THE MINIMIZATION OF TOOL SWITCHES PROBLEM AS A NETWORK FLOW PROBLEM WITH SIDE CONSTRAINTS

Horacio Hideki Yanasse
Instituto Nacional de Pesquisas Espaciais - INPE/LAC
Av. dos Astronautas, 1758–Jd. da Granja. CEP 12227-010. São José dos Campos–S.P.

Maria José Pinto
Instituto Nacional de Pesquisas Espaciais – INPE/LAC
Av. dos Astronautas, 1758–Jd. da Granja. CEP 12227-010. São José dos Campos–S.P.

Abstract: *Consider a list of jobs that are to be processed in a flexible manufacturing machine. To process these jobs a set of tools is required. In order to perform a job, all the tools required to process it must be in the machine tool magazine. The number of tools required to process all jobs is, in general, larger than the magazine capacity. Hence, tool switches are necessary in order to process all jobs. We are interested in determining a sequence that minimizes the number of tool switches to perform all the jobs. We suggest the use of a formulation of this problem as a minimum cost network flow problem to solve this problem. We hope this formulation is worthwhile to be implemented in practice for machines with small sized tool magazine or with relative large magazine capacity (relative to the cardinality of the set of tools) .*

Key-words: Minimization of tool switches problem, mathematical formulation, minimum cost network flow problem.

1. Introduction

Consider the situation where one desires to sequence N jobs that require M tools to be processed in a machine with a tool magazine capacity of C. All the tools required to process a job must be in the machine before it is processed, hence, an implicit assumption is that each job requires at most C tools. Assuming $M > C$, there has to be tool switches in order to process all jobs. The minimization of the number of tool switches problem - MTSP consists in determining a sequence to process the jobs that leads to the least possible number of tool switches. The interest in finding such a sequence is, for instance, when the machine interruption time is proportional to the number of tools changed. Therefore, reducing the number of toosl switches implies in a larger avaliable machine production time.

The MTSP was considered by Tang and Denardo (1988), where a linear integer programming formulation of this problem is presented. The formulation uses a set of assignment constraints (the order in which the jobs are sequenced) together with other constrains that impose that the required tools for each job must be in the machine tool magazine whenever that job is processed.

According to Tang and Denardo (1988), their formulation performed poorly using commercial software solvers, that is, the computational time require was prohibitive, even when solving small instances of the problem. Therefore, these authors suggested the use of heuristics to solve the MTSP.

Tang and Denardo (1988) argued that MTSP is NP-Hard. Unfortunately, as pointed out in Yanasse (1997) and Crama et al. (1994), their argument has flaws. Crama et al. (1994) then reduced the problem of *minimal length traveling salesman path on edge graphs* to MTSP, and demonstrated that the problem is in fact NP-Hard (an even simpler proof uses a reduction from *consecutive blocks minimization* (Garey and Johnson, 1979, [SR17]).

Other formulations of the problem appeared in Bard (1988) and Crama et al. (1994). They propose integer programming formulations with a quadratic objective function. We did not find other works in the literature that explore the mathematical model formulation for MTSP of Tang and Denardo (1988), or Bard (1988) or Crama et al. (1994).

Recently Pinto and Yanasse (2001) performed a series of computational tests trying to solve MTSP using the comercial software CPLEX 6.5 in a Sun UltraSPARC II model Ultra 60 workstation, using Tang e Denardo's (1988) mathematical formulation of the problem. In a set of nineteen instances of varying sizes solved with M varying within the interval [8,10], N within the interval [8,10] and C within the interval [3,8], the average time for CPLEX to find a solution was 4.87 minutes. The range of the computational times observed was quite spread, varying from 0.32 minutes ($M = 10$, $N = 9$, $C = 8$) up to 32.21 minutes ($M = 9$, $N = 10$, $C = 3$). Even instances with the the same parameter sizes present such variation. For example, another instance of size $M = 9$, $N = 10$, $C = 3$ took less than 2 minutes. This seems to indicate that the computational time to solve MTSP with CPLEX 6.5 seems to be quite dependent on the problem data. As we increase the size of the instances considered the computational times become considerably larger.

Considering that, in practice, the magazine capacity is usually fixed, the fact that a large variability of time to get a solution for instances of the same size is not a desirable feature, the small size of the instances considered and the reasonable amount of computational effort necessary to solve these instances, we thought that there would be space for an alternative way to solve the MTSP which could handle or reduce some of the above mentioned problems. In the next section we present one such alternative which is based in a non-polynomial formulation of MTSP as a minimum cost network flow problem - NTFP. In section 3, we discuss some practical implementation issues of this formulation and in section 4, some final remarks are made.

2. MTSP as a NTFP

The total number of combinations of M tools into sets of C tools is given by $n = \binom{M}{C}$. Consider an undirected graph where each node represents one of such combinations. Any two nodes N_i and N_j of this graph is connected with an arc with cost C_{ij} given by the number of different tools in nodes N_i and N_j, that is, the number of tool switches that will occur when we move from node N_i to node N_j or vice-versa.

Consider two additional nodes in this graph, an origin node O and a destination node D. Nodes O and D are connected to all nodes N_i, for all i, in the graph and the cost of these connecting directed arcs are all equal to 0.

Let $J_1, J_2, \ldots, J_N$, be the N jobs to be processed and let $S_1, S_2, \ldots, S_N$, be the set of all nodes in the graph such that they contain all the tools required to perform jobs $J_1, J_2, \ldots, J_N$, respectively. The problem we want to solve is to send a unit of flow from node O to node D, at a minimum cost, with the additional constraints that at least one unit of flow must enter some node in sets $S_1, S_2, \ldots, S_N$, from some other node not in $S_1, S_2, \ldots, S_N$, respectively.

A solution to this problem defines a path joining O to D visiting at least one node of the sets $S_1, S_2, \ldots, S_N$, since there is an inflow to at least a node in set S_i, for any $i = 1, 2, \ldots, N$,. Following this path from O to D, we can determine the sequence in which the path visits (some node of) the sets $S_1, S_2, \ldots, S_N$. The node visited defines the tools that will be present in the magazine. Observe that if we are in some node of set S_i, the tools required by job J_i, $i = 1, 2, ..., N$, are available, therefore, job J_i can be processed. Hence, since we visit at least one node of the sets $S_1, S_2, \dots, S_N$, following the path we are able to process all jobs required. In other words, any feasible solution to this flow problem is a feasible solution to the tool switches problem. Conversely, any feasible solution to the tool switches problem is a feasible path in the flow problem considered. Therefore, any minimum cost solution to the flow problem will provide an optimal solution to the MTSP.

Let $\mathbf{x}_{jk}$ be a decision variable representing the flow going from node j to node k. Consider the following network flow problem model with side constraints:

Problem (P)

$$\text{Min} \sum\sum C_{ij}\,\mathbf{x}_{ij} \tag{1}$$

Subject to

$$\sum_j \mathbf{x}_{ij} - \sum_k \mathbf{x}_{ki} = 0 \quad \text{for i} = 1, 2, \dots, n \tag{2}$$

$$\sum_j \mathbf{x}_{Oj} = 1 \tag{3}$$

$$\sum_k \mathbf{x}_{kD} = 1 \tag{4}$$

$$\sum_{i \notin Sk} \sum_{j \in Sk} \mathbf{x}_{ij} = 1, k = 1, 2, ..., N \tag{5}$$

$$\sum_j \sum_k \mathbf{x}_{jk} \le N+1 \tag{6}$$

$$\sum_{i \notin Q} \sum_{j \in Q} \mathbf{x}_{ij} = 1, \text{ for all possible subsets Q} \tag{7}$$

$$\mathbf{x}_{ij} = 0 \text{ or } 1 \text{ for all i and j.} \tag{8}$$

The objective function (1) together with constraints (2), (3) (4) and the non-negativity of the decision variables would compose a regular minimum cost network flow cost where a unit of flow is being sent from node O to D. For this particular flow problem, constraint (8) is automatically satisfied since the optimal solution given by some simplex based method would provide a 0-1 integer solution.

We introduce the side constraints (5), (6) and (7). Constraints (5) simply state that a unit of flow, from some node outside set S_k, must enter into some node in S_k, k = $1, 2, ..., N$. Constraint (6) imposes that the flow must pass through at most (N+1) arcs. The flow has to go from node O to node D passing through some nodes i = $1, 2, \dots, n$. In fact the flow has to pass through N sets of nodes (S_k, k = $1, 2, ..., N$), according to (5). Hence, there must be at most N+1 arcs where the flow must pass to form a path from O to D passing in one node of each set S_k, k = $1, 2, ..., N$.

Unfortunately, these constraints are not enough to prevent cycles. We have to introduce constraints (7) to prevent cycles. The subsets Q are unions of the sets $S_1, S_2, \dots, S_N$, and we have to consider all possible combinations. These constraints impose that for any subset of the sets $S_1, S_2, \dots, S_N$, there must be a unit of inflow coming from node O or nodes not belonging to the combined subsets. Observe that in constraints (7) constraints (5) are included. We are presenting both in this way for convenience of the discussions presented in the subsequent sections.

In the next section, some issues concerning this formulation and computational implementation are considered.

3. Implementation issues

We can see that there are a few drawbacks with the proposed approach. The first issue that deserves attention is the number of nodes *n* necessary to be considered. Let us assume, for the moment, that the integrality constraint of the proposed 0-1 linear integer problem can be relaxed and the problem can be solved using any commercial LP code. LP solvers, nowadays, are quite efficient and can solve quite large problems. Therefore, for *n* not very large, the proposed approach would be better or competitive to other approaches already suggested in the literature. *n* is the number of possible combinations of all tools in sets of size *C*. Therefore, *n* will be "small" when *C* is small or when *C* is relatively large with respect to *M* (the total number of tools). In Table 1 we provide the value of *n* for some values of *C* and *M*.

M	*C*	*n*	*M*	*C*	*n*	*M*	*C*	*n*
10	2	45	10	8	45	20	2	190
10	3	120	10	7	120	20	3	1140
10	4	210	10	6	210	20	4	4845
10	5	252				20	5	14535
9	2	36	9	7	36	20	6	33915
9	3	84	9	6	84	20	7	62985
9	4	126	9	5	126	20	8	125970
8	2	28	8	6	28	30	3	4060
8	3	56	8	5	56	30	4	27405
8	4	70				40	3	9880
11	2	55	11	9	55	40	4	91390
11	3	165	11	8	165	50	3	19600
11	4	330	11	7	330	50	4	230300
11	5	462	11	6	462	60	3	34220
12	2	56	12	10	56	60	4	487635

Table 1 – Number of nodes in the graph as a function of *M* and *C*

From Table 1, we can see that for instances with *M* varying within the interval [8,10], and *C* within the interval [3,8], the largest graph will have at most 252 nodes (not counting the nodes that are replicated). For such a size, the LP relaxation of the problem can be solved without much difficulty.

We may be able, in certain cases, to reduce the total number of nodes to be considered with a pre-processing analysis of the data. Consider that there are *C* tools in the tool magazine but these tools are not sufficient to process any of the required jobs. This combination, therefore, will never be used in a practical solution to the problem, hence, there is no need to be considered in the graph problem as well, that is, a node corresponding to such a combination can be deleted from the graph. Consider, for example, the instance given in Table 2 with *M* = 5, *N* = 8 and *C* = 3. In Table 2 the tools required by each job are indicated with an 1; an 0 indicates that the corresponding tool is not necessary to process that job; the job number is presented in the first column.

For this instance, the number of nodes to be considered in the graph is $\binom{5}{3} = 10$ nodes and they are listed in Table 3. Among these nodes, node 1, corresponding to the combination 123, does not need to be considered since with these tools in the magazine none of the 8 jobs can be processed. This can be also verified in Table 4 where the nodes contained in the sets S_1, S_2, …, S_8 are indicated.

Jobs	**Tools**				
	1	2	3	4	5
1	1	0	0	1	0
2	1	0	1	0	1
3	0	1	0	1	0
4	0	0	1	1	0
5	1	0	0	0	1
6	0	0	1	0	1
7	0	0	0	1	1
8	1	1	0	1	0

Table 2: Problem instance, $M = 5$, $N = 8$ and $C = 3$.

Nodes	**Tools**				
	1	2	3	4	5
1	1	1	1	0	0
2	1	1	0	1	0
3	1	1	0	0	1
4	1	0	1	1	0
5	1	0	1	0	1
6	1	0	0	1	1
7	0	1	1	1	0
8	0	1	1	0	1
9	0	1	0	1	1
10	0	0	1	1	1

Table 3: Nodes in the graph for instance given in Table 2

Sets	Nodes									
	1	2	3	4	5	6	7	8	9	10
S_1		x		x		x				
S_2					x					
S_3		x					x		x	
S_4				x			x			x
S_5			x		x	x				
S_6					x			x		x
S_7						x			x	x
S_8		x								

Table 4 – Nodes contained in the sets S_i , for the instance given in Table 2

In some cases, it is possible to reduce the number of constraints (5) in problem (P). Consider the instance given in Table 2. From Table 4, we see that node 2 is the only possible node that satisfies the requirements for J_8. Therefore, in an optimal solution, a flow of one will necessarily pass through node 2. Since we already know that this node will be part of the path in an optimal solution, then there is no need to include constraint (5) for $k = 1$ and $k = 3$ since they will be automatically satisfied. Similarly, node 5 is the only possible node that satisfies the requirements for J_2, therefore there is no need to include constraint (5) for $k = 5$ and $k = 6$ since they will be automatically satisfied. If we delete the rows corresponding to S_1, S_3, S_5 and S_6 we obtain Table 5.

Sets	Nodes									
	1	2	3	4	5	6	7	8	9	10
S_2					x					
S_4				x			x			x
S_7						x			x	x
S_8		x								

Table 5 – Set S_i needed to be considered, for the instance given in Table 1

Suppose we already reduced the number of constraints (5). In correspondence with constraints (5), the right hand side value of constraint (6) will become 5 (the flow has to go from node O to node D passing only through these 4 sets, S_2, S_4, S_7 and S_8) instead of 9. Also, the combinations to be considered in constraints (7) are reduced accordingly.

It is also worthed noting that if the number of tools M and the capacity of the magazine C are fixed, the cost matrix to be considered in the model is totally defined and is independent of the instance being solved. For instance, for the instance being considered in Table 2 and all problems of sizes $M = 5$ and $C = 3$, the matrix cost is the one given in Table 6. Recall that the cost are determined by computing the number of tool switches that occur when we move from a node to another. For instance, when we go from node 5 to node 6, the cost is 1 since there is a single switch of tool 3 to tool 4; when we move from node 1 to node 10, the cost is 2 since there are two switches, tools 1 and 2 for tools 4 and 5, and so on.

C_{ij}	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	1	2	1	1	2	2
2	1	0	1	1	2	1	1	2	1	2
3	1	1	0	2	1	1	2	1	1	2
4	1	1	2	0	1	1	1	2	2	1
5	1	2	1	1	0	1	2	1	2	1
6	2	1	1	1	1	0	2	2	1	1
7	1	1	2	1	2	2	0	1	1	1
8	1	2	1	2	1	2	1	0	1	1
9	2	1	1	2	2	1	1	1	0	1
10	2	2	2	1	1	1	1	1	1	0

Table 6 – Costs matrix C for problems of sizes $M = 5$ and $C = 3$

The costs C_{Oj} and C_{jD}, for j = 1, 2, …, 10 are zero (and C_{jO} and C_{Dj}, for all j are infinite).

Let us discuss now the dificulties in solving the proposed network flow problem (P). If there were no side constraints (5), (6) and (7), an integer optimal solution can be obtained easily relaxing the integrality constraint and using any specialized simplex method for solving a minimum cost network flow problem.

Suppose we solve problem (P) by the simplex method with the constraints (8) replaced solely by constraints requiring the non-negativity of the decision variables. If the solution obtained is integer, the solution is also optimal for (P) and we are done. The problem is that there is no guarantee that the solution obtained with the relaxed problem will be integer.

Every solution obtained by the simplex method will determine a path or paths from O to D where some positive amount of the commodity is flowing from O to D. If there is a single path with positive flow, then we are done (the amount of the commodity flowing should be 1). Otherwise, a fractional value of the commodity is being sent using different paths from O to D.

We are confident that in practice, in most of the instances, a traditional branch-and-bound method that sets the values of the decision variables to 0 or 1 for some of the fractionary variables will quickly converge to an integer solution. Following a path with positive flow, we propose that the variable that should be set to 0 or 1 corresponds to the first arc in the path from O to D that is fractionary. Once this is done, all the subsequent arcs in the path that had fractionary values will be forced to be integer (0 or 1). Observe that setting some other fractional value variable to 0 or 1 will force the subsequent arcs in the path to 0 or 1, but not necessarily all the previous arcs in the path.

Unfortunately, for the time being, we have no theoretical based support to guarantee that this quick convergence to an integral solution will happen. Some limited computational tests will be performed to verify empirically whether this proposed approach is effective.

Let us discuss now the constraints (7) that prevent loops. If N is the number of sets S_1, S_2, …, S_N, the total number of possible combinations is 2^N. For $N \leq 10$, $2^{10} \leq 1024$, that is, the number of constraints is large but not that much. For larger values of N the number of constraints increases rapidly. We propose to solve problem (P) without the constraints (7) (only with constraints (5)) and we introduce them when necessary. Whenever a solution with a loop is found we introduce the corresponding constraint (7) that prevents that loop to occur and we solve the problem again. We are also confident that the number of constraints to be considered until an optimal solution is obtained will not be large because of the constraint (6). We also have to perform some computational tests to verify this feeling.

4. Final remarks

In this work we present a minimum cost network flow problem formulation with side constraints for the minimization of the tool switches problem.

The proposal seems to be promising, mainly for practical instances of small sizes or with small tool magazine capacities or even with relatively large tool magazine capacities. Observe that in practical settings, M can vary from instance to instance but the machine capacity C is fixed. Assume that $M \leq$ U, for some U > 0 and integer. Then, it is possible to generate beforehand the graphs associated to any instance of the problem and the corresponding cost matrix C. In fact, it is sufficient to generate the graph G_U associated with an instance with $M =$ U and corresponding cost matrix C_U. Any other instance with $M <$ U can be obtained from the graph G_U by deleting some nodes and the corresponding cost matrix can be obtained by deleting some rows and columns of C_U. So, in practice, to solve an instance, only the constraints (5), (6) and (7) associated to the instance need to be generated.

The proposed network flow formulation of the problem has some special features that may be explored to develop a specialized algorithm that can solve it in a more efficient way. There is also space for research to build some theoretical support to guarantee that the solution procedure suggested will quickly converge to an integer solution.

It is worthed observing that in the pre-processing process suggested there are sometimes some useful information that arises. For instance, from Table 5 and the cost matrix C, one can easily observe that unless node 10 is in the optimal flow path, the lower bound of 2 (an immediate lower bound for the number of tool switches in this instance is M - C = 2; see Yanasse, 1997) for this instance of the problem cannot be achieved since all C_{ij} are greater or equal to 1 for all $i \neq j$ and if node 10 is not part in the optimal path, then the flow from O to D must pass through at least 4 nodes (nodes 2 and 5 plus at least two other nodes). Therefore, we can solve this instance imposing that a flow will have to pass through node 10. We may obtain a solution of 2, in which case we know it is optimal, or a solution of 3 or more. If the solution obtained is 3, we are also done. There is no other solution better than 3. If the solution is larger than 3, we have to proceed with the search, checking for paths not passing through node 10.

Finally, another possibility that is being explored by these authors is the use of an alternative network flow formulation for the problem using two commodities as described in Ahuja, Magnanti and Orlin (1993), p.624, that avoids the enumeration of constraints (7).

Acknowledgements: This work was partially financed by FAPESP, CNPq and CAPES.

5. References

Ahuja, R.K.; Magnati, T.L.; Orlin, J.B. Network Flows – Theory, Algorithms, and Applications, Prentice Hall, Englewwod Cliffs,1993.

Bard, J. F. (1988), "*A Heuristic for Minimizing the Number of Tool Switches on Flexible Machine*". IIE Transactions vol. 20, 382-391.

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

Garey, M.; Johnson, D.S. Computers and Intractability, Freeman, New York, 1979.

Pinto, M.J; Yanasse, H.H. *"The minimization of tool switches problem: proposals for its solution"*. Presented at XXXIII SBPO – Brazilian Operations Research Symposium, held in Campos de Jordão, SP, 6 to 9 of September, 2001. Published in CD-ROM, p.1410-1419 (in portuguese).

Tang, C. S.; Denardo, E. V.(1988), "*Models Arising from a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches*". Operations Research 36, 767-777.

Yanasse, H.H. "On a pattern sequencing problem to minimize the maximum number of open stacks". European Journal of Operational Research, 100, 454-463, 1997.