2016 International Conference on Applied Mathematics and Mechanics (ICAMM 2016)
ISBN: 978-1-60595-399-1

Machines Setup Reduction in Lean Manufacturing Environment

Francesco NUCCI

Department of Engineering for Innovation, University of Salento, Via Monteroni, 73100, Lecce, Italy

Keywords: Tool management, Setup reduction, Lean manufacturing.

Abstract. We aim to minimize of Total Idle Time for Tool Change (TITTC) in machines equipped with rotating tool magazines and un-buffered tool chain. The subject of this work consists in the optimization of TITTC that is the sum of the tool setup time and tool change time. The problem consists in determining the optimal tool configuration for each part type considering the transition between the current part type and the next one. We propose a Genetic Algorithm to solve the problem. A numerical real test study is performed in order to investigate the problem. We proposed different frameworks to evaluate the proposed approach advantages. First, we reduce TITTC considering the transition between the current part type production and the next one. Finally, we discuss the variation of the part program sequence, according to precedencies. Results show important time saving can be obtained with the proposed approach.

Introduction

In today's manufacturing, shorter cycles and diminishing batch sizes characterize part processing, while the variety of product types and models continues to increase. A quickly respond to changing customer demands requires the use of manufacturing systems easy to re-configure and expanded. Lean manufacturing is an important starting point for companies wanting to face such problems. Indeed, the main concept of lean manufacturing refers to 'less is better' approach and leads to a simplified environment tuned to the manufacturer's demands.

Numerically Controlled (NC) machine tools use different types of cutting tools to machine a part. The tools, kept in a magazine, are taken automatically for use in the order determined by the manufacturing Part Program (PP). Tool change time has to be minimized. Commonly in lathes, vertical machining centers and punching machines, tool revolvers contain between 4 and 70 tools. Revolver rotation performs the tool selection process. After the operation has completed, the tool returns to its previous position in the magazine, which rotates until the next tool is in the loading position. The revolver type tool magazine is common in small machining centers. Rotating tool magazines operate so that machine spindle loads the tool directly from its magazine position. On one hand, an automatic tool change system reduces cycle times by changing tools between cuts (buffered tool change). On the other, if no automatic tool change system exists (unbuffered tool change), optimizing tool location reduces magazine rotation time. Changing of tool positions involves manual updating of the tool data in the machine controller and editing of NC programs. In the case of revolver type magazines, the optimization of tools is particularly relevant because of tool weight, the slow tool changing movements, and the large number of tools. The tool change waiting time caused by magazine movement is affected by the order in which the tools are positioned in the magazine. Positioning tools in the magazine so that tools successively used in the PP are situated close to each other, leads to minimize magazine movements. We assume that the number of tools used is smaller than magazine size. In the case of unbuffered tool change, machine must wait while magazine rotates. This time is related to the rotating length, used as optimization criterion. In this case, we have the classical quadratic assignment problem (QAP), [1]. The problem is to assign M tools to N magazine positions ($M \leqslant N$), in order to minimize the revolver magazine total traveling. The QAP is well known and hard to solve optimally. Generally, problems with more than 1500 variables (tools) cannot be solved optimally, [1]. However, efficient heuristics produce good results. In [2], an optimal graph theoretic solution method for a non-reversing magazine is presented. In [3], a review of the literature is performed. The presented problem has been addressed by [4] using Genetic Algorithms (GAs). A novel representation-scheme enables easy manipulation during feasible neighborhood solution generation. In [5] and [6] the important task consists in generating optimal index positions of cutting tools to reduce the non-machining time and to optimize process plans. Recently, in [7], tool duplication and new questions, concerning the demanded parts, were considered. In this paper, we focused on optimization of Total Idle Time for Tool Change (TITTC) that is the sum of the tool setup time and tool change time. If the number of demanded parts is high, a complete setup magazine leads to easily minimize the TITTC. On the contrary, if few parts are demanded, the complete magazine setup time can be greater than the saving obtained in terms of tool change time. This paper is organized as follows. In Section 2, the notation and the formal problem statement are reported. Section 3 considers the solution algorithm. In Section 4, a variant for the considered problem is presented. In Section 5, an experimental campaign based on industrial case studies is presented to evaluate the proposed techniques.

Problem Statement

Assigned an initial Tool Magazine Configuration (TMC), the problem consists in assigning M tools to N tool positions in the magazine, which minimizes the TITTC as the sum of Tool Setup Time (TST) and magazine rotation time. The first represents the time required to reallocate a set of tools in the magazine. The second is the time spent by the magazine in rotation during tool changes. We assume, for simplicity, that each tool is used once in the PP. For each tool change, at least one magazine rotation unit has to be performed. If two consecutive tools are not adjacent, two or more rotation units are made. A Magazine Extra-Rotation (MER) is the number of rotation units performed by the magazine minus one. If two consecutive tools are adjacent, MER is null. On one hand, if the magazine is completely configured, so that all consecutive tools are in adjacent positions, TST assumes the maximum value, whereas MER time is null. On the other, if no tool is reallocated in the magazine, TST is null but MER is considerable. Table 1 reports parameters and variables. Since each tool is used once, δ_{ij} are binaries. Terms p_{ik} describe the previous magazine configuration. Variables s and P define the trade-off between TST and MER time. Variables x_{ik} and y_i denote the solution. The model is stated as [1] in Eq. 1-4. We first focus on the objective function in Eq. 1, representing TITTC. The first part of Eq. 1 represents the total TST, whereas, the second considers the MER time. Indeed, the extra-traveling time generated by assigning tool i to location k and tool j to location l is the product of MER time between the two locations k and l and the number of tool changes from tool i to j, that is $\delta_{ij}d_{kl}x_{ik}x_{jl}$. The sum of the total traveling distance is in Eq. 1. Eq. 2 constraint guarantees that at most one tool is assigned to each position. Eq. 3 ensures that each tool is assigned to a tool position. Eq. 4 checks if the setup on tool i is performed.

Table 1. Parameters and variables.

Symbol	Description
$i\ j$	tool index, $i,j=1,...,M$
$k\ l$	magazine position index, $k,l=1,...,N$
$\square_{ij}$	number of tool changes between tools i and j
d_{kl}	MER time between tool positions k and l
p_{ik}	=1 if tool i is initially at magazine position k, otherwise 0
$\square$	setup time for a generic tool
P	number of pieces to be produced
x_{ij}	=1 if tool i is assigned to position k, otherwise 0
y_j	=1 if setup is required for tool i, otherwise 0

$$\min s \sum_{i=1}^{M} y_i + P \sum_{i=1}^{M} \sum_{k=1}^{N} \sum_{j=1}^{M} \sum_{l=1}^{N} \delta_{ij} d_{kl} x_{ik} x_{jl} \tag{1}$$

$$\sum_{i=1}^{M} x_{ik} \leq 1 \qquad \forall k = 1, \dots, N \tag{2}$$

$$\sum_{k=1}^{N} x_{ik} = 1 \qquad \forall i = 1, \dots, M \tag{3}$$

$$x_{ik} - p_{ik} \leq y_i \qquad \forall i = 1, \dots, M \quad \forall k = 1, \dots, N \tag{4}$$

The number of possible relevant arrangements in a full magazine is $(M-1)!/2$. The number does not depend on the location of the first tool and half of the possible arrangements can be left out because of symmetry. Literature problems consider only the minimization of the second part of our objective function (in Eq. 1). Instead, in our case, we deal with the trade-off between TST and MER time in order to reduce the TITTC of the machine.

A simple illustrative example follows. Magazine capacity is N=5. PP is made of 5 operations using 5 tools, M=5. Initial TMC is [1 2 3 4 5], that is Tool I is placed at position $k=i$. Part type PP, expressed as tool sequence, is [3 1 2 5 4], that is $\delta_{31}=\delta_{12}=\delta_{25}=\delta_{54}=1$. We assume TST is $s=30$ time units and MER time $d_{13}=d_{24}=d_{35}=\ldots=1$ time units. Finally, the number of demanded parts is P=10. If no setup is performed (Scenario A), the processing uses the previous TMC and no TST is required. For each part, we have 2 MER (from Tool 3 to 1 and from Tool 2 to 5) of 1 time unit each. Consequently, to process P=10 parts, $TITTC^A$=20 time units. In case of a complete setup, according to the part program sequence (Scenario B), the new TMC is [2 1 3 4 5]. In this case, no MER is required but 2 tool setup operations are necessary to swap Tool 1 and Tool 2. Consequently, in this scenario $TITTC^B$=60 time units, and scenario A is preferable. Instead, assuming the number of demanded parts is P'=50, we obtain $TITTC^{A'}$=100 time units; consequently a complete setup (Scenario B) is optimal. Usually, the empirical rule 'CS-or-NS' (Complete Setup or No Setup) can determine the TMC as in the previous example. Instead, our approach consists in finding a good compromise between the 2 scenarios "Complete Setup" and "No Setup", based on TITTC.

Proposed Approach

We employ a Genetic Algorithm (GA) based approach to solve the problem described in Section 2. First, we encode each potential solution in the form of a string and evaluate it by a fitness function. In a GA-based method, three operators further process the solution population: Reproduction, Crossover, and Mutation. We used a fixed-length integer codification for the chromosomes to provide the tool magazine configuration. Fig. 1 shows the encoding string where $z_i=k$ so that x_{ik}=1; string length is M. We reported the position z_i for each tool i. In Fig. 1, shows the example described in previous section example (Scenario B, complete setup). We randomly generate the initial population. Crossover is the process of transfer of genetic material from selected chromosomes to child to generate a new chromosome set, having the characteristics of both parents. Crossover operator consists in combining the properties of current solutions to generate still better solutions. A simple crossover scheme does not work as it makes the chromosomes inconsistent i.e. some positions may be repeated. For this reason, we adopted Partially Matched Crossover (PMX) mechanism. First, crossover is performed. In the offspring obtained, some coefficients z_i can be replicated. Cross-referencing with the parent of the alternate chromosome produces consistent offspring. Mutation performs changes in the chromosome to search in a wide search space to reach a global optimum. We adopted an ordinary approach: we randomly select 2 tools and exchange their positions in the magazine. We selected fitness value on the basis of Eq. 1.

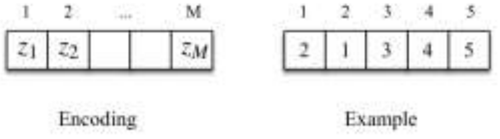

Figure 1. Encoding.

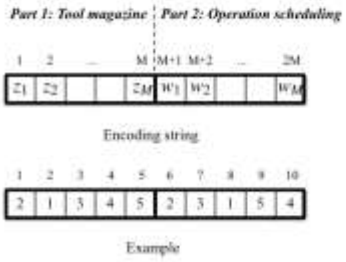

Figure 2. Encoding with operation scheduling.

Operation Scheduling

Usually a part type processing is made of a specific sequence of operations referred to as part program. Part type requirements determine the PP. Such a list considers the precedence relationships between operations i.e. real technological constraints. Usually, multiple sequences satisfy such precedence relationships. The best one is selected based on user-defined criteria. In the considered problem, we select the best PP, among those ones that satisfy precedence relationships, considering the 'TITTC minimization'. In this case, we re-formulated our problem as scheduling with constraints between tasks. Consequently, PP sequence turns in a decisional variable. For each tool i, a set of tools Γ_i, preceding i, is specified. In sucha way, we obtain a Network Part Program (NetPP), see [8]. NetPP is more general than a Sequence Part Program (SPP) because the order is not set in advance. We used parameter $\gamma = \frac{1}{M}\sum_{i=1}^{M}|\Gamma_i|$to evaluate the degree of rigidity of NetPP. In case of SPP, γ assumes the maximum$(M-1)/2$. Whereas, if no precedence constraint exists, any operation order is valid and $\Gamma_i=\emptyset$ and $\gamma=0$. In the new formulation, parameter δ_{ij} turns into a variable, because tool changes between tools i and j depend on the PP scheduling. We introduce a new variable w_i indicating the sequence number in the PP of tool i operation. We added constraints in Eq. 6-10 to the previous model (Eq. 1-4). In particular, constraints in Eq. 6-7 allow having distinct and consecutive numbers for w_i. Eq. 8 illustrates constraints. Constraint in Eq. 9 establishes the relationship between variables w_i, w_j and δ_{ij}.

$$1 \leq w_i \leq M \qquad \forall i = 1, \dots, M \tag{6}$$

$$w_i \neq w_j \qquad \forall i, j = 1, \dots, M \quad i \neq j \tag{7}$$

$$w_j < w_{ji} \qquad \forall i = 1, \dots, M \qquad \forall j \in \Gamma_i \tag{8}$$

$$1 - (w_i - w_j - 1)/M \leq \delta_{ji} \;\; \forall i, j = 1, \dots, M \quad i \neq j \tag{9}$$

Since in the objective function of the new model, Eq. 1, we turned parameters δ_{ij} into variables, we obtained a new model harder to solve than previous one. Consequently, we define a new GA to find the optimum. Considering the original Algorithm developed at Section 3, the GA solving the new problem takes into account additional questions. First, the encoding considers also the PP data, as reported in Fig. 2: an example is reported with the encoding for magazine configuration and operation scheduling described Section 2. Crossover and Mutation phases are adapted in order to work both on part 1 and 2 of the new string in Fig. 2. In the fitness function, we considered also the Number of Failed Precedence Constraints.

Computational Experience

In this section, we analyzed the performance of algorithm described in Sections 3 and 4. We developed different experimental campaigns. First, focusing on the base problem, we compared the QAP model in Eq. 1-4 in Section 2, with the corresponding GA reported in Section 3.Second, considering the operation-scheduling problem in Section 4, we measured the difference between the Section 3 and 4 GAs. IBM CPLEX v.12.5 is adopted to solve QAP model with a time limit equal to the execution time of the GA counterpart. The industrial test case is described in Table 2 in terms of part types, parts to be produced P, tools M and degrees of rigidity of both SPP and NetPP. The magazine capacity is $N=60$. Tool magazine rotation time is 1 sec/slot (bidirectional). Tool setup time is $s=30$ sec.

Table 2. Real industrial test case.

Part type	P	M	γ^{SPP}	γ^{NetPP}
1	370	45	22.0	20.4
2	379	46	22.5	20.8
3	359	47	23.0	21.3
4	359	50	24.5	21.5
5	347	51	25.0	22.8

Analysis GA vs. QAP

We compared the QAP (Section 2) with the GA (Section 3). We dealt with 3 experimental campaign factors: the Part type to be processed (Table 2); the reduction coefficient ρ of the number of demanded parts (the number of demanded pieces is *P/ρ*); the ‘Initial Configuration’ of the tool magazine. The KPI consists in the TITTC saving with GA (Section 3) respect to the QAP. Table 3 reports results. For example, considering the part type 1 and *ρ*=10, the number of demanded parts is 370/10=37. If the initial TMC is that one associated to part type 2, the TITTC obtained with GA is 0.5% lower than the one achieved with QAP. QAP supplies reasonable results when ρ=1 and ρ=100 are considered. Instead for ρ=10, since the trade-off existing between ‘complete setup’ and ‘no-setup’ policies, the gap between GA performs better than QAP.

Analysis SPP vs. NetPP

We compared the SPP GA (Section 3 with NetPP GA (Section 4). NetPP allows selecting the next operation based on the TITTC minimization. Table 4 reports results. NetPP supplies better results, especially when part type 4 is considered. Indeed in Table 2, part type 4 has the maximum reduction of γ passing from SPP to NetPP.

Table 3. Comparison between GA and QAP.

Initial TMC		1	2		3	4	5
Part type	*ρ*		% TITTC saving of GA vs. QAP				
1	1	-	0.0		0.1	0.0	0.0
2	1	0.1	-		0.1	0.1	0.0
3	1	0.0	0.1		-	0.1	0.0
4	1	0.0	0.0		0.0	-	0.0
5	1	0.0	0.1		0.0	0.0	-
1	10	-	0.5		1.6	1.8	1.2
2	10	1.6	-		1.4	0.6	1.2
3	10	3.9	2.4		-	1.9	3.4
4	10	1.8	0.6		1.8	-	2.3
5	10	1.8	2.3		2.0	4.6	-
1	100	-	0.5		0.1	0.5	0.1
2	100	0.3	-		0.4	0.7	1.3
3	100	0.9	0.5		-	0.2	0.7
4	100	0.7	0.2		0.3	-	0.8
5	100	1.1	0.8		0.2	0.3	-

Table 4. Comparison between NetPP and SPP.

Initial TMC		1	2	3	4	5
Part type	*ρ*	% TITTC saving of NetPPvs. SPP				
1	1	-	4.1	2.6	2.5	0.4
2	1	1.0	-	0.2	1.2	1.9
3	1	2.9	4.0	-	1.4	2.1
4	1	9.7	3.4	12.4	-	4.4
5	1	1.1	5.8	0.4	4.7	-
1	10	-	0.4	0.1	1.5	3.0
2	10	1.9	-	3.7	4.8	0.3
3	10	0.6	3.2	-	4.6	2.3
4	10	10.3	5.6	1.7	-	3
5	10	1.1	3.9	3.6	0.7	-
1	100	-	3.4	4.1	2.5	3.7
2	100	4.8	-	1.5	3.8	4.3
3	100	4.5	3.4	-	3.6	4.4
4	100	0.5	6.9	2.0	-	10.6
5	100	4.2	1.6	5.8	3.7	-

References

[1] E. Cela. The quadratic assignment problem: theory and algorithms. Kluwer, Dordrecht, 1998.

[2] G. Levitin, J. Rubinovitz, Algorithm for tool placement in an automatic tool change magazine. International Journal of Production Research 33 (1995) 351-360.

[3] E. Niemi, Optimal tool magazine operation. Part 1: Rotating magazines with unbuffered tool change. The International Journal of Advanced Manufacturing Technology, 22 (2003) 720-726.

[4] T. Dereli, I. Filiz, Allocating optimal index positions on tool magazines using genetic algorithms. Robotics and Autonomous Systems, 33(2000) 155-167.

[5] A.G. Krishna, K.M. Rao, Optimal allocation of index positions on tool magazines using an ant colony algorithm. Int. Journal of Advanced Manufacturing Technology, 30(2005) 717-721.

[6] M. Solimanpur, R. Rastgordani, Minimising tool switching and indexing times by ant colony optimisation in automatic machining centres. Int. Journal of Oper. Research 13(2012) 465-475.

[7] A. Baykasoğlu, F.B. Ozsoydan, An improved approach for determination of index positions on CNC magazines. International Journal of Production Research 54(2016) 742-760.

[8] A. Matta, T. Tolio, F. Tontini, Tool management in flexible manufacturing systems with network part program. International Journal of Production Research 42(2004) 3707-3730.