these occurrences happen to lie far from the manufacturing world which primarily retains our attention here. We take this as additional evidence of the importance of these models.

Second, we will mention a number of concepts and techniques that have been applied by various authors to the solution of the models under discussion. We hope thereby to illustrate the richness of the models and to underline their attractiveness from a purely theoretical viewpoint. We will also identify a number of problems currently left open or that seem to deserve further attention.

In Section 2, we briefly present some basic concepts pertaining to automated manufacturing systems. In Section 3, we discuss several scheduling models that incorporate tool management considerations, and in Section 4, we consider models for simultaneous scheduling of parts and material handling systems.

As a final note, let us stress once more (in case this would not be evident from the above discussion yet) that our discussion will be limited to the *theoretical* underpinnings of *combinatorial optimization models* arising in production planning for flexible manufacturing systems. Other points of view are obviously very useful as well but fall outside the scope of our essay. (See for instance [66] for an overview of the basic quantitative techniques used for the solution of problems arising in automated manufacturing).

2 Automated manufacturing systems

2.1 What is an automated manufacturing system?

Our focus in this essay is on modern, highly automated manufacturing systems, i.e. production systems presenting one or more of the following characteristics (see for instance [8,52,102,110]).

A number of machines are used to perform operations on parts. Parts are moved and delivered to the machines by some (automated) material handling system (conveyor belt, automated guided vehicles, robot, ...). Each operation requires a set of tools that can be stored in the tool magazines of the machines. During production, an automatic tool interchanging device is used to quickly interchange the tools between the magazine and the tool holder of each machine (i.e. the device that holds the active tool). This rapid interchange capability enables the machines to successively perform several operations with virtually no setup time between operations. Typically, however, the tool magazine of each machine is too small to hold all the tools necessary for the execution of a production plan. Additional tools are therefore kept at a central storage area and can be brought

to the machines as needed (either manually or by an automated handling device). Finally, a network of computers take care of the control of parts, machines, tools and material handling. (A detailed description of an actual automated facility is found in [103].)

The above definition is broad enough to encompass stand-alone numerically controlled machines (NC, DNC, CNC), robotic cells, robotic flow shops and flexible assembly lines, as well as completely integrated flexible manufacturing systems (FMS). (As a matter of fact, we will indifferently use the words ‘flexible’ or ‘automated’ to qualify such systems.) Systems of this nature are common in the metal-working industry. Much of the literature has therefore concentrated on this type of industry, where our terminology also directly draws its origins. However, very similar settings are also widespread in the electronics industry for the production of printed circuit boards (PCBs). As a consequence, production planning for the assembly of PCBs and for other FMSs often rely on nearly identical models (this similarity has been for instance explicitly noticed in [67]; see also [4,29]).

2.2 What is flexible in a flexible system?

It is important to understand what differentiates, in the spirit of this paper, a flexible production system from a traditional one. In other words, what justifies the use of specific production planning models for flexible production systems?

The answer is, of course, flexibility. But what exactly means ‘flexibility’?

Automated manufacturing systems display flexibility in multiple and intertwined ways, pertaining to the equipment, processes, products, production volumes, etc. (See e.g. [18,42,94]). Sethi and Sethi [94] have accordingly enumerated 50 different terms used in the literature for various types of flexibilities, and have themselves chosen to distinguish between 11 distinct concepts. Among the more important of these concepts, let us mention the following ones (we borrow from the classification in [94], which is itself inspired from [18]):

- *machine flexibility*, that is the ability of the machines to perform various types of operations without requiring a prohibitive effort in switching from one operation to another;
- *material handling flexibility*, that is the ability of the material handling system to move different parts efficiently for proper positioning and processing through the manufacturing facility;
- *operation flexibility* of a part, that is its ability to be produced in different ways;
- *process flexibility*, which relates to the set of part types that the system can produce without major setups;
- *routing flexibility*, that is the ability of a manufacturing system to produce a part by alternate routes through the system.

We will concentrate in this article on machine flexibility and material handling flexibility, which, together with operation flexibility, constitute what Sethi and Sethi [94] term ‘component’ or ‘basic’ flexibilities of a system. These basic flexibilities restrict and determine to a very large extent the remaining ones. Mathematical formulations geared at modeling machine, material handling or operation flexibilities provide therefore some of

the fundamental ingredients for developing more comprehensive models of automated facilities. This explains our focus on this part of the literature.

Let us observe, however, that mathematical programming models have also been extensively used in order to develop production plans that either maximize or otherwise exploit routing flexibility. Such concerns are for instance very obviously present in [1,2,5,12,64,97,100,101,102], as well as in much of the literature addressing the assembly of PCBs in automated shops, such as [4,24,76,67]. Even though we are not going to discuss any routing models in detail, they clearly deserve special attention from the operations research community.

Machine flexibility is directly related to the capacity of the tool magazine and to the ease with which tools can be interchanged between the magazine and the tool holder. Therefore, planning models that explicitly take into account the bounded capacity of the tool magazine or the limitations and opportunities offered by tool changes are, in our view, especially characteristic of flexible production systems as opposed to conventional ones. This class of models will be discussed in Section 3.

Models for managing material handling flexibility, on the other hand, are already well established in the literature on traditional manufacturing systems (see e.g. [8]). What appears to be a specific feature of flexible systems, though, is the high degree of integration attained between the material handling system and the machines. This feature is directly reflected in several combinatorial optimization models which simultaneously address the processing requirements of the parts on the machines and the transportation of these parts through the facility. We will describe some of these models in Section 4.

Observe that, at the scheduling level, machine and material handling problems can be viewed as *scheduling problems under resource constraints*, i.e. problems whose description involves various resources (tools, robots, ...) in an essential way, besides the usual characteristics of the jobs (processing times, release dates, deadlines, ...). This point of view is discussed in [16] in a spirit very close to that of our essay.

3 Machine flexibility and tool management

The influence of tool management on the overall performance of automated production facilities has been stressed by several authors. The recent articles [40,111] contain comprehensive surveys of the literature on this topic.

At the individual machine level (as opposed to the tool or system levels, see [40]), tool management subsumes *the problem of allocating tools to the machine and simultaneously sequencing the parts to be processed so as to optimize some measure of production performance*. This generic *one-machine scheduling problem*, or *loading problem* in the terminology of Stecke [102], can somehow be seen as the FMS analog of the fundamental one-machine scheduling problem of traditional manufacturing [14,69].

A more precise formulation of the problem can be stated as follows. A *part set* or *production order* containing N parts must be processed, one part at a time, on a single flexible machine. Each part requires a subset of tools which have to be placed in the tool magazine of the machine before the part can be processed. The total number of tools needed to process *all* the parts is denoted by M . We represent these data by an $M \times N$ *tool-part* matrix A , with

$$a_{ij} = 1 \quad \text{if part } j \text{ requires tool } i \\ = 0 \quad \text{otherwise,}$$

for $i = 1, \dots, M$ and $j = 1, \dots, N$. The tool magazine of the machine features C tool slots. When loaded on the machine, tool i occupies s_i slots in the magazine ($i = 1, \dots, M$). We assume that no part requires more than C tool slots for processing. We refer to the number C as the *capacity* of the magazine and to s_i as the *size* of tool i . (Typical magazine capacities lie between 30 and 120. Tool sizes are usually in the range $\{1, 2, 3\}$, with tools of size 1 being most common. See e.g. [103].) The total number of tools required, i.e. M , can be much larger than C , so that it is sometimes necessary to change tools while processing the order. A *tool switch* consists in removing one tool from the magazine and replacing it by another one. A batch of parts is called *feasible* if it can be processed without any tool switches.

Additional data for the problem may include the processing time of each part, the total time available on the machine, the tool changeover time, the operating cost of each tool, the number of tools available of each type, etc (see e.g. [12, 64, 80, 91, 92, 102, 113]). Due dates, however, are usually not taken in consideration in this short-term framework (see [51, 85, 91, 115] for some exceptions to this statement).

The objective function of the one-machine loading problem can reflect various tactical goals: maximizing the number of parts that can be produced without tool switches ([27, 48, 50, 51, 102, 104]), maximizing the use of the magazine ([102]), minimizing the number of tool switches ([10, 25, 32, 90, 106]), minimizing the number of times that production must be interrupted for tool switches ([28, 50, 107, 91, 92, 104]), or optimizing more general performance measures ([64, 80, 91, 113, 114, 115]). (In multimachine systems, balancing workloads often becomes a primary objective; see [12, 97, 102, 104], etc.)

In the next subsection, we briefly present the main integer programming formulations that have been proposed in order to model the magazine capacity constraint and the interdependence between tools and parts. Then, we successively tackle three of the most basic objective functions (maximizing the number of parts in a feasible batch, minimizing the number of tool switches instants, minimizing the total number of tool switches).

Notice that, by restricting our attention to a few fundamental one-machine scheduling models, we will only touch upon the surface of tool management issues in flexible systems. Many combinatorial models of interest also arise in connection with other tool management problems (tool provisioning, arrangement and retrieval of tools in the magazine, ...). For a discussion of these, we refer to the surveys [40, 111].

3.1 Modeling the magazine capacity constraint

3.1.1 A linear model

We concentrate here on the simple case where no tool switches are allowed during the planning period (or, equivalently, we restrict our attention to a production period between successive tool switches). Under this restriction, the loading problem becomes one of selecting an ‘appropriate’ feasible batch of parts. Integer programming formulations of this problem usually rely on (some or all of) the following variables:

$$\begin{aligned} x_i &= 1 && \text{if tool } i \text{ is loaded in the tool magazine} \\ &= 0 && \text{otherwise} \end{aligned}$$

and

$$\begin{aligned} y_j &= 1 && \text{if part } j \text{ is processed during the period under consideration} \\ &= 0 && \text{otherwise} \end{aligned}$$

for $i = 1, \dots, M$ and $j = 1, \dots, N$.

With these variables, the basic constraints of the one-machine loading problem can be formulated as:

$$a_{ij} y_j \leq x_i \quad \text{for } i = 1, \dots, M \text{ and } j = 1, \dots, N \quad (1)$$

$$\sum_{i=1}^M s_i x_i \leq C \quad (2)$$

$$x_i \in \{0,1\} \quad \text{for } i = 1, \dots, M \quad (3)$$

$$y_j \in \{0,1\} \quad \text{for } j = 1, \dots, N. \quad (4)$$

The first group of constraints (1) expresses that all the tools needed for processing the selected parts must be loaded in the magazine. Constraint (2) translates the limitations imposed by the capacity of the magazine.

The simple formulation (1)–(4), augmented of a general objective function of the form

$$\text{maximize } \sum_{j=1}^N \alpha_j y_j + \sum_{i=1}^M \beta_i x_i, \quad (5)$$

has been proposed in [48] to model the one-machine loading problem. It has been further examined in [27,39,50,51,104]. The formulation can also be easily extended to account for multiple machines or for the possibility to switch tools between production periods. Multi-period, multi-machine extensions can be found for instance in [38,52,64,80,91,106,113] and in numerous other papers. Thus, formulation (1)–(4) can be viewed as an essential component of many complex FMS tool loading models.

Interestingly, model (1)–(5) has also been considered in areas rather remote from the FMS world. It appears for instance in an order selection problem with high setup costs ([31]), in capital budgeting models ([77,112]), in repair kit selection problems ([77]) and in a provisioning model proposed in [68]. An application to the allocation of memory space in databases is described in [39]. A restricted version (corresponding to the case where two tools are needed for processing each part) also arises in a column generation approach to a graph partitioning model of compiler construction ([54]). As observed in [27], the full model (1)–(5) similarly arises if column generation is used to attack hypergraph partitioning problems relevant to VLSI layout.

3.1.2 Nonlinear models

Problem (1)–(5) can alternatively be modeled as a nonlinear integer programming problem in (at least) two different ways. To see this, let us first assume without loss of generality that, in (5), $\alpha_j \geq 0$ for $j = 1, \dots, N$ (if this is not the case, then the corresponding variable y_j can be set to 0 in the optimal solution of (1)–(5)). Then, by elimination of the y -variables, (1)–(5) can be equivalently reformulated as a *nonlinear knapsack problem*:

$$\text{maximize } \sum_{j=1}^N \alpha_j \prod_{i: a_{ij}=1} x_i + \sum_{i=1}^M \beta_i x_i \quad (6)$$

$$\text{subject to } \sum_{i=1}^M s_i x_i \leq C \quad (7)$$

$$x_i \in \{0,1\} \quad \text{for } i = 1, \dots, M. \quad (8)$$

The nonlinear knapsack problem with a *quadratic* objective function has been investigated in [20,34,46]. More generally, because the coefficients of all its nonlinear terms are nonnegative, the objective function (6) is supermodular; hence, (6)–(8) can be viewed as a special case of the supermodular knapsack problem studied in [35].

On the other hand, Stecké [102] suggested to model the magazine capacity constraint by the following nonlinear inequality in the y -variables:

$$\sum_{J \neq \emptyset} (-1)^{|J|+1} s(J) \prod_{j \in J} y_j \leq C \quad (9)$$

where the summation runs over all nonempty subsets $J \subseteq \{1, \dots, N\}$ and $s(J)$ denotes the sum of the sizes of the tools common to all the parts in J . Denote by $g(y)$ the left-hand side of (9). For any $y \in \{0,1\}^M$, $g(y)$ is nothing but the inclusion-exclusion formula computing the sum of the sizes of the tools needed in order to process the batch $\{j \in N \mid y_j = 1\}$. Hence, (9) is a valid expression of the capacity constraint. This expression can be viewed as resulting from the elimination of the x -variables in (1)–(2).

When the x -variables do not appear in the objective function (5) (which is the case in the framework of [102]), then an alternative formulation of (1)–(5) is given by

$$\text{maximize } \sum_{j=1}^N \alpha_j y_j \quad (10)$$

$$\text{subject to } \sum_{J \neq \emptyset} (-1)^{|J|+1} s(J) \prod_{j \in J} y_j \leq C \quad (11)$$

$$y_j \in \{0,1\} \quad \text{for } j = 1, \dots, N. \quad (12)$$

This model has a linear objective function and a single nonlinear constraint, and only involves the y -variables. (Contrast with the formulation (6)–(8), which has a nonlinear objective function, a linear constraint, and only involves the x -variables.) The merits of this model with respect to the linear model (1)–(5) will be discussed in the next subsection.

3.2 Solving the batch selection problem

Let us now survey some of the approaches that have been proposed to solve the *batch selection problem* (1)–(5) (or, equivalently, (6)–(8) or (10)–(12)).

The batch selection problem is strongly NP-hard. As a matter of fact, Gallo, Hammer and Simeone [34] observed that it is already NP-hard when each part requires two tools (i.e., when (6) is quadratic), $\alpha_j = 1$ for $j = 1, \dots, N$, $\beta_i = 0$ and $s_i = 1$ for $i = 1, \dots, M$. A further refinement of this result is established by Goldschmidt, Nehme and Yu [39]. These authors also identify some polynomially solvable cases of the problem and describe a dynamic programming algorithm for its solution. Several authors have developed and tested branch-and-bound codes based on the ideas surveyed hereunder.

In an early attempt to apply mathematical programming techniques to the solution of the batch selection problem, Stecké [102] investigated various procedures to transform (10)–(12) into a 0–1 linear programming problem (see also [12,79,81] for related studies). One of these procedures is the standard linearization method proposed in [37]. This method can be viewed as replacing (11) by a family of weaker linear inequalities, each of which is derived by substituting a linear function for every monomial of $g(y)$. Crama and Mazzola [26] proved that the linear description obtained in this way is weaker than the description (1)–(4), in the sense that its continuous relaxation defines a larger feasible region than the relaxation of (1)–(4). As a consequence, algorithmic approaches based on the inclusion-exclusion formulation (11)–(12) appear less promising than those based on the linear formulation (1)–(4).

A fundamental observation regarding the batch selection problem is that the subproblem defined by (1), (3)–(5) (and omitting the capacity constraint (2)) is polynomially solvable. Indeed, the constraint matrix of (1) is totally unimodular, so that the subproblem is reducible to a network maximum flow problem. It follows that the Lagrangian dual of (1)–(5) obtained by relaxation of the capacity constraint is polynomially solvable and has the same optimal value as the linear relaxation of (1)–(5). This observation has been exploited by several authors in order to efficiently solve the linear relaxation of (1)–(5), and hence to compute an upper-bound on the optimal value of the problem ([20,35,51,77]; see also [34,48] for related approaches).

Unfortunately, as observed e.g. in [28,31,51,54], the linear relaxation of (1)–(5) is often extremely weak. This has motivated several researchers to investigate families of inequalities which could be used to strengthen the formulation (1)–(5). Specifically, Dietrich, Lee and Lee [31] concentrate on cuts ‘implied by optimality considerations’, that is on inequalities satisfied by at least one optimal solution better than an incumbent solution (assuming that such an optimal solution exists). For the quadratic case (two tools per part), Johnson, Mehrotra and Nemhauser [54] present strong valid inequalities and facets for the convex hull of the feasible region defined by (1)–(4). Crama and Mazzola [27] systematically investigate the polyhedral structure of (1)–(4). They derive several families of valid and/or facet-defining inequalities, some of which generalize results of [54]. When $N = 1$ (one part) and $s_i = 1$ for $i = 1, \dots, M$, they obtain a complete description of the convex hull of the feasible region (subproblems with $N = 1$ naturally arise when one attempts to solve (1)–(5) via Lagrangian decomposition). Computational experiments will be necessary to establish the usefulness of the polyhedral approach to the batch selection problem.

Numerous heuristics have been proposed for the solution of the batch selection problem. Most of these heuristics are of the greedy type: a selection rule is iteratively used to add parts to the current batch, as long as magazine capacity allows. Heuristics proposed in [28,31,91,92,105,107,114] are of this nature. As far as we are aware, the relative perform-

ance of these heuristics has not been directly tested on the batch selection problem, but only when batch selection is iteratively solved to produce a solution of the tool-part grouping problem (see Subsection 3.3 below, and [28,65]). In this framework, the most effective selection rules appear to rest on (variants of) the following principle: among the parts not yet in the batch, select one that has the largest number of tools in common with the parts already in the batch (this is the Maximal Intersection rule described in [28]; see also the MIMU rule in [107] or [114]).

A local search heuristic is proposed in [31]. Heuristics based on Lagrangian relaxation schemes have been tested in [51,77]. Heuristics directly motivated by the quadratic knapsack formulation are considered in [46].

In conclusion, it seems that the basic integer programming model (1)–(5) is a rather ubiquitous one, and that the simplicity of its appearance is quite misleading. We believe that it deserves even more attention than it has received up to now, both from the theoretical and from the computational viewpoints. In particular, some intriguing questions remain open regarding the worst-case performance of heuristics for the batch selection problem. Crama and van de Klundert [23,60a] observe that most greedy-type heuristics for batch selection have unbounded worst-case ratio for this problem. For instance, on some instances, the MIMU heuristic could produce a batch involving (roughly) C parts while the optimal batch contains

$\binom{C}{C/2}$ parts. In [23,60a], it is also proved that the existence of a polynomial time algorithm with constant worst-case ratio for batch selection would imply the existence of a polynomial time approximation scheme for the same problem. The existence of such effective approximation algorithms, however, has not been completely ruled out. On the positive side, Kortsarz and Peleg [61] describe a polynomial approximation algorithm with ratio $O(M^{0.5})$ for the special case where each part requires exactly two tools, $\beta_i = 0$ and $s_i = 1$ for $i = 1, \dots, M$ (the exponent 0.5 can be slightly reduced at the cost of additional work). It would be interesting to extend such results to more general cases of the problem.

3.3 Grouping of parts and tools

In the previous subsection, we have focused on the problem of selecting a feasible batch, i.e. a subset of parts that could be produced without any tool switches. In general, however, the problem faced by the shop is that of processing the *whole* set of N parts as efficiently as possible. One way of turning this informal goal into a precise planning problem goes as follows: find a partition of the parts into a minimum number of feasible groups (batches). Equivalently, the objective function of this *tool-part grouping* problem is to minimize the number of tool switching instants. This objective is appropriate in situations where the automatic tool interchanging device can switch a set of tools simultaneously ([48,104,107]). It is also relevant if the machine has to be shut down during tool interchanges. Indeed, in such a case, the fixed cost incurred for interrupting production may dwarf the time spent in actual tools switches ([38,50]).

Let us also mention that the tool-part grouping problem is closely related in spirit to the *group technology cell formation* problem, which has recently generated an enormous amount of literature (see e.g. [8,30]). The objective functions of these two problems, however, are quite different. Moreover, many of the best known approaches to the cell

formation problem (e.g. [56,82]) do not explicitly model the capacity constraints which play an essential role for tool-part grouping.

The tool-part grouping problem has been studied by several authors, often ignoring each other. Rajagopalan [91,92] and Tang and Denardo [107] observed that partitioning parts into a minimum number of batches can be seen as an extension of the classical bin packing problem (see also Stecke and Talbot [105]). Hence, the part grouping problem is NP-hard [107]. Crama and Oerlemans [28] have further noticed that the problem remains NP-hard even when $C = 3$, and that deciding whether there exists a partition of the parts into two feasible groups is NP-complete. The latter result is especially interesting, as it establishes an unexpected connection between tool management problems and certain optimization problems related to VLSI and PLA (Programmable Logic Array) layout, such as those discussed in [84]. More precisely, in our terminology, the *PLA block folding* problem can be interpreted as that of determining the minimum value of the magazine capacity C such that the parts can be partitioned into two feasible batches. The PLA block folding problem is known to be NP-hard and is itself closely related to a wide variety of combinatorial problems (see [84] for details). It may be worthwhile to further exploit this intriguing connection between FMS and VLSI models (more on this topic in the next subsection).

Various heuristic approaches have been proposed for the tool-part grouping problem. Most of these heuristics iteratively solve (some version of) the batch selection problem in order to sequentially create feasible batches containing a ‘large’ number of parts (see [28,50,65,91,92,104,107,114]). Based on computational experiments with randomly generated instances, this greedy approach seems to perform very well when the batch selection subproblem is solved by heuristics based on some variant of the Minimum Intersection principle (see Subsection 3.2 and [28]).

From a theoretical viewpoint, however, most of the above heuristics have been shown in [23,60a] to have extremely poor worst-case performance ratios (this is essentially a consequence of the bad worst-case ratios achieved by batch selection heuristics; see previous section). For instance, the MIMU heuristic [107] can produce a

partition involving (roughly) $\binom{C}{C/2} / C$ more batches than an optimal partition. This is

especially striking in view of the fact that any *arbitrary* partition never uses more than $\binom{C}{C/2}$ times the optimal number of batches!

Goldschmidt, Hochbaum and Yu [38] describe polynomial approximation algorithms for some special cases of the tool-part grouping problem. But more general results are called for. Let us notice here that, in particular, any ρ -approximation polynomial algorithm for the batch selection problem can be turned into a polynomial algorithm with approximation ratio $O(\rho \log N) = O(\rho C)$ for the tool-part grouping problems, as follows from general properties of greedy algorithms (see e.g. [21]).

Hirabayashi, Suzuki and Tsuchiya [48] formulated the tool-part grouping problem as a large-scale set covering problem. In this formulation, each column corresponds to a feasible batch of parts and the constraints express that each part must be included in some batch. The authors mention in [48] the possibility to solve this set covering formulation using a column generation approach, and concentrate further on developing

a branch-and-bound procedure for the column generation subproblem. The latter subproblem turns out to be exactly our old friend, the batch selection problem (1)–(5).

Hwang [50] (see also [51]) and Tang and Denardo [107] rediscovered the previous set covering formulation, but, because of its size, did not use it for algorithmic purposes. These authors decided instead to rely on greedy heuristics (as mentioned above) or specialized branch-and-bound algorithms. Crama and Oerlemans [28] have implemented a column generation procedure that solves the linear relaxation of the set covering formulation of the problem. Computational experiments indicate that this approach delivers extremely strong lower bounds (almost always equal, in these experiments, to the optimal value of the problem). Moreover, the approach also lends itself to the computation of good heuristic solutions. Altogether, the column generation approach allows to solve to optimality much larger and sparser instances than those previously tackled.

3.4 Tool switching

In some situations, the total number of *tool switches* incurred while processing an order appears to be a more relevant performance criterion than the number of *switching instants* (i.e., the number of batches). This is for instance the case when the setup time of operations is proportional to the number of tool interchanges, or when the tool transportation system is congested. The distinction between number of tool switches and number of switching instants is clearly drawn in [106,107] and is also discussed at length in [5,97,104], etc.

We address in this section the following *tool switching problem*: determine a part input sequence and an associated sequence of tool loadings such that all the tools required by the j -th part are present in the j -th tool loading and the total number of tool switches is minimized. In this form, the tool switching problem has been investigated in [10,25,32,52,89,90,93,106,113,115]. All these papers are restricted to the special case where the tools occupy exactly one slot in the tool magazine. We shall assume that this condition holds throughout the section. (Notice that the formulation of the problem becomes ambiguous when the assumption is lifted.)

Crama, Kolen, Oerlemans and Spieksma [25] proved that the tool switching problem is NP-hard for any fixed $C \geq 2$. They also observed that deciding whether there exists a job sequence requiring exactly M tool setups is NP-complete. The latter result follows immediately from the kinship of the tool switching problem to the VLSI *gate matrix permutation problem* discussed in [84]. In our terminology, the gate matrix permutation problem asks for the minimum value of the tool magazine capacity such that no tool needs to be set up twice. Since this problem is NP-hard, the tool switching problem must clearly be NP-hard, too. (Notice the similarity of this argument with the one presented earlier for the tool-part grouping problem.)

The tool switching problem decomposes into two interdependent problems, namely:

- (1) *part sequencing*: determine an optimal part sequence, and
- (2) *tooling*: given a fixed part sequence, determine a tool loading sequence that minimizes the number of tool switches.

(This is a two-level scheduling model in the sense of [16]).

Tang and Denardo [106] established that the tooling subproblem can be solved in time $O(MN)$ by applying the so-called *Keep Tool Needed Soonest* (KTNS) policy. This policy prescribes that, whenever tools must be removed from the magazine in order to make room for the tools required by the next part, then the tools that are kept should be those that will be needed the soonest in the future. As observed in Privault and Finke [90] (see also [16]), the optimality of the KTNS principle was previously established by Belady [11] for a restricted version of the tooling problem, in his investigation of paging techniques for computer memory management. In a two-level memory system, a *page fault* occurs when a page must be moved from fast memory to slow memory to make room for a new page. The *paging problem* is that of deciding, for a given sequence of page requests, which pages to keep in a fast memory of C pages in order to minimize the number of page faults (see e.g. [11,83]). Thus, the paging problem is formally equivalent to a tooling problem in which each part would require exactly one tool.

Tang and Denardo's proof of correctness for the KTNS principle relies on ad hoc combinatorial arguments. Crama, Kolen, Oerlemans and Spieksma [25] present a more compact proof based on an appropriate integer programming formulation of the tooling subproblem. The constraint matrix of this formulation has the *consecutive ones property for columns*, i.e. it is an *interval* matrix in the sense of [33,86]. This directly implies that the tooling subproblem is reducible to a network maximum flow problem, even in its generalized version where each tool i has its own setup time b_i and the objective is to minimize the sum of all setup times (see [25]). When all setup times are equal, i.e. when the objective is only to minimize the total number of switches, then the integer program can be solved by a greedy algorithm which turns out to be equivalent to the KTNS algorithm (this follows from more general results on *greedy* and *totally balanced* matrices; see [49,86]).

The previous results have been further extended by Privault and Finke [90]. These authors give a direct network flow formulation of the tooling subproblem which allows them to model changeover costs of the form d_{ik} when loading tool i after unloading tool k . This approach leads to an $O(N^2)$ optimization algorithm for the generalized tooling subproblem. Interestingly, the formulation in [90] is inspired from related work on the computer paging problem mentioned above. Similar reformulations have also been exploited by several authors in the context of lot-sizing with sequence-dependent changeover costs; see Wolsey [116] for references.

(We observe here, in passing, that nothing seems to be known concerning the complexity of the tooling subproblem when tool sizes are not uniform.)

In spite of the simplicity of the tooling subproblem, the tool switching problem remains a hard nut to crack. Many heuristics have been proposed for its solution, but we are not aware of any successful attempts to solve reasonably large instances to *optimality*. As a consequence, the performance of different heuristics can only be compared relative to each other, as is done in [25,32,89,90,106], etc. Crama, Oerlemans and Spieksma [29] present partial results and suggest directions which may be worth exploring in order to compute tight *lower bounds* on the optimal value of the tool switching problem. More work is clearly needed on this question.

Heuristics for the tool switching problem come in two flavors: *construction* heuristics, which progressively construct a single, hopefully good part sequence, and *local search* heuristics, which iteratively modify an initial part sequence. In the first class, sev-

eral approaches are based on approximate formulations of the tool switching problem as a traveling salesman problem, where the ‘distance’ between two parts is an estimate of the number of tool switches required between these parts (see e.g. [25,89,90,106]). It may be interesting to notice that one of these traveling salesman formulations (namely, the block minimization model of [25]) is in fact an exact model for a database management problem closely resembling the tool switching problem (see [62]). Another type of construction heuristics fall into the category of ‘greedy’ heuristics: parts are successively added to a current subsequence on the basis of some (dynamically updated) priority criterion (see [25,32]). In [89,90], an efficient greedy-type heuristic is developed by drawing on ideas used in [83] for the solution of on-line paging problems.

Various local search strategies (2-exchanges, tabu search, ...) for the tool switching problem have been tested in [25,32,89,90,93,106,115].

4 Material handling flexibility

Classical scheduling models often ignore the constraints imposed by material handling systems (MHS) on the efficiency of the productive system as a whole, thereby implicitly assuming that the MHS does not constitute a bottleneck or a limited resource (as an example, transportation times are assumed to be zero between the machines of a traditional flow shop). On the other hand, in flexible manufacturing systems, material handling operations (transfers, loading, unloading) on parts, raw materials and tools are usually performed by automated devices such as robots or automated guided vehicles (AGVs). Such automated MHSs share a high degree of integration with the workstations, with the consequence that the performance of the system is directly affected by material handling activities. Therefore, the material handling system should be explicitly taken into account in FMS scheduling models. This point has been recently stressed by several researchers, e.g. in [15,16,57,60,95].

In order to discuss some of the recent results in this direction, we first propose a rather general *multiserver scheduling model* in which most of the results that we want to address fit rather well. We consider a part set made up of N parts. This part set must be produced either one time only, or infinitely many times, depending on the situation addressed. The manufacturing system consists of an *input station*, an *output station*, M machines and K identical servers. All parts are initially available at the input station and must end up at the output station after processing. Each machine can only process one part at a time, nonpreemptively; the processing time of part i on machine j is equal to p_{ij} ($i = 1, \dots, N; j = 1, \dots, M$). The servers model the material handling system in the following way. We assume that the presence of a server is necessary to load parts on machines before processing, to unload the parts after processing, and to transport the parts from machine to machine. Each server can only handle one part at a time. The servers can circulate unaided in the shop, possibly along some predetermined routes. Moving from machine j to machine l requires d_{jl} units of time. Loading or unloading operations on part i at machine j require time ε_{ij} . The multiserver scheduling problem is now to construct *simultaneous schedules for the parts and the servers* so as to optimize an arbitrary measure of production performance (makespan, cycle time, ...).

It does not take much imagination to realize that this basic multiserver model can be either enriched or restricted in various ways, e.g. by specifying the organization of the shop (parallel machines, flow shop, job shop, ...), by allowing preemption of operations, by defining release dates, due dates or weights for the parts, etc. More generally, almost any classical scheduling model can be turned as above into a multiserver model. Additionally, constraints can also be imposed on the set of moves available to the servers (unidirectional AGV path, hoists sharing a common track, ...).

Special cases of our general multiserver model have been investigated by several researchers, chiefly for flow shops and parallel machine shops (although some authors have considered more general situations; see for instance Lieberman and Turksen [74]). In the following subsections, we give a brief overview of these research contributions.

4.1 Flow shops

In *multiserver flow shop scheduling problems*, the machines are assumed to be laid out on a line or on a circle (say, in the order 1, 2, ..., M), each part must be successively processed on machines 1, 2, ..., M , and the servers can only travel from machine j to machine l by traversing all intermediate locations (the latter assumption can be modeled through an appropriate definition of travel times, e.g. by setting $d_{jl} = d_{jk} + d_{kl}$ if $j < k < l$).

Two main streams of research can be identified in the multiserver flow shop literature: one is concerned with scheduling problems in robotic cells (as in [3,22,43,44,55,57,58,59,60,75,95,99]), and the other one with hoist scheduling problems (as in [6,47,70,71,72,78,88,96]).

4.1.1 Robotic flow shops

Robotic flow shop scheduling problems are characterized by the presence of a unique server. It is commonly assumed that the machines have no buffers, the part set must be processed infinitely many times and the objective is to minimize the long-run cycle time of the schedule. This type of situation is typical of manufacturing cells where a number of machines served by a single robot are used for repetitive production of a minimal part set (see e.g. [7]).

Early work on this problem is due to Sethi, Sriskandarajah, Sorger, Blazewicz and Kubiak [95], as well as Kise, Shioyama and Ibaraki [60] for the case of makespan minimization. These authors study small shops, composed of 2 or 3 machines, and most of their investigations focus on *one-unit* cycles, i.e. sequences of robot moves in which each machine is loaded and unloaded exactly once and which can therefore be repeated indefinitely. In [95], the authors observe that there are only $M!$ one-unit cycles for any M . When $M = 2$, they show that, for any *fixed* one-unit cycle, the part sequence (i.e. the permutation of 1, 2, ..., N) which minimizes the long-run cycle time can be computed in polynomial time through a reduction to a special case of the traveling salesman problem which is amenable to the Gilmore-Gomory algorithm (see [36] for a discussion of this algorithm). (In [60], a similar result is derived for makespan minimization.)

However, as observed by Hall, Kamoun and Sriskandarajah in [43], the optimal cycle time is not necessarily attained by repeating infinitely many times the same one-unit cycle, even when $M = 2$: in some cases, it could be more advantageous to switch back and forth between the two possible one-unit cycles. The authors also show in [43] how to palliate this difficulty; they describe an algorithm that jointly determines the optimal robot move sequence and the part input sequence for a two-machine robotic flow shop, again by a reduction to the Gilmore-Gomory algorithm.

When $M \geq 3$, the robotic flow shop scheduling problem becomes much less tractable. As a matter of fact, Hall, Kamoun and Sriskandarajah [44] prove that, for $M = 3$, the problem is strongly NP-hard as soon as $N \geq 2$, whether robot moves are restricted to one-unit cycles or not. Moreover, Sriskandarajah, Hall, Kamoun and Wan [99] classify the complexity of the part sequencing problem obtained for any *fixed* one-unit cycle. They show that the part sequencing problem is polynomially solvable for $2M-2$ of the $M!$ available cycles and NP-hard for the remaining cycles. Finally, at the lowest level of complexity, Cohen, Dubois, Quadrat and Viot [20a], Ioachim and Soumis [51a] and van de Klundert [60a] provide polynomial algorithms for the following subproblem: given a feasible robot move sequence and a part input sequence, what is the smallest attainable long-run cycle time?

The previous results leave open the complexity of the robotic flow shop problem for *identical parts*, that is for $N = 1$ and arbitrary M . This special case is interesting in several respects. Sethi, Sriskandarajah, Sorger, Blazewicz and Kubiak [95] formulated the following conjecture (call it the *SSSBK conjecture*): for identical parts, one-unit cycles dominate multi-unit cycles, in the sense that long-run cycle time can be minimized by repeating infinitely many times the same one-unit cycle. (As mentioned before, this statement is definitely not true for multiple part types, i.e. when $N \geq 2$.) Sethi et al. [95] observed that the conjecture holds true when $M = 2$. For $M = 3$, its validity was recently established in van de Klundert [60a]. But in its full generality, this nice conjecture remains desperately open.

If the SSSBK conjecture is correct, then one-unit cycles are all we need to consider in the identical parts problem. On the other hand, independently of the truth of the conjecture, one-unit cycles remain attractive in industrial settings because of their ease of implementation. For identical parts and arbitrary M , Crama and van de Klundert [22] established that *pyramidal cycles* dominate all other one-unit cycles. To define the concept of pyramidal cycle, notice first that a one-unit cycle can be represented as a permutation (i_0, i_1, \dots, i_M) of the unloading activities $\{0, 1, \dots, M\}$ (where activity i corresponds to unloading a part from machine i , and the input station is identified with machine 0). Then, a pyramidal cycle is a permutation (i_0, i_1, \dots, i_M) such that, for some index s in $\{0, 1, \dots, M\}$, $i_0 < i_1 < \dots < i_s$ and $i_s > i_{s+1} > \dots > i_M$. The concept of pyramidal permutation has been thoroughly investigated in connection with the traveling salesman problem (and, coincidentally (?), with the Gilmore-Gomory algorithm repeatedly mentioned in previous paragraphs; see [36]). But the role played by pyramidal cycles in the identical parts problem is rather puzzling, as this problem does not seem to be directly reducible to the TSP. In spite of this, it is possible to exploit the dominance properties of pyramidal cycle and to solve the identical parts, one-unit, robotic flow shop problem in $O(M^3)$ time (see [22]).

Variants of the basic robotic flow shop problem have been investigated by several researchers. For instance, Kise [58] shows that the 2-machine problem is NP-hard when the machines feature infinite-capacity buffers, and King, Hodgson and Chafee [57] implement a branch-and-bound algorithm for this problem. Finke, Gueguen and Brauner [31a] prove that the SSSBK conjecture is valid (for any number of machines) when each machine has a single unit of buffer space. Kats and Levner [55a] and Levner, Kats and Levit [73a] consider no-wait problems for identical parts. Agnetis, Pacciarelli and Rossi [3] and Liu and Lin [75] consider swapping devices which act as temporary buffers at the robot arm. Heuristic, exact and Petri net approaches for other versions of the problem are proposed and tested in [55,59,60,73b,etc].

4.1.2 Hoist scheduling

Hoist scheduling problems, although closely related to robotic flow shop problems, exhibit several distinguishing features. First – and this is probably the main difference with the previous problem – the processing times p_{ij} are replaced here by two sets of parameters a_{ij} and b_{ij} with the following interpretation: the time spent by part i on machine j must be at least a_{ij} and at most b_{ij} , for $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, M$. The intervals $[a_{ij}, b_{ij}]$ are referred to as *time windows* by some authors (though these are not ‘time windows’ in the traditional acceptance of the vehicle routing literature).

Second, more than one server can be present in the flow shop. The servers are viewed as hoists sharing a common track. Because of this particular layout, the hoists cannot cross each other and collisions must be avoided.

Finally, all parts are usually assumed to be identical (i.e. $N = 1$). An infinite number of parts must be processed, and the objective is to minimize long-run cycle time. Much of the literature is restricted to the optimization of one-unit cycles.

Applications of the hoist scheduling problem are found for instance in the electronics industry, where printed circuit boards must undergo a sequence of chemical and electroplating operations in successive tanks. Hoists are used to lower PCBs into tanks, to lift them from tanks, and to transport them between tanks. The nature of the chemical processes imposes upper and lower bounds on the sojourn time of PCBs in each tank. Moreover, identical PCBs are often processed in large batches, thus justifying some of the assumptions of the model.

Contrary to the robotic flow shop problem, few results of a theoretical nature are available concerning hoist scheduling. The single-hoist, identical-parts problem is proved to be strongly NP-hard in van de Klundert [60a] (a previous proof by Lei and Wang [70] involves somewhat « artificial » instances, with non-zero travel time from a machine to itself). For the same problem, Lei [69a] proposes a polynomial algorithm which computes the optimal cycle time for any fixed sequence of hoist moves.

Much of the remaining research effort has concentrated on developing and testing practical algorithms for special cases of the problem. Phillips and Unger [88], in a seminal paper, gave an integer programming formulation of the single-hoist problem. Branch-and-bound procedures for the same problem have been proposed in [6,72,96,86a]. Multi-hoist problems have been considered in [47,71,78].

4.2 Parallel machines

In *multiserver scheduling problems with parallel machines*, the machines are assumed to be identical and each part needs to be processed on one machine only. The servers could be robots or AGVs. Settings of this type tend to arise frequently in flexible manufacturing systems. Indeed, due to their high versatility, individual machines are typically able to perform all operations required on each part ([52]). When tooled identically, such machines actually behave like parallel ones (see e.g. [101,102,103,105]).

Research on multiserver parallel machine problems is still pretty much in its infancy. Systems with a unique server have been investigated in [45,53]. Jeng, Lin and Wen [53] compare the performance of heuristic and branch-and-bound procedures for minimizing the sum of completion times. Hall, Potts and Sriskandarajah [45] consider a variant of our basic single-server parallel machine problem in which travel times and load/unload times are not explicitly modeled, but are rather aggregated into a global setup time s_j for each job j . They investigate the complexity of this scheduling problem for several objective functions (makespan, maximum lateness, sum of completion times, number of late jobs, total tardiness). They prove that the resulting problems are all NP-hard in their most general form, and they further identify some polynomially solvable special cases (most notably, when the processing times of all jobs are equal).

Blazewicz, Eiselt, Finke, Laporte and Weglarz [15] consider a version of the multiserver problem motivated by a real-world application. Here, the servers are AGVs and the facility is arranged following a unidirectional loop network layout. The AGVs are dispatched at regular intervals on the network and take of loading, unloading and transfers of parts. The objective is to minimize makespan. For a fixed part schedule (i.e., for a fixed assignment of parts to machines and fixed operations starting times), the authors show how to compute a feasible server schedule, if one exists. Their algorithm is based on a variant of the Earliest Due Date rule and runs in $O(M \log N)$ time. When the part schedule and the server schedule must be jointly determined, then the problem clearly becomes NP-hard. But the authors show that an optimal solution can be computed in pseudo-polynomial time by dynamic programming.

A generalization of the previous problem is studied by Blazewicz, Burkard, Finke and Woeginger [13]. Here, the transportation network consists of two loops sharing a common stretch and the emphasis is on developing a server schedule that is both collision-free and compatible with a given production schedule. As shown in [13], this question can be addressed by making use of elementary arithmetic notions. Namely, collision-free schedules can be enforced by releasing the vehicles at times that are incongruent modulo $\gcd(A, B)$, where A and B are the travel times along the two loops.

4 Conclusions

In this paper, we have presented a few combinatorial optimization models which explicitly reflect some of the characteristic features of flexible manufacturing systems, namely their tool interchange capability and their high-degree of integration with auto-

mated material handling devices. Besides their practical relevance in production planning, these models turn out to be highly attractive from the viewpoint of optimization theory as well. They are for instance closely related to models investigated in other application areas (we have mentioned VLSI layout and PLA folding, paging problems, database management, compiler construction, ...). Solution techniques and complexity analyses of these models rely on all tools of the trade (Lagrangian relaxation, dynamic programming, column generation, cutting planes, greedy algorithms,...) and make unexpected use of concepts like totally balanced matrices or pyramidal permutations.

Many questions are still left open concerning the models that we discussed. Some of these questions have been mentioned in previous sections. Many more will be found in the references cited in the text.

Acknowledgments

I am grateful to Dominique de Werra, Catherine Roucairol and Roman Slowinski, who gave me the opportunity to present this paper at the EURO XIV Conference. The views that it contains have evolved through numerous stimulating conversations and joint research projects with Joe Mazzola and the 'Maastricht team' consisting of Joris van de Klundert, Antoon Kolen, Alwin Oerlemans, Maarten Oosten and Frits Spieksma. I owe many thanks to all of them. This work has been partially supported by AFOSR (grant F49620-93-1-0041), ONR (grants N00014-92-J-1375 and N00014-92-4083) and NATO (grant CRG 931531).

References

1. Agnetis A., Arbib C., Lucertini M.: Combinatorial models for manufacturing: Optimizing flow management in flexible production systems. In: Ciriani T.A. and Leachman R.C., eds. *Optimization in Industry*. John Wiley & Sons, New York (1993), 81-99.
2. Agnetis A., Lucertini M., Nicoló F.: Flow management in flexible manufacturing cells with pipeline operations. *Management Science* **39** (1993) 294-306.
3. Agnetis A., Pacciarelli D., Rossi F.: Lot scheduling in a two-machine cell with swapping devices. *IIE Transactions* **28** (1996) 911-917.
4. Ahmadi R.H.: A hierarchical approach to design, planning, and control problems in electronic circuit card manufacturing. In: Sarin R.K., ed. *Perspectives in Operations Management*. Kluwer Academic Publishers, Dordrecht (1993), 409-429.
5. Amoako-Gyampah K.: A comparative study of FMS tool allocation and part type selection approaches for a varying part type mix. *The International Journal of Flexible Manufacturing Systems* **6** (1994) 179-207.
6. Armstrong R., Lei L., Gu S.: A bounding scheme for deriving the minimal cycle time of a single-transporter N -stage process with time-window constraints. *European Journal of Operational Research* **78** (1994) 130-140.

7. Asfahl C.R.: *Robots and Manufacturing Automation*. John Wiley & Sons, New York, 1985.
8. Askin R.G., Standridge C.R.: *Modeling and Analysis and Manufacturing Systems*. John Wiley & Sons, New York, 1993.
9. Balas E., Mazzola J.B.: Nonlinear 0-1 programming: I. Linearization techniques. *Mathematical Programming* **30** (1984) 1-21.
10. Bard J.F.: A heuristic for minimizing the number of tool switches on a flexible machine. *IIE Transactions* **20** (1988) 382-391.
11. Belady, L.A.: A study of replacement algorithms for virtual storage computers. *IBM Systems Journal* **5** (1966) 78-101.
12. Berrada M., Stecke K.E.: A branch and bound approach for machine load balancing in flexible manufacturing systems. *Management Science* **32** (1986) 1316-1335.
13. Blazewicz J., Burkard R., Finke G., Woeginger G.: Vehicle scheduling in two-cycle flexible manufacturing systems. *Math. and Computer Modelling* **20** (1994) 19-31.
14. Blazewicz J., Ecker K., Schmidt G., Weglarz J.: *Scheduling in Computer and Manufacturing Systems*. Springer-Verlag, Berlin Heidelberg, 1993.
15. Blazewicz J., Eiselt H.A., Finke G., Laporte G., Weglarz J.: Scheduling tasks and vehicles in a flexible manufacturing system. *The International Journal of Flexible Manufacturing Systems* **4** (1991) 5-16.
16. Blazewicz J., Finke G.: Scheduling with resource management in manufacturing systems. *European Journal of Operational Research* **76** (1994) 1-14.
17. Blazewicz J., Finke G., Haupt R., Schmidt G.: New trends in machine scheduling . *European Journal of Operational Research* **37** (1988) 303-317.
18. Browne J., Dubois D., Rathmill K., Sethi S.P., Stecke K.E.: Classification of flexible manufacturing systems. *The FMS Magazine* **2** (1984) 114-117.
19. Buzacott J.A., Shantikumar J.G.: *Stochastic Models of Manufacturing Systems*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.
20. Chaillou P., Hansen P., Mahieu Y.: Best network flow bounds for the quadratic knapsack problem. In: Simeone B., ed. *Combinatorial Optimization*. Springer-Verlag, Berlin (1989), 225-235.
- 20a. Cohen G., Dubois D., Quadrat J.P., Viot M.: A linear-system-theoretic view of discrete-event processes and its use for performance evaluation in manufacturing. *IEEE Transactions on Automatic Control* **AC-30** (1985) 210-220.
21. Crama Y., van de Klundert J.: Approximation algorithms for integer covering problems via greedy column generation. *RAIRO—Operations Research* **28** (1994) 283-302.
22. Crama Y., van de Klundert J.: Cyclic scheduling of identical parts in a robotic cell. Working paper (1994). To appear in *Operations Research*.
23. Crama Y., van de Klundert J.: The approximability of tool management problems. Working paper (1996).
24. Crama Y., Kolen A.W.J., Oerlemans A.G., Spieksma F.C.R.: Throughput rate optimization in the automated assembly of printed circuit boards. *Annals of Operations Research* **26** (1990) 455-480.
25. Crama Y., Kolen A.W.J., Oerlemans A.G., Spieksma F.C.R.: Minimizing the number of tool switches on a flexible machine. *The International Journal of Flexible Manufacturing Systems* **6** (1994) 33-54.

26. Crama Y., Mazzola J.B.: On tool loading, dense subhypergraphs and nonlinear knapsack problems. Unpublished manuscript, 1992.
27. Crama Y., Mazzola J.B.: Valid inequalities and facets for a hypergraph model of the nonlinear knapsack and FMS part-selection problems. *Annals of Operations Research* **58** (1995) 99-128.
28. Crama Y., Oerlemans A.G.: A column generation approach to job grouping for flexible manufacturing systems. *European Journal of Operational Research* **78** (1994) 58-80.
29. Crama Y., Oerlemans A.G., Spieksma F.C.R.: *Production Planning in Automated Manufacturing*. Springer-Verlag, Berlin, 1996 (2nd edition).
30. Crama Y., Oosten M.: Models for machine-part grouping in cellular manufacturing. *International Journal of Production Research* **34** (1996) 1693-1713.
31. Dietrich B.L., Lee J., Lee Y.S.: Order selection on a single machine with high set-up costs. *Annals of Operations Research* **43** (1993) 379-396.
- 31a. Finke G., Gueguen C., Brauner N.: Robotic cells with buffer space. In: O'Connor R. and Magee P., eds., *ECCO IX - Conference Proceedings*. Center for Teaching Computing, Dublin, Ireland, 1996.
32. Follonier J.-P.: Minimization of the number of tool switches on a flexible manufacturing machine. *JORBEL - Belgian J. of Operations Research, Statistics and Computer Science* **34** (1994) 55-72.
33. Fulkerson D.R., Gross D.A.: Incidence matrices and interval graphs. *Pacific Journal of Mathematics* **15** (1965) 835-855.
34. Gallo G., Hammer P.L., Simeone B.: Quadratic knapsack problems. *Mathematical Programming Study* **12** (1980) 132-149.
35. Gallo G., Simeone B.: On the supermodular knapsack problem. *Mathematical Programming* **45** (1988) 295-309.
36. Gilmore P.C., Lawler E.L., Shmoys D.B.: Well solved special cases. In: Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G., Shmoys D.B., eds. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. John Wiley & Sons Ltd., Chichester New York Brisbane Toronto Singapore (1985), 87-143.
37. Glover F., Woolsey E.: Converting the 0-1 polynomial programming problem to a 0-1 linear program. *Operations Research* **22** (1974) 180-182.
38. Goldschmidt O., Hochbaum D.S., Yu G.: Component assembly in the semiconductor industry: A study of covering in graphs and hypergraphs. Technical Report ORP92-5, The University of Texas at Austin, 1992.
39. Goldschmidt O., Nehme D., Yu G.: On the set-union knapsack problem. *Naval Research Logistics* **41** (1994) 833-842.
40. Gray A.E., Seidmann A., Stecke K.E.: A synthesis of decision models for tool management in automated manufacturing. *Management Science* **39** (1993) 549-567.
41. Gunasekaran A., Martikainen T., Yli-Olli P.: Flexible manufacturing systems: An investigation for research and applications. *European Journal of Operational Research* **66** (1993) 1-26.
42. Gupta Y.P., Goyal S.: Flexibility of manufacturing systems: Concepts and measurements. *European Journal of Operational Research* **43** (1989) 119-135.

43. Hall N.G., Kamoun H., Sriskandarajah C.: Scheduling in robotic cells: Classification, Two and three machine cells. Working Paper 93-56, College of Business, The Ohio State University, 1993. To appear in *Operations Research*.
44. Hall N.G., Kamoun H., Sriskandarajah C.: Scheduling in robotic cells: Complexity and steady state analysis. Working Paper 93-55, College of Business, The Ohio State University, 1993.
45. Hall N.G., Potts C.N., Sriskandarajah C.: Parallel machine scheduling with a common server. Working Paper 94-21, College of Business, The Ohio State University, 1994.
46. Hammer P.L, Rader D.J.: Efficient methods for solving quadratic 0-1 knapsack problems. Rutcor Research Report 40-94, Rutgers University, 1994. To appear in *INFOR*.
47. Hanen C., Munier A.: Periodic scheduling of several hoists. In: *Proc. Fourth International Conf. Project Management and Scheduling* (1994), 108-110.
48. Hirabayashi R., Suzuki H., Tsuchiya N.: Optimal tool module design problem for NC machine tools. *J. of the Operations Research Society of Japan* **27** (1984) 205-228.
49. Hoffman A.J., Kolen A.W.J., Sakarovitch M.: Totally balanced and greedy matrices. *SIAM Journal on Algebraic and Discrete Methods* **6** (1985) 721-730.
50. Hwang S.: A constraint-directed method to solve the part selection problem in flexible manufacturing systems planning stage. In: Stecke K.E. and Suri R., eds. *Proc. of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems*. Elsevier Science Publishers B.V., Amsterdam (1986), 297-309.
51. Hwang S., Shogan A.W.: Modelling and solving an FMS part selection problem. *International Journal of Production Research* **27** (1989) 1349-1366.
- 51a. Ioachim I., Soumis F.: Schedule efficiency in a robotic production cell. *International Journal of Flexible Manufacturing Systems* **7** (1995) 5-26.
52. Jaikumar R., Van Wassenhove L.N.: A production planning framework for flexible manufacturing systems. *Journal of Manufacturing and Operations Management* **2** (1989) 52-79.
53. Jeng W.D., Lin J.T., Wen U.P.: Algorithms for sequencing robot activities in a robot-centered parallel-processor workcell. *Computers and Operations Research* **20** (1993) 185-197.
54. Johnson E.L., Mehrotra A., Nemhauser G.L.: Min-cut clustering. *Mathematical Programming* **62** (1993) 133-151.
55. Kamoun H., Hall N.G., Sriskandarajah C.: Scheduling in robotic cells: Heuristics and cell design. Submitted for publication in *Management Science*.
- 55a. Kats V., Levner E.: The constrained cyclic robotic flowshop problem: A solvable case. In: Levner E., ed., *Intelligent Scheduling of Robots and Flexible Manufacturing Systems*. Center for Technological Education, Holon, Israel, 1995, 115-128.
56. King J.R., Nakornchai V.: Machine component group formation in group technology: review and extension. *International Journal of Production Research* **20** (1982) 117-133.
57. King R.E., Hodgson T.J., Chafee F.W.: Robot task scheduling in a flexible manufacturing cell. *IIE Transactions* **25** (1993) 80-87.
58. Kise H.: On an automated two-machine flowshop scheduling problem with infinite buffer. *Journal of the Operations Research Society of Japan* **34** (1991) 354-361.

59. Kise H., Matsuo H., Sullivan R.S.: Optimal cyclic scheduling for automated two-machine flowshops with no buffer. In: *1990 Japan – U.S.A. Symp. on Flexible Automation*. ISCIE, Kyoto (1990), 1179-1185.
60. Kise H., Shioyama T., Ibaraki T.: Automated two machine flowshop scheduling: a solvable case. *IIE Transactions* **23** (1991) 10-16.
- 60a. van de Klundert, J.: *Scheduling Problems in Automated Manufacturing*. Doctoral Dissertation, University of Limburg, Maastricht, The Netherlands, 1996.
61. Kortsarz G., Peleg D. On choosing a dense subgraph. Extended abstract, Dept. of Applied Math. and Comp. Science, The Weizmann Institute, Rehovot, Israel, 1993.
62. Kou L.T.: Polynomial complete consecutive information retrieval problems. *SIAM Journal on Computing* **6** (1977) 67-75.
63. Kouvelis P.: Design and planning problems in flexible manufacturing systems: A critical review. *Journal of Intelligent Manufacturing* **3** (1992) 75-99.
64. Kouvelis P., Lee H.L.: Block angular structures and the loading problem in flexible manufacturing systems. *Operations Research* **39** (1991) 666-676.
65. Kuhn H.: *Einlastungsplanung von flexiblen Fertigungssystemen*. Physica-Verlag, Heidelberg, 1990.
66. Kusiak A.: Applications of operational models and techniques in flexible manufacturing systems. *European Journal of Operational Research* **24** (1986) 336-345.
67. van Laarhoven P.J.M., Zijm W.H.M.: Production preparation and numerical control in PCB assembly. *The International Journal of Flexible Manufacturing Systems* **5** (1993) 187-207.
68. Lawler E.L.: *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, New York, 1976.
69. Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G., Shmoys D.B.: Sequencing and scheduling: Algorithms and complexity. In: Graves S.C., Rinnooy Kan A.H.G. and Zipkin P.H., eds. *Logistics of Production and Inventory*. Elsevier Science Publishers, Amsterdam (1993), 445-522.
- 69a. Lei L.: Determining the optimal starting times in a cyclic schedule with a given route. *Computers and Operations Research* **20** (1993) 807-816.
70. Lei L., Wang T.J.: A proof: The cyclic hoist-scheduling problem is NP-complete. Working Paper 89-16, Graduate School of Management, Rutgers University, 1989.
71. Lei L., Wang T.J.: The minimum common-cycle algorithm for cyclic scheduling of two material handling hoists with time window constraints. *Management Science* **37** (1991) 1629-1639.
72. Lei L., Wang T.J.: On the optimal cyclic schedules of single hoist electroplating processes. *IIE Transactions* **26** (1994).
73. Lengauer T.: *Combinatorial Algorithms for Integrated Circuit Layout*. John Wiley & Sons Ltd, Chichester, 1990.
- 73a. Levner E., Kats V., Levit V.E.: An improved algorithm for a cyclic robotic scheduling problem. In: Levner E., ed., *Intelligent Scheduling of Robots and Flexible Manufacturing Systems*. Center for Technological Education, Holon, Israel, 1995, 129-141.
- 73b. Levner E., Kogan K., Levin I.: Scheduling a two-machine robotic-cell: A solvable case. *Annals of Operations Research* **57** (1995) 217-232.

74. Lieberman R.W., Turksen I.B.: Crane scheduling problems. *AIIE Transactions* **13** (1981) 304-311.
75. Liu S.-C., Lin L.: Dynamic sequencing of robot moves in a manufacturing cell. *European Journal of Operational Research* **69** (1993) 482-497.
76. Lofgren C.B., McGinnis L.F., Tovey C.A.: Routing printed circuit cards through an assembly cell. *Operations Research* **39** (1991) 992-1004.
77. Mamer J.W., Shogan A.W.: A constrained capital budgeting problem with applications to repair kit selection. *Management Science* **33** (1987) 800-806.
78. Manier M.-A., Varnier C., Baptiste P.: A multi-hoists scheduling problem approach. In: *Proc. Fourth International Conf. Project Management and Scheduling* (1994), 110-115.
79. Mazzola J.B.: Generalized assignment with nonlinear capacity interaction. *Management Science* **35** (1989) 923-941.
80. Mazzola J.B., Neebe A.W., Dunn C.V.: Production planning of a flexible manufacturing system in a material requirements planning environment. *The International Journal of Flexible Manufacturing Systems* **1** (1989) 115-142.
81. Mazzola J.B., Schantz R.H.: Single-facility resource allocation under capacity-based economies and diseconomies of scope. *Management Science* **41** (1995) 669-689.
82. McCormick W.T., Schweitzer P.J., White T.W.: Problem decomposition and data reorganization by a clustering technique. *Operations Research* **20** (1972) 993-1009.
83. McGeoch L.A., Sleator D.D.: A strongly competitive randomized paging algorithm. *Algorithmica* **6** (1991) 816-825.
84. Möhring R.H.: Graph problems related to gate matrix layout and PLA folding. In: Tinhofer G. et al., eds. *Computational Graph Theory*. Springer-Verlag, Vienna (1990), 17-51.
85. Moreno A.A., Ding F.-Y.: Heuristics for the FMS-loading and part-type-selection problems. *The International Journal of Flexible Manufacturing Systems* **5** (1993) 287-300.
86. Nemhauser G.L., Wolsey L.A.: *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, 1988.
- 86a. Ng W.C.: A branch-and-bound algorithm for hoist scheduling of a circuit board production line. *International Journal of Flexible Manufacturing Systems* **8** (1996) 45-65.
87. Oerlemans A.G.: Flexible manufacturing: Concepts and methods. *JORBEL - Belgian J. of Operations Research, Statistics and Computer Science* **34** (1994) 5-22.
88. Phillips L.W., Unger P.S.: Mathematical programming solution of a hoist scheduling program. *AIIE Transactions* **8** (1976) 219-225.
89. Privault C.: *Modèles Mathématiques pour la Gestion Off-line et On-line des Changements d'Outils sur une Machine Flexible*. Doctoral thesis, Université Joseph Fourier, Grenoble, 1994.
90. Privault C., Finke G.: Tool management on NC-machines. In: *Proc. Int. Conf. on Indust. Engineering and Prod. Management*. Mons, Belgium (1993), 667-676.
91. Rajagopalan S.: Scheduling problems in flexible manufacturing systems. Working Paper, G.S.I.A., Carnegie-Mellon University, Pittsburgh, PA. 1985.
92. Rajagopalan S.: Formulation and heuristic solutions for parts grouping and tool loading in flexible manufacturing systems. In: Stecke K.E. and Suri R., eds. *Proc. of the Second*

- ORSA/TIMS Conference on Flexible Manufacturing Systems. Elsevier Science Publishers B.V., Amsterdam (1986) 311-320.
93. Roger C.: *La Gestion des Outils sur Machines à Commande Numérique*. DEA thesis in Operations Research, Université Joseph Fourier, Grenoble, 1990.
 94. Sethi A.K., Sethi S.P.: Flexibility in manufacturing: A survey. *The International Journal of Flexible Manufacturing Systems* **2** (1990) 289-328.
 95. Sethi S.P., Sriskandarajah C., Sorger G., Blazewicz J., Kubiak W.: Sequencing of parts and robot moves in a robotic cell. *The International Journal of Flexible Manufacturing Systems* **4** (1992) 331-358.
 96. Shapiro G.W., Nuttle H.L.W.: Hoist scheduling for a PCB electroplating facility. *IIE Transactions* **20** (1988) 157-167.
 97. Sodhi M.S., Agnetis A., Askin R.G.: Tool addition strategies for flexible manufacturing systems. *The International Journal of Flexible Manufacturing Systems* **6** (1994) 287-310.
 98. Solot P., Van Vliet M.: Analytical models for FMS design optimization: A survey. *The International Journal of Flexible Manufacturing Systems* **6** (1994) 209-233.
 99. Sriskandarajah C., Hall N.G., Kamoun H., Wan H.: Scheduling large robotic cells. Working Paper, College of Business, The Ohio State University, 1994.
 100. Sriskandarajah C., Ladet P.: Some no-wait shops scheduling problems: Complexity aspect. *European Journal of Operational Research* **24** (1986) 424-438.
 101. Sriskandarajah C., Sethi S.P.: Scheduling algorithms for flexible flowshops: Worst and average case performance. *European Journal of Operational Research* **43** (1989) 143-160.
 102. Stecke K.E.: Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems. *Management Science* **29** (1983) 273-288.
 103. Stecke K.E.: Algorithms for efficient planning and operations of a particular FMS. *The International Journal of Flexible Manufacturing Systems* **1** (1989) 287-324.
 104. Stecke K.E., Kim I.: A study of part type selection approaches for short-term production planning. *The International Journal of Flexible Manufacturing Systems* **1** (1988) 7-29.
 105. Stecke K.E., Talbot F.B.: Heuristics for loading flexible manufacturing systems. In: Raouf A. and Ahmad S.I., eds. *Flexible Manufacturing: Recent Developments in FMS, Robotics, CAD/CAM, CIM*. Elsevier, Amsterdam (1985), 73-85.
 106. Tang C.S., Denardo E.V.: Models arising from a flexible manufacturing machine, Part I: Minimization of the number of tool switches. *Operations Research* **36** (1988) 767-777.
 107. Tang C.S., Denardo E.V.: Models arising from a flexible manufacturing machine, Part II: Minimization of the number of switching instants. *Operations Research* **36** (1988) 778-784.
 108. Tempelmeier H., Kuhn H.: OR-Modelle zur Planung flexibler Fertigungssysteme. *OR Spektrum* **14** (1992) 177-192.
 109. Tempelmeier H., Kuhn H.: *Flexible Fertigungssysteme - Entscheidungsunterstützung für Konfiguration und Betrieb*. Springer, Berlin Heidelberg, 1992.
 110. Van Looveren A.J., Gelders L.F., Van Wassenhove L.N.: A review of FMS planning models. In: Kusiak A., ed. *Modelling and Design of Flexible Manufacturing Systems*. Elsevier Science Publishers B.V., Amsterdam (1986), 3-31.

111. Veeramani D., Upton D.M., Barash M.M.: Cutting-tool management in computer-integrated manufacturing. *The International Journal of Flexible Manufacturing Systems* **3/4** (1992) 237-265.
112. Weingartner H.M.: Capital budgeting of interrelated projects: Survey and synthesis. *Management Science* **12** (1966) 485-516.
113. de Werra D., Widmer M.: Loading problems with tool management in flexible manufacturing systems: A few integer programming models. *The International Journal of Flexible Manufacturing Systems* **3** (1990) 71-82.
114. Whitney C.K., Gaul T.S.: Sequential decision procedures for batching and balancing in FMSs. *Annals of Operations Research* **3** (1985) 301-316.
115. Widmer M.: Job shop scheduling with tooling constraints: A tabu search approach. *Journal of the Operational Research Society* **42** (1991) 75-82.
116. Wolsey L.A.: MIP modeling of changeovers in production planning and scheduling problems. In: Slowinski R., ed., *Semi-Plenary Papers – 14th European Conference on Operational Research*. Poznan, Poland (1995), 197-210.