Maastricht University

The approximability of tool management problems

Citation for published version (APA):

Crama, Y., & van de Klundert, J. (1996). The approximability of tool management problems. (METEOR research memorandum; No. 034). Maastricht: METEOR, Maastricht University School of Business and Economics.

Document status and date:
Published: 01/01/1996

Document Version:
Publisher's PDF, also known as Version of record

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

The Approximability of Tool Management Problems

Yves Crama [1] [2]
Joris van de Klundert [2]

November 18, 1996

[1]Département de Gestion, Université de Liège, 4000 Liège, Belgium
[2]Department of Quantitative Economics, Faculty of Economics and Business Adminstration, University of Limburg, Maastricht, The Netherlands

Abstract

Since the introduction of flexible manufacturing systems, researchers have investigated the various planning and scheduling problems that the users of such systems are facing. Several of these problems are not encountered in more classical production settings, and so called tool mamagement problems appear to be among the more fundamental ones of these problems. Many researchers have proposed approximate solution techniques for tool management problems, most of which are hard to solve. In this paper we investigate the quality of these algorithms by means of worst case analysis. We show that all of the polynomial approximation algorithms proposed to date exhibit rather poor worst case behavior. We also investigate the complexity of solving these problems approximately. In this respect, we investigate the interrelationships between tool management problems and their relationships with other well known combinatorial problems, such as Set Covering and Clique, and give several negative results on the approximability of various tool management problems.

1 Introduction

Regardless of the precise definition of flexibility in the term flexible manufacturing systems, the ability of machines to perform various operations on various products or parts, is a most vital component of this flexibility. This flexibility of the machines is achieved by equipping them with a tool magazine, which enables the machines to hold a set of tools from which, depending on the operation the machine has to perform, it uses one tool or another. The resulting flexibility may be advantageous from a strategic or even tactical viewpoint, it comes at a price. The complexity of the operational planning and scheduling of the machines increases considerably, even when considering the machines in isolation. Apart from the part sequencing decisions, that normally constitute a solution to a single machine scheduling problem, one has to specify tool handling decisions. Hence, problems concerning the scheduling of a single flexible machine, which are so fundamental to the scheduling of flexible manufacturing systems, differ essentially from classical single machine scheduling problems. For this reason, such problems received considerable attention in the literature, since the introduction of these machines.

A good deal of the literature concerning tool management problems in single machine scheduling problems takes a practical position. The authors extract a problem from a more or less real life situation and propose an approximate solution strategy. Other authors, by contrast, are interested in the mathematical models underlying one or several of these problems. In this paper we also are primarily interested in mathematical properties of single flexible machine scheduling problems. More specifically, we will be interested in the worst case ratios of polynomial approximation algorithms for single flexible machine scheduling problems. This means that we borrow from both theoretical as well as applied papers. Many of the algorithms as they are proposed in the literature have appeared in applied papers. On the other hand, a proper classification of the complexity of the models and their approximability requires a more theoretical background. In subsequent sections we introduce the problems studied in this paper and study their complexity, and the complexity of solving them approximately, at length. We now give a brief overview of related results.

Few attempts have been made to date to classify the problems discussed in this paper with respect to their approximability. Rajagopalan [1985] establishes that a simple 'First Fit Decreasing' heuristic 'can do almost arbitrarily bad' for certain batching problems. Kortsarz and Peleg [1993] consider a special case of the batch selection problem that may be interpreted as the problem of finding a densest induced subgraph. They present an approximation algorithm for finding a dense subgraph of a graph $G(V, E)$ of cardinality at most C, whose worst case ratio is $O(|V|^{\frac{7}{18}})$. Goldschmidt et al. [1992] also propose several approximation algorithms for special cases of both the batch selection and the related job grouping problem, with constant worst case ratios or worst case ratios that are linear in the tool magazine capacity C. Goldschmidt et al. [1993] suggest a dynamic programming formulation for the batch selection problem and discuss conditions under which it can be implemented to run in polynomial time.

Of course, there is also a vast amount of literature dealing with the same type of problems but focusing on other topics than their approximability. Hirabayashi et al. [1984] propose a quite general mathematical programming formulation for the batch selection problem. Crama & Mazzola [1995] investigate polyhedral properties of this formulation. Dietrich, Lee & Lee [1993], and Johnson, Mehrotra, and Nemhauser [1993] also study valid inequalities for (special cases of) the batch selection problem. Hwan & Shogan [1989] propose a branch & bound algo-

rithms for the batch selection problem and a langrangean relaxation based solution technique. Hirabayashi et al. [1984] propose a set covering formulation for the job grouping problem, and develop a branch & bound algorithm for the batch selection problem. Crama & Oerlemans [1994] solve the job grouping problem by a column generation approach that is based on the aforementioned set covering formulation. For a more detailed overview of models and solution techniques for single machine tooling problems we refer to Crama [1995].

In the next section, we first discuss several of the basic single flexible machine scheduling problems and we also briefly discuss some mathematical models, their complexity, and their relationships with other combinatorial problems. Section 3 investigates the worst case behavior of several algorithms as they are proposed in the literature to date. This investigation will lead us to the conclusion that all of these algorithms have very poor worst case behavior. However, as yet it is not known whether there (can) exist polynomial approximation algorithms with a better worst case behavior. We give negative results on this topic in Section 4. In section 5 we conclude by discussing directions for further research.

2   Models and complexity

In this section we briefly discuss several models as they arise naturally in the context of flexible machine scheduling, and their complexity. Our main purpose is to facilitate the analysis in the subsequent sections.

To start with, let us take a look at the physical characteristics of flexible machine scheduling. First of all, there is a *machine* on which a set of *jobs* or *parts* have to be processed. (We use jobs and parts interchangeably). Processing means that the machine performs one or several operations on these jobs, and the execution of each of these operations requires one or more *tools*. The machine can store tools in its *tool magazine*. In this paper, we assume the magazine contains C slots, and that each tool requires exactly one slot, although more general models are possible of course (Crama [1995]). Using tools from the tool magazine requires little set up time, and thus, as long as a sequence of jobs that have to be processed only requires tools that are present on the machine, in the tool magazine, set up times are (negligibly) small. However, if the number of tools required by a sequence of jobs exceeds the *tool magazine capacity C*, it is unavoidable that some tool is removed from the magazine to be replaced by another tool. We refer to such an event as a *switch*. Switches can not take place during processing operations. Further, switches usually take nonnegligible time, and thus set up times are 'large' whenever a switch is required.

Under many reasonable objective functions of such scheduling problems, e.g. makespan minimization, we have to minimize the sum of the set up times. The total set up time is usually computed in one of the two following ways. If tools cannot be switched simultaneously, total set up time depends linearly on the number of tool switches. On the other hand, supposing that tool switches may be performed (completely) simultaneously, total set up time depends linearly on the number of *switching instants*. Let us define a *loading strategy* as a specification of the contents of the tool magazine at the beginning of the processing of each job. We now identify the following four basic scheduling problems, whose names are taken from Crama [1995]:

2

1. **Tool switching :** The problem of finding an input sequence for the parts and a loading strategy for the tool magazine with minimum total set up time, in case total set up time depends linearly on the number of switches.

2. **Loading problem :** The problem of finding for a given part input sequence, a loading strategy with minimum total set up time, in case total set up time depends linearly on the number of switches.

3. **Job grouping :** The problem of finding an input sequence for the parts and a loading strategy for the tool magazine with minimum total set up time, in case total set up time depends linearly on the number of switching instants.

4. **Batch selection :** The problem of finding the largest group of jobs that can be processed without tool switches.

We have enumerated here the optimization versions of the four problems, but we refer to the decision versions by the same name.

The loading problem is a special case of the tool switching problem that is interesting not only from a computational viewpoint ; both the tool switching problem and the loading problem also arise in the context of (mainframe) computer memory management (Blazewicz & Finke [1994]).

Notice that the only difference between the tool switching problem and the job grouping problem is the underlying cost structure. However, the job grouping problem is in a way less sensitive to the exact part input sequence. Given two consecutive switching instants, the order in which the jobs are processed between these instants is irrelevant. Let us therefore call a set of jobs that can be processed without tool switches, a *batch*. Then, the job grouping problem boils down to finding a partitioning of the jobs in a minimum number of *batches*. These observations motivate our interest for the fourth problem, the batch selection problem. Sometimes batches are referred to as groups, which explains the name job grouping.

Without going into any further detail of the mathematical models that exist for the four problems, let us discuss their complexity status first. Crama, Kolen, Oerlemans & Spieksma [1994], show that the tool switching problem is strongly NP-Complete, even for fixed $C \geq 2$. The loading problem is investigated by Tang & Denardo [1988], Crama, Kolen, Oerlemans & Spieksma [1994], and Privault & Finke [1993] who provide a network flow formulation.

The job grouping problem has also been shown to be NP-hard by several authors. Crama & Oerlemans [1994] show that the problem is strongly NP-complete, even for fixed $C \geq 3$, and that it is NP-complete to decide whether there exists a partitioning of size two. By showing that the well known set covering problem may be viewed as a special case of the job grouping problem in which all maximal batches are known, they also establish that the problem remains strongly NP-complete in such a case. In general, however, the problem is in a way even harder, since the batch selection problem is known to be strongly NP-complete even when each part requires two tools (Gallo, Hammer & Simeone [1980]).

In this paper we will be primarily interested in the approximability of the job grouping problem and the batch selection problem. The loading problem is polynomially solvable, making the study of approximation algorithms for this problem less interesting. The approximability of tool switching is briefly discussed in relation to the approximability of job grouping in Section 4.

For the moment we restrict the analysis to job grouping and batch selection and we first make clear that both problems require the same data. An instance consists of a tool magazine capacity C, a set J of jobs $p_1, \ldots, p_n$ and a set T of tools $t_1, \ldots, t_m$. Further, for each job $p_i, i = 1, \ldots, n$ we specify which tools it requires. To this purpose, we introduce a $(0, 1)$ matrix A, whose m rows correspond to tools and whose n columns correspond to the jobs. Naturally, we let $a_{ij} = 1$ if tool t_i is required by job p_j, and zero otherwise. Thus, an instance of (an optimization version of) job grouping or batch selection is completely specified by a $(0, 1)$ matrix A and a positive integer C, the tool magazine capacity. In the remainder, we assume that for each pair of jobs, the sets of tools required for each of the jobs do not contain one another as a subset.

As such, the batch selection problem is to find a maximum cardinality subset J of the columns, such that $|\{i | \sum_{j \in J} a_{ij} > 0\}| \leq C$. The matrix A, which we shall refer to as the *tool-job matrix*, may also be viewed as the node incidence matrix of a hypergraph $H(V, E)$. Each row i corresponds to a vertex $v_i \in V$, and each column j to a hyperedge $e_j \in E$. Indeed, $a_{ij} = 1$ indicates that edge e_j contains vertex v_i. In this setting the batch selection problem is to find a densest subset of the vertices of cardinality at most C, i.e. a subset of the vertices of cardinality C, whose induced subgraph contains the largest number of hyperedges. Now, in the decision version of clique, one may have to find a subset V' of the vertex set, with $|V'| = C$, such that the subgraph induced by V' contains $\frac{1}{2}|V'|(|V'| - 1)$ edges. This establishes that batch selection is already NP-complete when each job requires two tools (Gallo, Hammer & Simeone [1988]).

Of course, job grouping may also be interpreted in terms of $(0, 1)$ matrices and hypergraphs. One has to find a minimum cardinality set of subsets $S_1, \ldots, S_K$ of the vertices, such that the subhypergraphs $H^i(S_i, E_{S_i})$ induced by these subsets S_i form a covering of H, i.e. $\cup_i E_{S_i} = E$. As observed before, this problem is already NP-hard when all maximal subhypergraphs are known. In such a case we obtain an ordinary covering problem as follows. We introduce a job-group matrix B, in which the rows correspond to jobs and the columns correspond to groups. Indeed $b_{jk} = 1$ if job p_j is in group g_k, and zero otherwise. Letting the matrix B be the constraint matrix, the job grouping problem is then turned into a set covering problem (should we give an example of this transformation?).

We informally conclude that the job grouping problem does not appear to be easier than set covering, and that the batch selection problem is closely related to clique. Both set covering and clique are notoriously hard from a viewpoint of approximation. We discuss their exact status in Section 4. For the moment however, we should have modest expectations with respect to the worst case ratios of polynomial approximation algorithms for job grouping and batch selection as they are given in the next section.

3 Polynomial time approximation algorithms and worst case ratios

In this section we overview approximation algorithms for job grouping and batch selection, as they have been proposed in the literature. We also study their worst case behavior.

From the literature we have extracted the following list of approximation algorithms for the batch selection problem. Each algorithm is characterized by a selection rule, which specifies how

to select jobs or tools. The selection rule is to be repeatedly applied as long as some stopping criterion is not satisfied, e.g. the total number of tools (required by the selected jobs) does not exceed the tool magazine capacity.

1. MIMU rule (Tang and Denardo [1988b]) : Select the job that has the largest number of tools in common with the jobs already in the batch. In case of a tie, select the job which requires the smallest number of additional tools : Maximal Intersection, Minimal Union.

2. MI (Maximal Intersection) rule, break ties arbitrarily.

3. MU (Minimal Union) rule, break ties arbitrarily.

4. Whitney & Gaul rule (Whitney & Gaul [1985]) : Let $t(Y)$ be the number of tools required by the jobs in the set of jobs Y. Let B be the set of already selected jobs. Select the job p that maximizes $(t(B \cup \{p\}) + 1)/(t(\{p\}) + 1)$.

5. Rajagopalan rule (Rajagopalan [1985]) : Define the weight of each tool to be the number of jobs that require it among the jobs not yet assigned to the batch. Select the job for which the sum of the weights of the tools that are to be added when this job is selected is maximum.

6. Modified Rajagopalan rule (Crama & Oerlemans [1992]) : Define the weight of a tool to be the number of jobs already selected that require this tool. Select the job for which the sum of the weights of the tools needed by this job is maximum.

7. Chaillou, Hansen & Mahieu [1989] rule : Create an initial batch consisting of all jobs by selecting all tools. Then, iterate deleting tools from the set of selected tools until the number of selected tools equals the magazine capacity. In each iteration delete the tool which causes the smallest number of jobs to be eliminated from the batch.

8. Marginal gain rule (Dietrich, Lee & Lee [1991]) : Define the weight of a job to be the number of jobs that can be added without tool addition when this job is selected. Select the job with maximum weight.

Rajagopalan [1985] also proposed a rule of the Maximal Union type, and showed that it can perform arbitrarily bad on certain instances. This was the motivation to introduce rule 6 described above.

Every approximation algorithm A_{BS} for the batch selection problem gives rise to an approximation algorithm for job grouping A_{JG} in the following manner. We apply A_{BS} to find a first group. Then we eliminate the jobs in this group from the instance and we apply A_{BS} again. We repeat this procedure until there are no jobs left. The sequence of groups that is iteratively produced by A_{BS} forms a solution of the job grouping problem on the same data. As a matter of fact, all heuristics for the job grouping problem known to the authors are of this type.

In the remainder of this section, we analyse the worst case ratio of the rules given above. In the instances we present, the optimal solution of the job grouping problem consists of a set of batches each of which is an optimal solution of the batch selection problem. Furthermore, repeatedly applying one of the aforementioned selection rules results in a sequence of batches containing the same number of jobs. Hence, the worst case performance of a rule on the job grouping problem is the reciprocal of the worst case ratio of this rule on the batch selection

5

$$\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & \quad 1 & 1 & 1 & 0 & 0 & 0 & \quad 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & \quad 1 & 0 & 0 & 1 & 1 & 0 & \quad 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & \quad 0 & 1 & 0 & 1 & 0 & 1 & \quad 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & \quad 0 & 0 & 1 & 0 & 1 & 1 & \quad 0 & 0 & 1 & 0 & 1 & 1 \\
\\
1 & 1 & 1 & 1 & 1 & 1 & \quad 0 & 0 & 0 & 0 & 0 & 0 & \quad 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \quad 1 & 1 & 1 & 1 & 1 & 1 & \quad 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \quad 0 & 0 & 0 & 0 & 0 & 0 & \quad 1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}$$

Figure 1: Tool job matrix of a worst case instance for heuristics 1,2,3,6 ($k = 4$)

problem. Therefore we primarily consider the behavior of the rules on the batch selection problem.

Before we analyse the worst case behavior of the proposed heuristics, let us present an upperbound on the worst case ratio of any heuristic (for the batch selection problem). For a magazine capacity C, the maximum number of jobs in a group can be seen to be $\begin{pmatrix} C \\ C/2 \end{pmatrix} = \Omega(\frac{2^C}{\sqrt{C}})$, whereas any heuristic finds a group of at least a single job. Hence, $\begin{pmatrix} C \\ C/2 \end{pmatrix}$ is an upperbound on the worst case ratio of any heuristic. We show here that rules 1 to 6 have worst case ratios in this order of magnitude, and may therefore perform arbitrarily bad :

Theorem 1 The worst case ratio of heuristics 1,2,3, and 6 is at least $\Omega(\frac{2^C}{C^{3/2}})$ and $\Omega(n/\log^2 n)$.

Proof. Let k be some even integer. We create instances in which there are k top tools and each job requires $k/2$ of them. We first consider an instance in which there are $k/2 + 1$ bottom tools, of which each job requires only 1. The tool magazine capacity $C = k + 1$. We have a set of $k/2 + 1$ jobs for each possible choice of $k/2$ top tools, one job in the set for each possible bottom tool. Thus we have $\begin{pmatrix} k \\ k/2 \end{pmatrix} \times (k/2 + 1)$ jobs. (see Figure 7.1.) Obviously, for the batch selection problem, an optimal batch is the set of jobs requiring the same bottom tools. Moreover, the optimal solution for the job grouping problem is to form $k/2 + 1$ groups (each of them an optimal batch), one for each bottom tool.

It is not hard to see that heuristics 1,2,3 and 6 start with an arbitrary job and may subsequently select the job requiring the same top tools but another bottom tool. In this way they obtain batches of size $k/2 + 1$, where the optimal batches consist of $\begin{pmatrix} k \\ k/2 \end{pmatrix}$ jobs. The ratio between the number of jobs in an optimal batch and the number of jobs in a batch found by either of the heuristics is $\Omega(\frac{2^C}{C^{\frac{3}{2}}})$. Notice that in this instance selecting jobs randomly could not have led to a worse solution, be it for the batch selection problem or for the job grouping problem. ∎

Theorem 2 The worst case ratio of heuristics 4,5 is at least $\Omega(\frac{2^C}{\sqrt{C}})$ and $\Omega(n)$.

$$\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & & 0 & 0 & 1 & 0 & 1 & 1
\end{pmatrix}$$

Figure 2: Tool job matrix of a worst case instance for heuristics 4 and 5, ($k = 4$)

Proof. Heuristics 4 and 5 achieve the ratio claimed in the theorem on the following instance. Let k again be some even integer. Again, there are two sets of tools, top tools and bottom tools. Each job requires either some of the top tools or some of the bottom tools. Hence we can also speak of toptool jobs and bottomtool jobs. There are k top tools and k bottom tools, and each job requirs $k/2$ tools, i.e. $k/2$ top tools or $k/2$ bottom tools. The tool magazine capacity $C = k$. We have a job for each possible choice of $k/2$ top tools out of k top tools, and a job for each possible choice of $k/2$ bottom tools. Thus we have $\binom{k}{k/2} \times 2$ jobs. (see Figure 7.2.) Obviously, both the set of all toptool jobs and the set of all bottomtools jobs are optimal solutions to the batch selection problem. Moreover, the optimal solution for the job grouping problem is to form 2 groups (each of them an optimal batch).

It is left to the reader to check that rules 4 and 5 may pick batches of size 2, consisting of a toptool job and a bottomtool job, whereas the optimal batches consist of $\binom{k}{k/2}$ jobs, yielding a worst case ratio of $\Omega(\frac{2^C}{\sqrt{C}})$ and $\Omega(n)$. ∎

Notice that the bounds of Theorem 2 realize the aforementioned upperbound on the worst case ratio of any algorithm. In case all jobs require the same number of tools, rule 4 boils down to the Maximum Union rule. Rajagopalan [1985] already analyzed this Maximum Union rule and showed that it can do 'arbitrarily bad'.

Rule 7 does not solve the instances proposed in the proofs of Theorem 7.1 and Theorem 7.2 to optimality, but it does not perform as poorly as rules 1-6.

Theorem 3 Heuristic 7 has worst case ratio of at least $\Omega\left(\binom{\sqrt{C}}{\frac{\sqrt{C}}{2}}/\sqrt{C}\right)$ and $\Omega(n/\log n)$.

Proof. Again, we introduce top and bottom tools. Let k again be some even integer. Define the two types of tools as follows. There are k top tools, each job requiring $k/2$ of them. There are $k/2 \times (k/2 + 1)$ bottom tools, of which each job requires $k/2 \times k/2$. The tool magazine capacity

$C = (k/2)^2 + k$. We divide the bottom tools into $k/2 + 1$ sets, each consisting of $k/2$ tools. Each job requires all tools in all but one of these sets, and none of the bottom tools in the remaining set. We have a set of jobs for each possible choice of $k/2$ top tools out of k top tools. This set contains one job for each possible bottom tool requirement. Thus we have $\binom{k}{k/2} \times (k/2 + 1)$ jobs (see Figure 7.3). Obviously, for the batch selection problem, an optimal batch is a set of $\binom{k}{k/2}$ jobs with identical bottom tool requirements. Moreover, an optimal solution for the job grouping problem is to form $k/2 + 1$ groups (each of them an optimal batch), one for each possible bottom tools requirement.

Let us now study the behavior of heuristic 7. The heuristic must delete $k/2$ tools. We claim that the heuristic deletes (or may delete) $k/2$ top tools. Suppose after iteration $i, i \in \{0, \ldots, k/2 - 1\}$ the heuristic has not deleted bottom tools yet. Then the number of remaining jobs equals $\binom{k-i}{k/2} \times (k/2 + 1)$. By symmetry, every bottom tool is required by a fraction of $\frac{k/2}{k/2+1}$ of all jobs. Similarly, every top tool is required by a fraction of

$$\frac{\binom{k-i-1}{k/2-1}}{\binom{k-i}{k/2}} = \frac{k/2}{k-i} \leq \frac{k/2}{k/2 + 1}$$

of all jobs. Hence the heuristic may select a top tool again. After $k/2$ such iterations we thus end up with $k/2$ top tools and all bottom tools and a batch of $k/2 + 1$ jobs. Now, since $C < k^2$, this yields a ratio of

$$\Omega\left(\binom{\sqrt{C}}{\sqrt{C}/2} / \sqrt{C}\right)$$

for the batch selection problem.

We now show that the heuristic performs equally bad on the job grouping problem. We show that when solving the job grouping problem by repeatedly appying heuristic 7 to form batches, we may get a batch for each possible top tool requirement. In view of the discussion above, it suffices to notice that for every set of jobs J such that J contains at least two jobs with distinct top tool requirements, there is always some top tool that is required by at most $\frac{k/2}{k/2+1}$ jobs. ∎

The heuristic proposed by Dietrich Lee & Lee [1991] solves the previous instances optimally. We have, however, the following theorem :

Theorem 4 Heuristic 8 has worst case ratio of at least $\Omega\left(\binom{\sqrt{C}}{\frac{\sqrt{C}}{2}}\right)$, and $\Omega(n)$.

$$
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & \; & 1 & 1 & 1 & 0 & 0 & 0 & \; & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & \; & 1 & 0 & 0 & 1 & 1 & 0 & \; & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & \; & 0 & 1 & 0 & 1 & 0 & 1 & \; & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & \; & 0 & 0 & 1 & 0 & 1 & 1 & \; & 0 & 0 & 1 & 0 & 1 & 1 \\
 & & & & & & & & & & & & & & & & & & & \\
1 & 1 & 1 & 1 & 1 & 1 & \; & 1 & 1 & 1 & 1 & 1 & 1 & \; & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & \; & 1 & 1 & 1 & 1 & 1 & 1 & \; & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & \; & 0 & 0 & 0 & 0 & 0 & 0 & \; & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & \; & 0 & 0 & 0 & 0 & 0 & 0 & \; & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & \; & 1 & 1 & 1 & 1 & 1 & 1 & \; & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & \; & 1 & 1 & 1 & 1 & 1 & 1 & \; & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{pmatrix}
$$

Figure 3: Tool job matrix of a worst case instance for heuristic 7, $(k = 4)$

Proof. Let m be an even integer. First, we construct a dummy tool job matrix D as follows. For each possible subset of cardinality $m/2$ of the integers $1, \ldots, m$, we have a column in D. Each row of D corresponds to a 2-element subset of the integers $1, \ldots, m$. Entry $d_{i,j} = 1$ if the i-th 2-element subset is contained in the j-th $(m/2)$-element subset, and zero otherwise (see Figure 7.4).

The jobs corresponding to the columns of D satisfy the following property (P) :

Let p_j, p_k and p_l be three jobs. There exists a row i of D such that $p_{ij} = 1, p_{ik} = p_{il} = 0$.

From the matrix D we derive an instance that yields the desired ratio as follows. Again we introduce top tools and bottom tools. There are two sets of bottom tools, called bottom sets, each job requires all the tools in one of the bottom sets, and none of the other bottom tools. Each set of bottom tools consists of

$$
\begin{pmatrix} m \\ 2 \end{pmatrix} - \begin{pmatrix} m/2 \\ 2 \end{pmatrix}
$$

tools. The top tools correspond to the tools as defined by the matrix D. Moreover for each column of D we have one job for each of the aforementioned bottom sets. Thus we have $\begin{pmatrix} m \\ m/2 \end{pmatrix} \times 2$ jobs. We set $C = 2 \begin{pmatrix} m \\ 2 \end{pmatrix} - \begin{pmatrix} m/2 \\ 2 \end{pmatrix}$. Then the heuristic proposed by Dietrich, Lee & Lee may pick batches in which all bottom tools are required and only $\begin{pmatrix} m/2 \\ 2 \end{pmatrix}$ top tools as follows. It selects an arbitrary job to begin with. All jobs whose top tool requirements and bottom tool requirements differ from the requirements of the already selected job, can not be added to the batch : the batch would require more than C tools. The single job that has the same top tool requirements, but requires the other set of bottom tools, has weight zero. Moreover,

9

	123	124	125	126	...	456
12	1	1	1	1		0
13	1	0	0	0		0
14	0	1	0	0		0
15	0	0	1	0		0
16	0	0	0	1		0
23	1	0	0	0		0
⋮						
56	0	0	0	0	...	1

Figure 4: The dummy tool job matrix D for $m = 6$.

because of (P), all jobs that have the same bottom tool requirements, but different top tool requirements, also have weight zero. Thus, the rule may select the single job with the same top tool requirements, filling up the tool magazine completely. In this way the rule selects groups of size 2, whereas the optimal solution contains 2 groups of size $\begin{pmatrix} m \\ \frac{m}{2} \end{pmatrix}$. Since $C = \Theta(m^2)$ this yields the desired ratio. ∎

Although, the bounds we have obtained in this section on the worst case behavior of the various algorithms imply a rather poor worst case performance, we have not shown that they are tight. Thus, some of the bounds, especially the bounds derived in Theorems 3 and 4 may be subject to improvement.

4 Negative results on the approximability of job grouping, batch selection and tool switching

The results in the previous section being discouraging, the question arises whether polynomial approximation algorithms with better worst case ratios can exist. The relationship between clique and batch selection and the relationship between job grouping and set covering, that was informally established in the Section 2, suggest that good approximation algorithms may not exist : both clique and set covering are notoriously hard to approximate. It has long been open whether a polynomial approximation algorithm with a constant worst case ratio could exist for clique. It was well known however, see Garey & Johnson [1979], that the existence of a polynomial approximation algorithm with a constant worst case ratio for clique would imply the existence of a polynomial approximation scheme for this problem, which was regarded to be unlikely to exist. Only recently, Arora et al. [1993] have shown that a polynomial approximation scheme for clique cannot exist unless P = NP. More precisely, they have shown that there is some ϵ such that there cannot exist a polynomial time $O(m^\epsilon)$ approximation algorithm for clique (where m is the number of nodes) unless P = NP. In this section we obtain a result in the spirit of the aforementioned result of Garey & Johnson [1979] for batch selection.

The same breakthrough that led to the negative result regarding the approximability of clique, enabled Lund & Yannakakis [1993] to show that for any $d, 0 < d < \frac{1}{4}$, a polynomial approximation algorithm for set covering with worst case ratio $d \log n$ cannot exist, unless NP is contained in $DTIME[n^{poly \log n}]$. In this section, we briefly discuss the implications of this result for job grouping. We also establish a relationship between the approximability of job grouping and tool switching.

A first negative result is the following theorem, in the spirit of Theorem 6.12 in Garey & Johnson [1979] :

Theorem 5 Either the batch selection problem can be solved by a polynomial time approximation scheme, or else there is no polynomial time approximation algorithm with constant worst case ratio for this problem.

Proof. Suppose that H is a polynomial time approximation algorithm with finite worst case ratio $r \geq 1$. Let I be an instance of batch selection. For any ϵ let l_ϵ be the smallest integer such that $r^{\frac{1}{l_\epsilon}} < 1 + \epsilon$. We construct an approximation scheme S that delivers a solution with value $S(I)$ for I, such that $OPT(I)/S(I) \leq 1 + \epsilon$. Moreover its running time depends polynomially on n, m and $\log C$ and on the running time of H, as required, and is exponential in l_ϵ.

To obtain a result as mentioned in the theorem, we need a method to 'square' an instance. Given an instance of batch selection, consisting of a magazine capacity C and a tool job matrix A, squaring gives in polynomial time a new instance of batch selection, with capacity C' and tool-job matrix A' such that

1. From any solution of I' with value $s'(I')$ we can construct a solution of I with value $s(I)$ such that $s(I)^2 \geq s'(I')$.

2. Moreover, $OPT(I') = OPT(I)^2$.

Notice that this suffices to prove the Theorem. Given an instance of batch selection, we square $\lceil \log(l_\epsilon) \rceil$ times, we then apply H and can construct a solution for the original problem instance of value at least $OPT(I)/(1 + \epsilon)$, by definition of l_ϵ.

We construct C' and A' from C and A as follows. The new magazine capacity C' is equal to $(C + 3)C$. There are n^2 columns and $(C + 3)m$ rows in A'. For each column of A, we get n columns in A'. For each row of A we get $C + 3$ rows in A'. Column $a_{(j-1)n+l}$ depends on the columns a_j and a_l ($j = 1, \ldots, n$ $l = 1, \ldots, n$). To be precise, the transformation is as follows :

1. if $a_{i,j} = 1$, then $a'_{i,(j-1)n+l} = 1$, for all $i = 1, \ldots, m$, $j, l = 1, \ldots, n$;

2. if $a_{i,l} = 1$, then $a'_{(C+2)m+i,(j-1)n+l} = 1$, for all $i = 1, \ldots, m$, $j, l = 1, \ldots, n$;

3. if $a_{i,j} = 1$ or $a_{i,l} = 1$, then $a'_{m+(C+1)(i-1)+k,(j-1)n+l} = 1$, for $k = 1, \ldots, C + 1, i = 1, \ldots, m$, $j, l = 1, \ldots, n$.

4. $a'_{ij} = 0$ otherwise.

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Figure 5: Batch selection instance I, $C = 2$, A

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ & & & & & & & & \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ & & & & & & & & \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Figure 6: The instance $I' = I^2$, $C' = 10, A'$.

An example of this transformation is given in Figures 5 and 6. Matrix A' may be interpreted as follows. For each ordered pair of jobs (p_j, p_l) in I, we obtain a job, $p'_{(j-1)n+l}$ in I'. Further, there are three sets of tools. One set, to be referred to as top tools, consists of tools $t'_1, \ldots, t'_m$ and duplicates the tool requirements of p_j. Another set consists of tools $t'_{(C+2)m+1}, \ldots, t'_{(C+2)m+m}$, duplicates the tool requirements of job p_l. The tools in this second set will be referred to as bottom tools. The third set of tools depends on both p_j and p_l. Indeed if some tool t_i is required by either p_j or p_l, then the tools $t'_{m+(C+1)(i-1)+k}, k = 1, \ldots, C + 1$ are required for $p'_{(j-1)n+l}$ and $p'_{(l-1)n+j}$.

It follows from this description that if a job requires tool $t'_{m+(C+1)(i-1)+p}$, for some $1 \leq i \leq n, 1 \leq p \leq C + 1$ then it requires $t'_{m+(C+1)(i-1)+q}$, for all $1 \leq q \leq C + 1$. Thus, in any solution, it is pointless to have exactly $r, 0 < r < C + 1$ of the tools $t'_{m+(C+1)(i-1)+q}, 1 \leq q \leq C + 1$. In the following we may therefore assume, without loss of generality, that a solution contains either all or none of the tools $t'_{m+(C+1)(i-1)+q}, 1 \leq q \leq C + 1$, for all i.

Consider again some job A_j in I', i.e. some column of A'. The definition of A' implies that for all $i, 1 \leq i \leq m$, if $a_{i,j} = 1$, then $a_{m+(C+1)(i-1)+q,j} = 1$ for all $1 \leq q \leq C + 1$. We conclude again that there is no benefit in selecting tool t_i in a solution unless all tools $t'_{m+(C+1)(i-1)+q}, 1 \leq q \leq C + 1$ are selected. Similarly, if a job requires tool $t'_{(C+2)m+i}$, then it also requires tools $t'_{m+(C+1)(i-1)+q}$, for all $1 \leq q \leq C + 1$. Again, there is no use in selecting tool $t'_{(C+2)m+i}$, unless tools $t'_{m+(C+1)(i-1)+q}$ are selected, for all $1 \leq q \leq C + 1$. For convenience, we call *block* i $(1 \leq i \leq m)$, the set of tools $t'_{m+(C+1)(i-1)+q}$ where $1 \leq q \leq C + 1$. We call t'_i the *corresponding top tool*, and call $t'_{m+(C+1)m+i}$ the *corresponding bottom tool*.

The above observations lead us to the conclusion that if we select w blocks, then we select at most w top tools and at most w bottom tools. Since $C' = (C + 3)C$, and each block consists of $C + 1$ tools, this implies that we may as well select at least C blocks. Let us first consider the case where we select at least $C + 1$ blocks. Since $C' = (C + 3)C < (C + 2)(C + 1)$, we cannot select $C + 2$ blocks. Hence suppose we select $C + 1$ blocks. This means we can select $C(C + 3) - (C + 1)(C + 1) = C - 1$ top and bottom tools altogether. But this in turn implies that we have selected at least two blocks for which there are no corresponding top and bottom tools. Hence, we can unselect these tools without reducing the number of jobs in the batch, which brings us again in the situation where the solution contains only $(C - 1)$ blocks.

This leaves us with the case in which we have selected C blocks. In this case, we can select $2C$ top and bottom tools altogether. We know however, that there is no benefit in selecting top and bottom tools whose corresponding block is not selected. Hence, given a selection of C blocks, we may assume that the remaining tools in the solution are the corresponding top and bottom tools.

Now that we have imposed some structure on the sets of tools that are selected, let us check the implications for the number of jobs in the corresponding batch. Consider a solution S to I, i.e. a set of selected tools $t_{i_1}, \ldots, t_{i_C}$, with value $s(I)$ and let p_1 to p_k be the jobs of I that can be performed using $t_{i_1}, \ldots, t_{i_C}$. We construct a solution to I' with value $s'(I')$ by selecting the top tools $t'_{i_1}, \ldots, t'_{i_C}$ as well as the corresponding blocks and bottom tools. It can be seen as follows that $s'(I') = s(I)^2$. Solution S' allows exactly the jobs of I' corresponding to all ordered pairs (p_j, p_l) , $j, l = 1, \ldots, k$ to be performed.

Conversely, given a solution S' of value $s'(I')$ to I' which requires (without loss of generality) C blocks and their corresponding top and bottom tools, we obtain a solution to I of value $\sqrt{(s'(I')}$ by selecting tool t_i in I, only if t'_i in I' is selected, $1 \leq i \leq m$. Let $t'_{i_1}, \ldots, t'_{i_C}$, denote the

top tools used in the solution S' to I', and let $p_1, \ldots, p_k$ be the jobs of I that can be performed using $t_{i_1}, \ldots, t_{i_C}$: Solution S' contains exactly those jobs of I' corresponding to all ordered pairs (p_j, p_l) for which p_j and p_l are in the batch corresponding to S. Hence, $k = \sqrt{s'(I')}$. ∎

In view of Theorem 5, we can exclude polynomial approximation algorithms with a constant worst case ratio if we are able to show that there is some constant $\epsilon > 1$ such that a polynomial time ϵ approximation algorithm cannot exist unless P = NP. Results in this spirit might be obtained using the connection between interactive proofs and approximation algorithms established by Feige et al. [1996]. We were able to derive such a result for the following generalization of batch selection, which we refer to as capacitated batch selection. The capacitated batch selection problem is identical to batch slection except for the introduction of so called tool classes. Every tool is in exactly one of these classes, and for each of these classes an upperbound is given on the number of tools in the class that may be contained in the tool magazine at the same time. (Indeed the batch selection problem arises when all tools are of the same class, and the magazine may be completely filled with tools of this class.) It is not hard to extend Theorem 5 to capacitated batch selection. Further, the problem maximum system of representatives (Bellare [1992]) can be viewed as a special case of capacitated batch selection. Bellare [1992] shows that there is a constant $\epsilon > 1$ such that a polynomial time ϵ approximation algorithm for maximum system of representatives implies $P = NP$. We conclude that for capacitated batch selection (nor for maximum system of representatives) polynomial time approximation algorithm with constant worst case ratio can exist, unless P = NP. We ommit a more formal prove of this statement, since we think these problems to be of limited interest.

The remainder of this section discusses the approximability of job grouping, and its relationship with the approximabilty of tool switching. For the job grouping problem we have the following negative result on its approximability :

Theorem 6 For any $d < \frac{1}{4}$, the job grouping problem cannot be approximated within a factor of $d \log n$ in polynomial time unless NP is contained in $DTIME[n^{poly \log n}]$, even if $C = m - 1$.

Proof. Lund & Yannakakis [1993] prove that for any $d < \frac{1}{4}$, set covering can not be approximated within a factor of $d \log k$ in polynomial time unless NP is contained in $DTIME[n^{poly \log n}]$, where k is the number of rows of the covering problem. Hence we prove simply by giving a mapping from set covering to the special case of job grouping where $C = m - 1$ and $n = k$. Notice that in such instances the batch selection problem can be solved trivially, so that finding a minimal cover, or partition, indeed poses the only difficulty.

Let $s_1, \ldots, s_p$ be the subsets in the set covering instance, and let $e_1, \ldots, e_q$ be the elements of its ground set. Then the covering matrix B has $b_{ij} = 1$ if s_j contains e_i and zero otherwise. The following construction of a tool job matrix A is due to Crama & Oerlemans [1992]. We set $a_{ij} = 1 - b_{ji}$ for all $1 \le i \le p, 1 \le j \le q$. Setting $m = p$, and $C = p - 1$ yields, together with A an instance of job grouping. One checks that if some subset s_i contains the two elements e_k and e_l, then both columns A_k and A_l have a zero in row i. Since the tool magazine $C = m - 1$ this means that jobs j_k and j_l may be in a same batch. More generally, every subset in the set covering problem corresponds to a batch in the job grouping problem, and every maximal batch is a subset in the set covering problem. ∎

14

We close this section by discussing the relationship between job grouping and tool switching. Notice again that both problems require the same data. Now let $v_{TS}(S)$ be the value of solution S to the tool switching problem. The same switches imply a solution S' for job grouping of value $v_{JG}(S') \leq v_{TS}(S)$. Conversely, let $v_{JG}(S')$ be the value of any solution S' to job grouping. Since between each two groups we can have at most C switches, there is a solution S to tool switching of value $v_{TS}(S) \leq C \times v_{JG}(S')$. Thus we also have that :

$$\frac{1}{v_{TS}(OPT_{TS})} \leq \frac{1}{v_{JG}(OPT_{JG})} \leq \frac{C}{v_{TS}(OPT_{TS})}.$$

If we have an approximation algorithm H with worst case ratio r_1 for job grouping giving solution s', then we can construct a solution S for tool switching, and it holds that

$$r_1 \geq \frac{v_{JG}(S')}{v_{JG}(OPT_{JG})} \geq \frac{v_{JG}(S')}{v_{TS}(OPT_{TS})} \geq \frac{v_{TS}(S)/C}{v_{TS}(OPT_{TS})}.$$

Thus, H yields a Cr_1 approximation algorithm for tool switching.

If we have an approximation algorithm H' for tool switching with worst case ratio r_2, giving solution S, then we can construct a solution S' of job grouping so that,

$$r_2 \geq \frac{v_{TS}(S)}{v_{TS}(OPT_{TS})} \geq \frac{v_{JG}(S')}{v_{TS}(OPT_{TS})} \geq \frac{v_{JG}(S')/C}{v_{JG}(OPT_{JG})}.$$

Thus H' yields an approximation algorithm with worst case ratio Cr_2 for job grouping.

5 Further research

In combination, the results of Sections 3 and 4 still leave a rather big gap between the ratio the best approximation algorithms achieve and what is likely to be unattainable. We conjecture that in order to close this gap both stronger negative results as well as better approximation algorithms are required. As yet, we have not investigated whether the bounds we derive on the worst case ratios of the approximation algorithms discussed in Section 3 are tight, nor have we investigated methods that enable tight worst case analysis. Such methods too, are required to close the aforementioned gap, and appear to be an interesting direction for further research.

The negative results of Section 3 are expressed in constants and n the number of jobs, in which case a superpolynomial bound on the worst ratio cannot be obtained. However, we have seen that, expressed in terms of the tool magazine C, superpolynomial worst case ratios are possible. It may well be the case that there is some $\epsilon > 0$, such that unless $P = NP$ approximation algorithms with worst case ratio smaller than or equal to $2^{C^{\epsilon}}$ cannot exist for both the job grouping and the batch selection problem. The possibility to achieve superpolynomial worst case results hopefully provides an extra challenge in investigating this possibility.

On the positive side, Crama & van de Klundert [1994] show that iteratively generating batches with an α approximation algorithm for the batch selection problem until all jobs are in some batch, yields an approximation algorithm for job grouping with approximation ratio of $O(\alpha \log(n/\alpha))$.

15

In view of this result, observe that a strengthening of the negative result regarding job grouping has its implication for what is likely to be attainable for batch selection. For example, if one could show that there is some ϵ such that there does not exist a polynomial approximation algorithm with worst case ratio n^ϵ for job grouping, unless P = NP, then there could not exist a polynomial approximation algorithm for batch selection with worst case ratio strictly smaller than n^ϵ unless P = NP. Lund & Yannakakis [1993] have obtained results for problems that are related in a similar fashion. For example, they were able to show that Graph Coloring cannot be approximated within a subpolynomial ratio, as is the case for its generating subproblem Independent Set (Arora et al. [1993]).

Although some of the results we present are rather strong, there appears to be room for even stronger results addressing the approximability of tool management problems. Not only are these problems among the more important scheduling problems in contemporary manufacturing, there are also interesting links with fundamental combinatorial problems. Hopefully, the research reported in this paper and the open problems we have identified incites further research on the approximability of tool management problems and their relationship with other combinatorial problems.

6   Literature

Arora, S., Lund, C., Motwani, R. Sudan, M., Szegedy, M., 1992, On the intractability of approximation problems, Early draft, AT&T Bell Labs, NJ.

Bellare, M., 1992, Interactive proofs and approximation, Research Report, IBM Research Division, T.J. Watson Research Center, Yorktown Heights, NY.

Blazewicz, J., Finke, G., 1994, Scheduling with resource management in manufacturing systems European Journal of Operational Research, Vol. 76, pp. 1-14.

Chaillou, P., Hansen, P., Mahieu Y., 1989, Best network flow bounds for the quadratic knapsack problem, In: Combinatorial Optimization, B. Simeone (ed.) Lecture notes in Mathematics, Vol 1403. Springer Verlag, Berlin, pp. 225-235.

Crama, Y., 1995, Combinatorial models for production scheduling in automated manufacturing systems, 14th European Conference on Operational Research, Semi-plenary Papers, pp. 237-259.

Crama, Y., Klundert, J. J. van de, 1994, Approximation algorithms for integer covering problems via greedy column generation, RAIRO-Operations Research 28, pp. 283-302.

Crama, Y., Kolen, A.W.J., Oerlemans A.G., Spieksma F.C.R., 1994, Minimizing the number of tool switches on a flexible machine, International Journal of Flexible Manufacturing Systems 6, pp. 33-54.

Crama, Y., Mazzola, J.B., 1995, Valid inequalities and facets for a hypergraph model of the nonlinear knapsack and FMS part-selection problems, to appear in Annals of Operations Research.!!!!!!!!!!!

Crama, Y., Oerlemans, A.G., 1994, A column generation approach to job grouping for flexible manufacturing systems, European Journal of Operation Research 78, 58-80.

Dietrich, B.L., Lee, J., Lee, Y.S., 1993, Order selection on a single machine with high set up costs, Annals of Operations Research 43, pp. 379-396.

Feige, U., Goldwasser, S., Lovász, L., Safra, S., Szegedy, M., 1996, Interactive proofs and the hardness of approximating cliques, Journal of the ACM, Vol 43., No 2, pp 268-292.

Gallo, G., Hammer, P.L., Simeone, B., 1980, Quadratic knapsack problems, Mathematical Programming Studies 12, pp. 132-149.

Garey, M.R., Johnson, D.S., 1979, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, New York, New York.

Goldschmidt, O., Hochbaum, D.S., Yu, G., 1992, Component assembly in the semiconductor industry : a study of covering in graphs and hypergraphs, Technical report ORP92-5, The University of Texas at Austin.

Goldschmidt, O., Nehme, D, Yu, G., 1993, On a generalization of the knapsack problem with applications to flexible manufacturing systems and database partitioning, Naval research Logistics!!!!!!!!!!!

Gray, A.E., Seidmann, A., Stecke, K.E., 1993, A synthesis of decision models for tool management in automated manufacturing, Management Science, Vol. 39, pp. 549-567.

Hwan, S.S., Shogan, A.W., 1989, Modelling and solving an FMS part selection problem, International Journal of Productions Research, Vol. 27, No. 8, pp. 1349-1366.

Johnson, E.L., Mehrotra, A., Nemhauser, G.L., 1992, Min-cut clustering, Mathematical programming Vol. 62, pp 1133-151.

Kortsarz, G., Peleg, D., 1993, On choosing dense subgraphs. Extended abstract, Dept. of Applied. Math. and Comp. Science., The Weizmann Institute, Rehovot, Israel.

Lund, C., Yannakakis, M., 1993, On the hardness of approximating minimization problems, Proc 25th ACM Symposium on the Theory of Computing, pp. 286-295.

Mamer, J.W., Shogan, A.W., 1987, A constrained capital budgeting problem with application to repair kit selection, Management Science, Vol. 33, pp 800-806.

Privault, C., Finke, G., 1993, Tool management on NC machines, Proc. Int. Conf. on Industr. Engineering and Prod. Management, Mons Belgium, pp. 667-676.

Rajagopalan, S., 1985., Scheduling problems in flexible manufacturing systems, Research Paper,

Graduate School of Industrial Administration Carnegie-Mellon University, Pittsburgh, PA.

Rajagopalan, S., 1986, Formulation and heuristics solutions for parts grouping and tool loading in flexible manufacturing. In Stecke K.E. and Suri R. eds. Proceedings of the Second ORSA TIMS Conference on Flexible Manufacturing Systems. Elsevier Science Publishing B.V., Amsterdam, pp. 311-320.

Stecke, K.E., 1983, Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems. Management Science Vol. 29, No. 3, pp. 273-288.

Tang, C.S., Denardo, E.V., 1988a, Models arising from a flexible manufacturing machine, part I: Minimization of the number of tool switches, Operations Research, Vol. 36, No. 5, pp. 767-777.

Tang, C.S., Denardo, E.V., 1988b, Models arising from a flexible manufacturing machine, part II: Minimization of the number of switches instants, Operations Research, Vol. 36, No. 5, pp. 778-784.

Whitney, C.K., Gaul T.S., 1985, Sequential decision procedures for batching and balancing in FMSs. Annals of Operations Research 3, pp. 301-316.