Available online at www.sciencedirect.com
ScienceDirect
Robotics and Computer-Integrated Manufacturing 23 (2007) 645–649
Robotics and Computer-Integrated Manufacturing
www.elsevier.com/locate/rcim

An exact solution to the TLP problem in an NC machine

G. Ghiani, A. Grieco, E. Guerriero*

Dip. di Ingegneria dell'Innovazione, Università Degli Studi di Lecce, Via per Arnesano, Corpo O II Piano, 73100 Lecce, Italy

Abstract

This paper considers a job sequencing problem for a single numerical controlled machining center. It is assumed that all the considered jobs must be processed on a single machine provided with a tool magazine with C positions, that no job requires more than C tools to be completely machined and that the tools may be loaded and unloaded from the tool magazine only when the machining operations for each job are completed. The decisional problem is referred to as the tool loading problem (TLP) and it determines the jobs machining sequence as well as the tools to load in the machine tool magazine before the machining operations on each job may start. In industrial cases where the tool switching time is both significant relative to job processing time and proportional to the number of tool switches, the performance criterion is the minimization of the number of tool switches. This paper demonstrates that the TLP is a symmetric sequencing problem. The authors enrich a branch-and-bound algorithm proposed in literature for the TLP with the new symmetric formulation. Computational experiments show the significant improvement obtained by the novel symmetric formulation of the TLP.

Keywords: Tool management; Tool switching problem; FMS; Branch and bound algorithms

1. Introduction

This paper addresses tool loading problem (TLP) encountered in the manufacturing of products by metal working with numerical controlled machines. A set of parts or batches of different types $J = \{1, \ldots, N\}$, each requiring a defined set of tools T_i $(i = 1, \ldots, M)$, are to be produced on a single numerical controlled (NC) machining center. The machining center is provided by a tool magazine with C slots. Since the tool loading and unloading are forbidden during the working operations, it is assumed that no job (or partial sequence of the job machining part program with technological precedence constraints) requires more than C different tools to be machined (i.e. $\max_i(T_i) \leqslant C$).

The total number of tools M required to complete the overall set of parts exceeds the tool magazine capacity, so that tool switches between the machining of following parts are usually necessary. A tool switch occurs when a tool is removed from the machine tool magazine and a different tool is inserted on the machine tool magazine. Each tool can be placed in any slot of the magazine and it is assumed that it requires only one slot. Before processing a part, all the tools required by the part under process must be loaded into the tool magazine. Since the time required for the tool switches can be significant relative to processing time, it is desirable to limit the amount of time associated with tool switches. The TLP consists of determining a sequence of parts and the corresponding set of tools loaded in the magazine at any time in order to minimize the overall makespan.

The TLP also arises in computer memory management, when pages (tools) have to be transferred from a slow memory to a fast memory (tool magazine) in order to execute computational tasks [1–3]. Another industrial application is related to automated printed circuit board (PCB) loading machines. These machines automatically place electrical components onto PCBs, each one requiring different components (tools) to be manufactured. Setting up the automated PCB loading machine with a new set of components for the next PCB requires huge amount of production time; moreover, the mix of components for different PCB types can be partially or completely different. Consequently, the optimization objective is to minimize the number of component swaps (i.e. the number of tools switches) [4].

*Corresponding author. Tel.: +39 0832297789.
E-mail address: emanuela.guerriero@unile.it (E. Guerriero).

0736-5845/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.rcim.2007.02.011

The NP-hardness of TLP has been proved in Refs. [5,6] for $C\geqslant 2$. Tang and Denardo [7] demonstrated that the TLP can be solved in polynomial time by applying the *keep tool needed soonest* (KTNS) policy. This policy has the following properties: (i) at any instant, no tool is inserted unless it is required by the next job and (ii) if a tool must be inserted, the tools that are kept (not removed) are those needed the soonest.

Several heuristics have been proposed in literature for the TLP [3,8]. Nevertheless, only two main contributions have been formulated for exact solution approaches [7,9]. In Ref. [7], the authors proposed an exact solution approach, based on an integer linear programming (ILP) formulation of the TLP, which yields poor result: the linear relaxation value, when no job has been fixed, is always equal to zero. Laporte et al. [9] proposed two exact algorithms for the TLP. The first is based on a LP-based branch-and-cut algorithm, which solves a ILP formulation providing strictly better linear programming lower bounds than the formulation of Tang and Denardo [7]. The second is a direct branch-and-bound approach that does not use LP. To test the efficiency of the proposed algorithms, a series of problem sets with $8\leqslant N\leqslant 25$, $15\leqslant M\leqslant 25$ and $4\leqslant C\leqslant 20$ were investigated. The branch-and-cut algorithm was capable of solving instances containing up to nine jobs, but had a very low success rate $N\geqslant 10$. The branch-and-bound algorithm solved instances containing 15, 20 and 25 jobs within relatively short computing times.

The contribution of this article is to demonstrate that the TLP is a symmetric sequencing problem. The paper is organized as follows. In Section 2, the symmetry property is stated and proved. In Section 3, we reformulate the branch-and-bound algorithm proposed in [9], so that the novel symmetric formulation of TLP is exploited. Sections 4 and 5 have been devoted to computational results and conclusion.

2. The symmetry property

The moment in time after processing the nth job, but before any tools are switched is called instant n. For a fixed job schedule $(j_1, \ldots, j_N)$, let us renumber the jobs so that job j_n is the nth job in the sequence. In the following, we will refer to $(j_1, \ldots, j_N)^{\mathrm{R}}$ as the job schedule $(j'_1, \ldots, j'_n)$ such that $j'_1 = j'_n, j'_2 = j'_{N-1}, \ldots, J'_N = J_1$. Finally, let W_n be the tool set that describes the tools on the machine at instant n.

Property 1. *Let f_1^* and f_2^* be the minimum number of tool switches for the job sequences $(j_1, \ldots, j_N)$ and $(j_1, \ldots, j_N)^{\mathrm{R}}$, respectively. It results that $f_1^* = f_2^*$.*

Proof. Let us suppose to apply the KTNS policy to $(j_1, \ldots, j_N)$. As stated above, the KTNS procedure is optimal and it determines one machine configuration W_n for each instant n. The total number of optimal tool switches f_1^* can be computed as

$$\sum_{n=2}^{N} \left| W_n \backslash W_{n-1} \right| = f_1^*. \tag{1}$$

In the same way, let W_n^{R} be the tool set representing the machine configuration for instant n, obtained applying the KTNS policy to the sequence $(j_1, \ldots, j_N)^{\mathrm{R}}$.

The total number of optimal tool switches f_2^* can be computed as

$$\sum_{n=2}^{N} \left| W_n^{\mathrm{R}} \backslash W_{n-1}^{\mathrm{R}} \right| = f_2^*. \tag{2}$$

Let us suppose that $f_1^* < f_2^*$. The sequence of machine configurations $(W_N, W_{N-1}, \ldots, W_1)$ is feasible to process the job sequence $(j_1, \ldots, j_N)^{\mathrm{R}}$, i.e. each W_n is feasible to process the job j_n with $n = 1, \ldots, N$.

We assumed that the total number of tools M exceeds the tool magazine capacity C. Consequently, it results that $|W_n| = C$ for each $n = 1, \ldots, N$ and that $|W_n \backslash W_{n-1}| = |W_{n-1} \backslash W_n|$. In this way, one can state that Eq. (1) represents the total number of tool switches associated to the sequence of machine configurations $(W_N, W_{N-1}, \ldots, W_1)$ and f_1^* is the corresponding total number of tool switches (see (3)),

$$\sum_{n=2}^{N} \left| W_{n-1} \backslash W_n \right| = \sum_{n=2}^{N} \left| W_n / W_{n-1} \right| = f_1^*. \tag{3}$$

Therefore, we have determined a sequence of machine configurations, i.e. $(W_N, W_{N-1}, \ldots, W_1)$, feasible for the job sequence $(j_1, \ldots, j_N)^{\mathrm{R}}$, which requires a lower number of tool switches, $f_1^* < f_2^*$. Consequently, supposing that $f_1^* < f_2^*$ disclaims the optimality of the KTNS policy.

In the same way, it can be proved that if we suppose that $f_2^* < f_1^*$ the optimality of the KTNS policy applied to $(j_1, \ldots, j_N)$ should be disclaimed. □

3. Branch-and-bound algorithm

We exploited the symmetry property of the TLP for developing an exact solution algorithm. The job sequencing component of the TLP corresponds to find a tour as long as it is introduced a dummy job, denoted by 0 and representing the start and the end of operations. Therefore, we can represent the TLP as a special case of TSP, where due to the switching tools component of the TLP, the arc cost functions are nonlinear. An instance of the TSP is given by a directed graph $G(V,A)$, where V is the set of nodes and A is the set of arcs. The binary variables x_{ij} are equal to one if and only if job i is immediately followed by job j $(i,j\in V = J\cup\{0\})$. If each tool set is such that $|T_i| = C \forall i\in J$, then any job j immediately following job i will generate $|T_j \backslash T_i|$ tool switches and the TLP is equivalent to TSP. In all other cases, the sequencing cost of nodes i and j (c_{ij}) should be computed taking into account all nodes foregoing node j. Based on these considerations, Laporte et al. [9] proposed a TSP branch and bound tailored for the TLP. The algorithm proposed in this paper follows the lower bounding scheme and the incumbent initialization described in Ref. [9], but the branching rule is defined in order to take into account the formulation of the TLP as a symmetric sequencing problem.

In the following sub-sections, the upgraded branch-and-bound algorithm is described.

The initial upper bound is obtained by the application of the following greedy heuristic. The first job is the one requiring the largest number of tools. To break ties, the job with the most frequently used tools is selected. At each iteration, the job with the largest number of tools in common with the last job in the current sequence is scheduled; to break ties, the selection criteria minimizes the total number of tools used by the last job in the current sequence and the job being scheduled.

While more sophisticated heuristics are available for the TLP, Laporte et al. in [9] observed that there is any advantage in refining this simple heuristic. Indeed, the computational time is not influenced by the initial upper bound: there was no meaningful difference between the performances obtained initiating the search tree with their heuristic solution or with the optimal one.

As observed in [5,10], a lower bound l_{ij} on the number of tool switches needed when j immediately follows i, can be defined as follows:

$$l_{ij} = \max\{0, |T_i \cup T_j| - C\}.$$

Given a partial sequence of jobs $(j_1, \ldots, j_p)$, let Q be defined as the set $J \backslash (j_1, \ldots, j_p)$.

As reported in [9], a lower bound on the optimal value can be obtained by applying the KTNS policy for the partial sequence, plus the maximum of the two lower bounds, referred to as LB1 and LB2 and illustrated in the following.

The lower bound LB1 is computed as the number of tools required by the last and forthcoming jobs, minus C. This lower bound has been formalized as

$$\left|\bigcup_{j\in Q} T_j \backslash T_{j_p}\right| - (C - |T_{j_p}|) = \left|T_{j_p} \cup \left(\bigcup_{j\in Q} T_j\right)\right| - |T_{j_p}| - C + |T_{j_p}|$$
$$= \left|T_{j_p} \cup \left(\bigcup_{j\in Q} T_j\right)\right| - C.$$

Indeed, a valid lower bound is the number of tools necessary for the forthcoming jobs and not necessary for the last job in the partial sequence, minus the number of free slots in the magazine.

The lower bound LB2 is computed as the minimum cost of an j_p-spanning defined on Q, plus the least cost edge connecting it to j_p. This lower bound exploits that jobs in $\{j_p\} \cup Q$ form a Hamiltonian chain and where cost edge are valued as l_{ij}, which represent a lower bound on the sequencing cost c_{ij}.

The computational complexity of both lower bounds is polynomial in N and M. Indeed, the first lower bound can be computed in $O(M)$ time, while the second takes $O(N^2 \log N)$ time if Kruskal's algorithm [11] is used to compute the spanning tree. Since there is no dominance between the two lower bounds, the maximum is chosen.

The branching rule is to consider jobs in the order in which they appear in the greedy algorithm. In order to take into account the symmetry property, we modify this branching rule, so that the algorithm considers a job sequence once, i.e. for the symmetry property reversing the job sequence the minimum number of tool switches does not change. In particular, each branch-and-bound node is characterized by both a partial sequence and a set of jobs L that can be used as ending jobs. If the next job selected for branching is an ending job, then it is removed from the set L. If the set L is empty, the node is fathomed.

The set L is initialized when the first job of the partial sequence is fixed. In particular, if the first job fixed is the hth job in the greedy algorithm, then the set L consists of all jobs occupying all positions from (h+1) to N in the greedy sequence. For example if the greedy sequence is (5-3-4-1-2), given the three partial sequences (4), (1,3,4), (3,5) the corresponding L sets are (1,2), (2) and (4,1,2), respectively.

4. Computational results

The algorithm just described as well as the branch-and-bound algorithm proposed in [9] were coded in C++ and run on a PC with a 800 MHz Pentium processor and a 256 MB RAM. We tested the algorithms on the eight instance types solved in [9,12] with a time limit of 3600 s. We obtained that the two branch-and-bound algorithms solved optimally tests requiring up to 10 jobs ($N = 10$) and up to 10 tools ($M = 10$). In the experimental campaign, it is assumed that the total number of jobs and the total number of tools required to machine all the jobs is equal to 10. Moreover for each instance, each job (or a partial sequence of the job machining part program with technological precedence constraints) requires a number of tools ranging from 2 to 4.

Test parameters and computational results are reported in Tables 1 and 2 under the following headings:

- T_avg: average number of tools for each test.
- Tool magazine saturation: the percentage tool magazine slots occupied by jobs on average.
- C: tool magazine capacity.
- Min: minimum number of tools required for a job.
- Max: maximum number of tools required for a job.
- Nodes: number of nodes in the search tree.
- Seconds: computation time in seconds.
- Case A: results referred to Branch-and-Bound Algorithm based on symmetric formulation.
- Case B: results referred to the original Branch-and-Bound Algorithm proposed in [9].
- Gain: the percentage value of computational time gain obtained considering the symmetry formulation.

Our results clearly show the advantage of exploiting the TLP symmetric formulation, when the problem instances are close to TSP instances (i.e. the higher is the tool magazine saturation, the closer to TSP the TLP instance is). Ten sets of 10 jobs have been considered in order to define four different classes of instances: for each job set, four TLP instances have been determined considering the tool magazine capacity ranging from 4 to 7. The average tool magazine saturation in the first class of problem is equal to 74% while it is equal to 42% in the last class. The first class tool magazine saturation value is representative of actual industrial applications while the remaining class values are meaningful mostly for a computational performance evaluation.

In Fig. 1, the box plot representation is reported for the four different problem classes related with the tool magazine capacity considered. In the first considered class of instances with the tool magazine capacity equal to 4, the median of the percentage gain is equal to 15% while the first quartile is equal to 7%. In the second and third classes, the gain is ever more significant while in the last class of problem, the two approaches are equivalent.

It should be noted in actual applications job clustering is often assumed. So that the total number of different jobs considered in the experimental cases are representative of actual applications. Therefore, since such job clustering hypothesis has been made in our computational campaign, our tests may be more difficult to solve than what is observed.

Table 1
Job sets parameters

Job sets	T_avg	Tool magazine saturation (%)			
		$C=4$	$C=5$	$C=6$	$C=7$
1	3.2	80	64	53	46
2	2.6	65	52	43	37
3	2.8	70	56	47	40
4	2.9	73	58	48	41
5	3.2	80	64	53	46
6	3.1	78	62	52	44
7	2.7	68	54	45	39
8	2.7	68	54	45	39
9	3.3	83	66	55	47
10	3	75	60	50	43

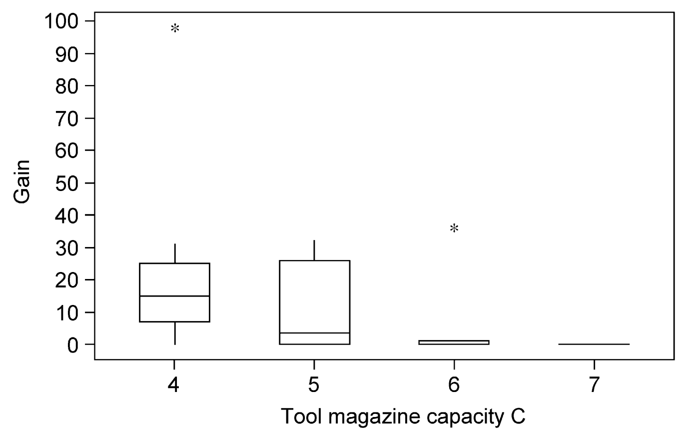

Fig. 1. Box plot representation.

Table 2
Computational results

Test	C	Case A		Case B		Gain (%)	Test	C	Case A		Case B		Gain (%)
		Nodes	Seconds	Nodes	Seconds				Nodes	Seconds	Nodes	Seconds	
1	4	14,516	110	16,951	124	13	21	6	83	0	83	0	0
2	4	1654	12	1983	14	17	22	6	10	0	10	0	0
3	4	8227	57	17,391	113	98	23	6	10	0	10	0	0
4	4	6157	48	8597	63	31	24	6	45	0	45	0	0
5	4	3749	31	4737	38	23	25	6	164	1	164	1	0
6	4	13,768	105	17,396	127	21	26	6	9459	81	13,493	110	36
7	4	5387	39	5954	42	8	27	6	238	1	238	1	0
8	4	1062	8	1232	9	13	28	6	60	0	60	0	0
9	4	3346	27	3525	28	4	29	6	169	1	169	1	0
10	4	1091	9	1169	9	0	30	6	56	0	56	0	0
11	5	2125	15	2150	16	7	31	7	53	0	53	0	0
12	5	40	0	40	0	0	32	7	10	0	10	0	0
13	5	10	0	10	0	0	33	7	10	0	10	0	0
14	5	1353	12	1751	15	25	34	7	19	0	19	0	0
15	5	801	7	972	9	29	35	7	10	0	10	0	0
16	5	7356	60	10,108	79	32	36	7	79	0	79	0	0
17	5	5967	46	6616	49	7	37	7	10	0	10	0	0
18	5	376	2	376	2	0	38	7	10	0	10	0	0
19	5	2585	21	2741	21	0	39	7	10	0	10	0	0
20	5	1068	9	1146	9	0	40	7	27	0	27	0	0

5. Conclusions

The symmetry property of the TLP has been proved. The authors enhanced a branch-and-bound algorithm reported in literature in order to take into account the demonstrated symmetry property of the TLP. Computational results indicate a performance increase, when the novel symmetric formulation of TLP is exploited in a branch-and-bound approach. In Ref. [9], it has been demonstrated that for TLP, the branch-and-bound approach is superior to mathematical programming approach. Our future research will be devoted to evaluate a mathematical programming solution approach based on the novel symmetric formulation of the TLP.

Acknowledgment

The authors thank Anna Arigliano for her excellent contribution to the implementation and testing phases.

References

[1] Belady LA. A study of replacement algorithms for virtual storage computers. IBM Syst J 1966;5:70–101.
[2] McGeoch LA, Sleator DD. A strongly competitive randomized paging algorithm. Algorithmica 1991;6:816–25.
[3] Privault C, Finke G. *k*-server problems with bulk requests: an application to tool switching in manufacturing. Ann Oper Res 2000;96:255–69.
[4] Jain SE, Williams PF. Set-up optimization for flexible manufacturing systems. US Patent 5,155,679, 1992.
[5] Oerlemans AG. Production planning for flexible manufacturing systems. PhD dissertation. University of Limburg, Maastricht, 1992.
[6] Crama Y, Kolen AWJ, Oerlemans AG, Spieksma FCR. Minimizing the number of tool switches on a flexible machine. Int J Flexib Manuf Syst 1994;6:33–54.
[7] Tang CS, Denardo V. Models arising from a flexible manufacturing machine. Part I: minimizing the number of tool switches. Oper Res 1988;36:767–77.
[8] Knuutila T, Hirvikorpi M, Johnsson M, Nevalainen O. Grouping PCB assembly jobs with typed component feeder units. Technical report 460. Finland: Turku Centre for Computer Science; 2002.
[9] Laporte G, Salazar JJ, Semet F. Exact algorithms for the job sequencing and tool switching problem. IIE Trans 2004;36(1):37–145.
[10] Avci S, Akturk MS. Tool magazine arrangement and operations sequencing on CNC machines. Comput Oper Res 1996;23:1069–81.
[11] Kruskal JB. On the shortest spanning subtree of a graph and the travelling salesman problem. Proc Am Math Soc 1956;7:48–50.
[12] Hertz A, Laporte G, Mittaz M, Stecke KE. Heuristics for minimizing tool switches when scheduling part types on a flexible machine. IIE Trans 1998;30:689–94.