

RESEARCH SUPPORT
UNIVERSITY OF MICHIGAN BUSINESS SCHOOL

NOVEMBER 1997

**HEURISTICS FOR MINIMIZING TOOL
SWITCHES ON A FLEXIBLE MACHINE**

WORKING PAPER #9709-02R

BY

**ALAIN HERTZ
GILBERT LAPORTE
MICHEL MITTAZ
KATHRYN E. STECKE**

Heuristics for Minimizing Tool Switches when Scheduling Part Types on a Flexible Machine

by

Alain Hertz¹
Gilbert Laporte²
Michel Mittaz¹
Kathryn E. Stecke³

June, 1997

Revised: November, 1997

¹Département de Mathématiques, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland.

²GERAD, École des Hautes Études Commerciales, 3000, chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7.

³School of Business Administration, University of Michigan, Ann Arbor, MI 48109-1234, USA.

Abstract

This article considers a tool loading problem whose objective is to minimize the number of tool switches over time in order to process several parts on a flexible machine. New heuristics are presented and compared. Some of these are shown to be superior to existing methods.

Key-words: Tool loading problem, flexible manufacturing, heuristics.



1 Introduction

We consider the following *Tool Loading Problem* (TLP) encountered in flexible manufacturing. A series of n parts of different types, each requiring a particular set of tools of different sizes are to be produced on a single flexible machine. The tools are to be loaded in a magazine containing c slots. Each part type requires at most c tools, but the total number of tools required for all part types typically exceeds c , so that tool switches between part types are usually necessary. Each tool can be placed in any slot of the magazine. Before processing a part, all tools required by that part must be installed. Since the time required for 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 sets of tools loaded in the magazine at any time in order to minimize the completion time of all parts.

Here we examine a TLP over time for a single flexible machine. The solution of this problem is a first step in solving the more general flexible manufacturing problem of selecting part types over time. Scheduling flexible manufacturing operations over time typically involves selecting overlapping batches of part types. When the production requirements of a part type are completed, its tools can be taken out of the tool magazines and the tools for new part types entering can replace these (see Stecke and Kim (1988, 1991)). It is ideal to sequence the processing of the different part types so as to minimize reloading tools that had previously been taken out. The TLP addressed here is a first step towards addressing the more general problem. Since the processing time of each part is sequence independent, we are only concerned with the time associated with tool switches. Some authors (e.g., Tang and Denardo (1988)) minimize the number of *switching instants*, i.e., the number of times at which one or more tools must be changed in the magazine in order to process the next part. Alternatively, one can minimize the number of *tool switches*, i.e., the total number of tools changes during the whole process (see, e.g., Bard (1988), Tang and Denardo (1988), Kiran and Krason (1988), Oerlemans (1992), Gray, Seidmann and Stecke (1993), Crama, Kolen, Oerlemans and Spijksma (1994), Follonier (1994), Sodhi, Askin and Sen (1994), Hertz and Widmer (1996), and Avci and Akturk (1996)). This paper deals with the latter objective.

As shown by Crama *et al.* (1994), the TLP is NP-hard by reduction from the *Matrix Permutation Problem* (Garey and Johnson, 1979). However, once the part sequence is known, optimally loading the tools in the magazine is easily accomplished by application of a “Keep Tool Needed Soonest (KTNS)” policy (Bard (1988), and Tang and Denardo (1988)). This policy states that when tool changes are necessary, those tools which are required the soonest for an upcoming part should be the first to be kept in the magazine.

When each part requires exactly c tools the TLP reduces to a *Traveling Salesman Problem* (TSP) with distances $d(i, j)$, where

$$d(i, j) = \frac{1}{2} \left(|T_i \cup T_j| - |T_i \cap T_j| \right) = c - |T_i \cap T_j|,$$

and T_i is the set of tools required by part i . Since a TSP solution is a tour, not a path, one can simply introduce a dummy job 0 with $d(0, j) = d(j, 0) = 0$ for all j (e.g., Crama *et al.* (1994)). This procedure disregards the cost of loading the magazine for the first part and of unloading it after the last part. Distances may of course be defined differently if the situation warrants it. In general not all parts require c tools, but standard TSP algorithms can still be applied to provide a heuristic solution to the problem, as suggested by Tang and Denardo (1988) and Crama *et al.* (1994), for example. However, all known algorithms based on this approach are myopic in the sense that they account for interactions of two parts at a time without a global view of the entire solution. As we show in this paper, this can partly be remedied by defining more adequate distances and by designing a more holistic TSP-based heuristic. We elaborate on these two points in Sections 2 and 3, respectively. In Section 4, we present computational results showing the relative efficiency of the proposed approach. The conclusion follows in Section 5.

2 Distance definitions

We have considered several ways of defining a “tooling” distance between two parts. Here we restrict our attention to the five most interesting definitions. The first two distances are simply

$$d_1(i, j) = c - |T_i \cap T_j|$$

and

$$d_2(i, j) = |T_i \cup T_j| - |T_i \cap T_j|.$$

Both are equivalent to $d(i, j)$ when $|T_i| = c$ for all i . These two distances are natural in the sense that they take a larger value when part types i and j have few tools in common. The first is an upper bound on the number of tool switches between i and j . The next distance

$$d_3(i, j) = \max \{0, |T_i \cup T_j| - c\}$$

used by Crama *et al.* (1994) represents a lower bound on the number of tool switches between i and j . This is stronger than the lower bound $|T_i \cup T_j| - c$ presented by Tang and Denardo (1988) since it is never negative. Note that if j immediately follows i , the value $|T_j \setminus T_i|$ (and not $|T_i \setminus T_j|$ as suggested by Crama *et al.* 1994) is a valid upper bound on the number of switches from i to j , but it is not symmetric.

The previous three distances consider only the interaction between two parts and do not take into account the $c - |T_i|$ tools present in the magazine when going from i to j nor those required by parts following j . We will present two new distance metrics that improve upon d_3 and d_2 by giving different weights to their terms. The first of these distances improves on d_3 by subtracting a quantity smaller than c if the tools required by i or j are not likely to be required before i or after j , and a larger quantity if they are more likely to be required before i or after j . For this, we compute $\lambda_k(i, j)$ as the number of parts, apart from i and j , requiring tool $k \in T_i \cup T_j$, and $\Lambda(i, j) = \sum_{k \in T_i \cup T_j} \lambda_k(i, j)$. The larger the $\Lambda(i, j)$, the more likely it is that the tools of $|T_i \cup T_j|$ are required for other parts. Observe that $\Lambda(i, j) \leq (n - 2)|T_i \cup T_j|$. Hence we define

$$d_4(i, j) = \max \left\{ 0, |T_i \cup T_j| - \left[\theta \frac{\Lambda(i, j)}{(n - 2) |T_i \cup T_j|} \right] c \right\},$$

where θ is a parameter in $[0, 1]$. Thus d_4 subtracts from $|T_i \cup T_j|$ a quantity in $[0, c]$, which is larger if the tools of $|T_i \cup T_j|$ are frequently utilized.

In the same spirit, we introduce a variation of d_2 by defining

$$d_5(i, j) = \left(\left[\frac{c + 1}{c} \right] |T_i \cup T_j| - |T_i \cap T_j| \right) \left[\frac{(n - 2) |T_i \cup T_j|}{\max \{ \Lambda(i, j), 0.5 \}} \right].$$

The factor $(c + 1)/c$ can vary between 1 and 2. It gives a larger weight to $|T_i \cup T_j|$ if the size of the magazine is small, i.e., if more tool changes are probable. The second

bracketed factor is similar to that used in d_4 . It is always at least equal to 1 and becomes larger if the tools of $|T_i \cup T_j|$ are seldom utilized. The value 0.5 is used to avoid dividing by 0 when $\Lambda(i, j) = 0$.

3 Algorithms

A natural decomposition strategy for the TLP is to first solve the associated TSP using one of the distances defined in Section 2, and then determine an optimal tool sequence using a KTNS policy. Crama *et al.* have applied this approach. They obtain the best heuristic results using the Golden and Stewart (1985) Farthest Insertion (FI) heuristic with all possible starting parts, combined with d_3 . In heuristic FI, a tour is gradually constructed as follows:

3.1 FI heuristic

Step 1. Consider a starting part i and a part j furthest from i . Construct the current tour (i, j, i) .

Step 2. For each part k not on the tour, compute the shortest distance $\delta(k)$ between k and all parts on the tour. Select part k^* maximizing $\delta(k)$ and insert this part between two consecutive parts on the tour in order to minimize the extra length of the tour. Repeat this step until all parts are on the tour.

3.2 GENI heuristic

A better construction heuristic is GENI, proposed by Gendreau, Hertz and Laporte (1992). This algorithm can be summarized as follows. Starting from three arbitrary parts, GENI inserts at each step a part k not yet on the current tour, between two parts i and j already on the tour and among the p closest neighbors of k . GENI is more than a standard insertion procedure as each insertion is executed simultaneously with a local reoptimization of the tour. Its complexity is $O(np^4 + n^2)$, where n is the number of parts.

A postoptimization phase, called US was developed, based on these generalized insertions. In US, each part is in turn removed from the tour using the reverse GENI operation, and the part is then reinserted in the tour using GENI. The procedure ends when no further improvement can be obtained by removing and reinserting any part. The complexity of GENIUS cannot be determined in terms of n and p as it can be applied as long as the objective function improves. The GENIUS heuristic (Gendreau *et al.* 1992) consists of executing US after GENI. On randomly generated TSP instances and on problems described in the literature, GENIUS has produced highly competitive results. Computation time and solution quality both increase with p . In practice selecting p in the range [3,7] produce good results.

3.3 Heuristics based on the TLP objective

In both FI and GENI, it is necessary to select at each step a part to be inserted in the current tour and its best position in the tour. The determination of the best insertion can be based on one of the distance functions defined in Section 2. We propose an improvement by which the TLP objective is used directly: for each tentative insertion, compute the number of tool switches using a KTNS policy, and perform the insertion yielding the smallest number of tool switches. This principle can be applied to any construction or improvement heuristic. Note that the distance criterion is still used to determine which part to insert in FI and to compute the neighbourhoods in GENI. We point out that Crama *et al.* (1994) have applied a similar idea within the framework of a Nearest Neighbour (NN) heuristic: parts are sequentially added in the last position of the tour according to the TLP objective. They have also developed a 2-opt improvement procedure based on the number of tool switches.

4 Computational results

We now describe the results of extensive tests performed to assess the performance of several algorithms, using the five distances described in Section 2 and data sets possessing different characteristics.

4.1 Data sets

We have produced sixteen types of problem instances as in Crama *et al.* (1994). Each instance type is characterized by the vector of parameters (n, m, \min, \max, c) , where

- n = number of parts;
- m = number of tools;
- \min = lower bound on the number of tools required for any part;
- \max = upper bound on the number of tools required for any part;
- c = tool magazine capacity.

The various instance types generated are described in Table 1. For each type, ten instances were randomly generated as in Crama *et al.*, resulting in a total of 160 instances.

Table 1: Instance types

n	m	\min	\max	c
10	10	2	4	4, 5, 6, 7
15	20	2	6	6, 8,10,12
30	40	5	15	15,17,20,25
40	60	7	20	20,22,25,30

4.2 Algorithms

We have first tested the following four basic heuristics.

- FI1: Successively apply FI using each part as a starting point. Select the shortest tour and apply KTNS to it.
- FI2: Successively apply FI using each part as a starting point and apply KTNS to each of the n tours. Select the solution with the least number of tool switches.

GENI: Apply the GENI heuristic followed by KTNS. In our implementation, we use a neighborhood size $p = 6$ in GENI.

GENIUS: Apply GENIUS (with $p = 6$), followed by KTNS.

In the next five heuristics, the TLP objective is used at each step to guide the search as explained in Section 3.3.

FI*: Successively apply FI by considering each part as a starting point and using the TLP objective to determine the best insertion. Select the best overall solution.

GENI*: Apply GENI (with $p = 6$) using the TLP objective to determine the best insertion.

GENIUS*: Apply GENIUS (with $p = 6$) using the TLP objective to determine the best insertion.

NN*: Apply the Nearest Neighbour heuristic (called “Greedy” in Crama *et al.* 1994) using the TLP objective to select the next part to be added to the current partial solution.

2-opt*: Apply the 2-opt interchange mechanism using the TLP objective.

4.3 Tests

We have run the nine algorithms just described, all programmed in Pascal, on each of the 160 instances generated. For each of the first seven algorithms we successively used the five distances described in Section 2. For NN* and 2-opt*, there is no distance involved. Preliminary tests were performed to determine the best value of θ in d_4 . The value $\theta = 0.25$ seems to be the best and it was used in all subsequent tests.

Computational results are summarized in Table 2. For each algorithm/distance combination, we report two average statistics over the 160 instances:

‰: deviation, in percent, of the value of the TLP objective function (number of tool setups, equal to c plus the number of tool changes) over the best known value;

Sec: computation time in seconds on an SG Indigo machine (100 MHz, IP20 Processor).

Table 2: Comparison of several algorithms and distances

Algorithm	Distance											
	d_1		d_2		d_3		d_4		d_5		None	
	%	Sec	%	Sec	%	Sec	%	Sec	%	Sec	%	Sec
FI1	15.8	4.1	12.2	4.4	25.9	3.9	12.4	4.3	13.1	4.5	-	-
FI2	8.5	5.0	6.5	5.2	17.0	4.5	5.9	5.1	5.7	5.4	-	-
GENI	10.9	0.5	11.6	0.6	25.6	0.3	12.3	0.5	10.4	0.6	-	-
GENIUS	10.3	2.8	10.7	2.7	25.3	1.7	9.5	2.6	8.7	3.8	-	-
FI*	7.7	437.1	3.9	455.9	4.5	460.6	3.1	462.0	3.8	454.4	-	-
GENI*	3.7	187.4	4.3	195.1	6.9	189.4	6.9	187.0	4.4	190.5	-	-
GENIUS*	1.0	1002.7	0.9	1206.1	2.8	1210.4	2.6	1168.0	1.3	1083.2	-	-
NN*	-	-	-	-	-	-	-	-	-	-	5.4	221.2
2-opt*	-	-	-	-	-	-	-	-	-	-	7.8	124.4

These results indicate that there are clearly three classes of algorithms. The first four methods, FI1, FI2, GENI, and GENIUS, are fast, but not the best in terms of solution quality. The three algorithms FI*, GENI*, and GENIUS* are much slower, but produce better solutions. The two algorithms NN* and 2-opt* are both dominated by other strategies. Among the first four algorithms, FI1 is dominated by GENI, and GENIUS for all five distance functions. The fastest algorithm is GENI, and the best in terms of solution value is FI2. Of all distance functions, d_3 is clearly the least interesting and d_5 is usually better than d_1 , d_2 and d_4 . It is interesting to note that Crama *et al.* used the FI1/ d_3 combination in their tests. Within this class of algorithms, FI2/ d_2 , d_4 and d_5 , and GENI/ d_1 , d_2 , d_4 and d_5 are worthwhile combinations. Some algorithms of the second group are dominated by FI2/ d_4 and d_5 . Otherwise, if solution quality is of prime concern, GENIUS* with d_1 or d_2 is probably the best choice.

In order to compare the behavior of the various problem types, we report in Table 3 the % value corresponding to the best version of each algorithm/distance combination for each problem type. It can be seen from this table that running time is directly related to problem size (m and n) and is almost always independent from

the remaining parameters (except for 2-opt* when $n = 30$). Problem difficulty is directly affected by the value of c . Large values of this parameter tend to produce solutions containing fewer tool switches and, in this case, it is likely that several heuristics generate optimal or near-optimal solutions.

Table 3: Comparisons of several algorithms on various types of problem

(n, m, min, max, c)	FI1(d_2)		FI2(d_5)		GENI(d_5)		GENIUS(d_5)		FI*(d_4)		GENI*(d_1)		GENIUS*(d_2)		NN*		2-opt*	
	%	Sec	%	Sec	%	Sec	%	Sec	%	Sec	%	Sec	%	Sec	%	Sec	%	Sec
(10,10,2,4,4)	9.6	0.3	4.8	0.4	8.0	0.1	7.2	0.7	2.4	3.0	2.4	6.1	0.8	34.0	0.8	1.2	8.8	0.5
(10,10,2,4,5)	6.5	0.3	2.8	0.4	8.3	0.1	6.5	0.6	3.7	2.9	1.9	6.2	0.0	34.1	1.8	1.5	7.4	0.5
(10,10,2,4,6)	4.9	0.2	1.0	0.4	4.9	0.1	3.0	0.6	2.0	2.9	2.0	6.2	0.0	33.1	2.0	1.5	4.0	0.4
(10,10,2,4,7)	0.0	0.3	0.0	0.4	1.0	0.1	0.0	0.6	0.0	2.8	0.0	6.1	0.0	33.7	0.0	1.5	0.0	0.4
(15,20,2,6,6)	13.4	0.7	7.4	1.0	10.8	0.2	9.7	1.4	1.9	15.2	4.1	26.1	1.1	118.9	6.7	7.6	11.5	3.4
(15,20,2,6,8)	14.5	0.7	6.4	1.0	9.5	0.2	9.1	1.2	4.1	15.2	5.0	23.8	0.4	121.1	4.5	7.8	8.6	2.6
(15,20,2,6,10)	12.1	0.7	3.5	1.0	8.6	0.2	6.6	1.2	4.0	15.3	2.5	25.1	0.0	113.7	2.5	8.2	7.1	2.5
(15,20,2,6,12)	6.8	0.7	0.5	1.0	3.6	0.2	2.6	1.2	2.6	15.3	1.6	25.3	0.0	110.6	0.0	8.3	0.5	2.4
(30,40,5,15,15)	14.1	4.8	8.0	6.2	11.6	0.8	9.5	4.7	2.2	325.8	4.9	183.4	1.3	1004.4	10.0	158.9	9.7	91.2
(30,40,5,15,17)	16.6	4.8	9.0	6.2	15.0	0.8	13.3	4.7	2.7	326.5	5.4	208.5	1.9	1084.5	10.6	158.9	10.4	83.6
(30,40,5,15,20)	18.9	4.8	9.6	6.2	17.0	0.8	15.0	4.8	5.3	326.2	6.9	192.9	2.4	1120.2	9.1	163.6	12.7	69.6
(30,40,5,15,25)	18.1	4.8	8.8	6.2	17.0	0.8	15.1	4.7	5.0	324.7	5.6	191.2	1.3	950.0	6.3	169.8	10.4	49.1
(40,60,7,20,20)	12.5	10.9	6.1	14.1	10.4	1.4	7.7	8.5	2.2	1459.2	2.8	509.3	1.2	2947.9	9.1	679.1	8.3	461.8
(40,60,7,20,22)	14.0	10.9	7.4	14.4	12.0	1.4	9.8	8.4	2.5	1579.1	4.1	520.1	1.7	2939.1	7.9	690.7	8.3	457.0
(40,60,7,20,25)	15.3	15.0	7.9	14.1	13.2	1.4	11.4	8.5	3.5	1485.4	4.5	533.9	1.4	2902.5	8.1	717.2	8.1	459.6
(40,60,7,20,30)	17.4	10.9	8.5	14.0	15.8	1.4	13.6	8.4	5.2	1493.1	6.0	534.9	0.8	2495.4	7.4	763.0	8.7	306.6
Average	12.2	4.4	5.7	5.4	10.4	0.6	8.7	3.8	3.1	462.0	3.7	187.4	0.9	1002.7	5.4	221.2	7.8	124.4

5 Conclusion

We have proposed several new families of heuristics for a difficult tool loading problem arising in flexible manufacturing. With respect to previous algorithms, we have introduced two new distance functions to guide the search, we have considered the use of generalized insertion methods (GENI and GENIUS), and we have developed new families of methods driven by the TLP objective. Tests performed on several sets of generated instances indicate that some of the proposed strategies yield very fast algorithms, or solution values that rank among the best available. In a given industrial context, the choice of the most appropriate method should depend on whether computation time or solution quality is the determinant factor.

Acknowledgements

This work was partially supported by the Canadian Natural Sciences and Engineering Research Council under grant OPG 0039682, by the Swiss National Scientific Research Council under grant 21-45574-95, and by the Center for International Business Education. This support is gratefully acknowledged. Thanks are also due to the referees for their valuable comments.

References

- [1] AVCI, S., AKTURK, M.S. (1996), "Tool Magazine Arrangement and Operations Sequencing on CNC Machines", *Computers & Operations Research* 23, 1069–1081.
- [2] BARD, J.F. (1988), "A Heuristic for Minimizing the Number of Tool Switches on a Flexible Machine", *IIE Transactions* 20, 382–391.
- [3] CRAMA, Y., KOLEN, A.W.J., OERLEMANS, A.G., SPIEKSMAN, F.C.R. (1994), "Minimizing the Number of Tool Switches on a Flexible Machine", *International Journal of Flexible Manufacturing Systems* 6, 33–54.
- [4] FOLLONIER, J.-P. (1994), "Minimization of the Number of Tool Switches on a Flexible Manufacturing Machine", *Belgian Journal of Operations Research, Statistics and Computer Science* 34, 55–72.
- [5] GAREY, M.R., JOHNSON, D.S. (1979), *Computers and Intractability. A Guide to the Theory of NP-Completeness*, Freeman, San Francisco.
- [6] GENDREAU, M., HERTZ, A., LAPORTE, G. (1992), "New Insertion and Postoptimization Procedures for the Traveling Salesman Problem", *Operations Research* 40, 1086–1094.
- [7] GOLDEN, B.L., STEWART JR., W.R. (1985), "Empirical Analysis of Heuristics", in *The Traveling Salesman Problem. A Guide Tour of Combinatorial Optimization*, (E.L. LAWLER, J.K. LENSTRA, A.H.G. RINNOOY KAN and D.B. SHMOYS, eds.), Wiley, Chichester, 207–249.

- [8] GRAY, A.E., SEIDMANN, A., STECKE, K.E. (1993), "A Synthesis of Decision Models for Tool Management in Automated Manufacturing", *Management Science* 39, 549-567.
- [9] HERTZ, A., WIDMER, M. (1996), "An Improved Tabu Search Approach for Solving the Job Shop Scheduling Problem with Tooling Constraints", *Discrete Applied Mathematics* 65, 319-345.
- [10] KIRAN, A.S., KRASON, R.J. (1988), "Automated Tooling in a Flexible Manufacturing System", *Industrial Engineering* 20, 52-57.
- [11] OERLEMANS, A.G. (1992), *Production Planning for Flexible Manufacturing Systems*, Ph.D. Dissertation, University of Limburg, Maastricht, The Netherlands.
- [12] SODHI, M.S., ASKIN, R.G., SEN, S. (1994), "Multiperiod Tool and Production Assignment in Flexible Manufacturing Systems", *International Journal of Production Research* 32, 1281-1294.
- [13] STECKE, K.E., KIM, I. (1988), "A Study of FMS Part Type Selection Approaches for Short-Term Production Planning", *International Journal of Flexible Manufacturing Systems* 1, 7-29.
- [14] STECKE, K.E., KIM, I. (1991), "A Flexible Approach to Part Type Selection Using Part Mix Ratios in Flexible Flow Systems", *International Journal of Production Research* 29, 53-75.
- [15] TANG, C.S., DENARDO, V. (1988), "Models Arising from a Flexible Manufacturing Machine. Part I: Minimizing the Number of Tool Switches", *Operations Research* 36, 767-777.