Scheduling cutting machines in a clothing industry

SCHEDULING CUTTING MACHINES IN A CLOTHING INDUSTRY

Alessandro Agnetis * **Paolo Detti** *

* *Dipartimento di Ingegneria dell'Informazione - Università degli Studi di Siena, via Roma 56 - 53100 Siena (Italy), e–mail: agnetis@dii.unisi.it, detti@dii.unisi.it*

Abstract: In a clothing industry, clothes are produced by assembling fabric cutouts. Fabric rolls on a line, and is cut by automated machines arranged along the line. A fabric cutout f_j, requiring time p_j to be cut, is available for cutting on the first machine at a release date r_j^1, and must be completed, on the same machine, within a deadline D_j^1. On machine M_i, $i = 2, \ldots, m$, release dates and deadlines related to the cutout f_j are $r_j^i = r_j^1 + \Delta(i-1)$ and $D_j^i = D_j^1 + \Delta(i-1)$ respectively, where Δ is a constant factor depending on the speed rate of the fabric. A set–up time occurs on a machine, when switching from one cutout to another. The problem of assigning and scheduling cutting tasks to the minimum number of machines is addressed.

Keywords: Manufacturing systems, scheduling algorithms, heuristics, planning, real time.

1. INTRODUCTION

This paper addresses a scheduling problem arising from an application in a real industrial context, namely the cutting department of a clothing industry. Clothes are produced by assembling fabric cutouts. In the cutting department, fabric rolls on a line and is cut by automated machines arranged along the line. Machines are all identical.

In figure 1 the cutting system and its technical data are reported.

A fabric cutout f_j, $j = 1, \ldots, n$, requiring a cutting time p_j, is available for cutting on the first machine M_1 at a release date r_j^1, and must be completed, on the same machine, within a deadline D_j^1. Since machines are arranged along a line, the same cutout f_j will be available for cutting on a machine M_i, $i = 2, \ldots, m$, at a release date $r_j^i = r_j^1 + \Delta(i-1)$ and within a deadline $D_j^i = D_j^1 + \Delta(i-1)$, $\Delta = \frac{Ws}{\sigma}$, where σ is the constant speed rate of the fabric and Ws

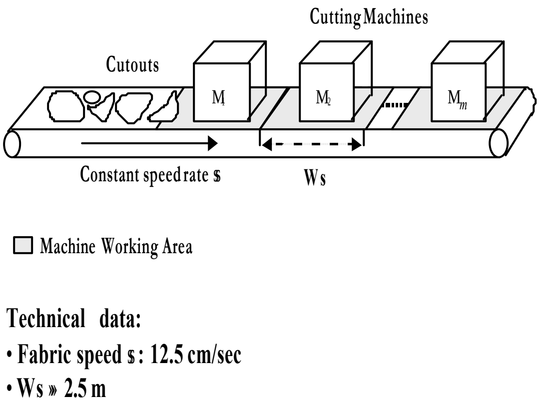

Fig. 1. The cutting department.

is the machine horizontal workspace, the same for all the machines.

Cutouts are grouped for cutting in batches. Let $B = \{B_1, \ldots, B_q\}$ be a set of batches to be produced in the production period, B is not known at the beginning. In fact, a batch B_h is available for cutting at a release date. Before this time, all the information related to the batch B_h (such as the number of cutouts of the batch, their spatial configuration on the fabric, cutting and set–up times, etc...) are not available. Let n_h be the number of cutouts of B_h ($n_h \simeq 50$), and denote with pjh, r^1_{jh} and D^1_{jh} the cutting time, the release date and the deadline of the cutout f_j of the batch B_h on the first machine. In Figure 2, a batch in which $n_h = 55$ is reported.

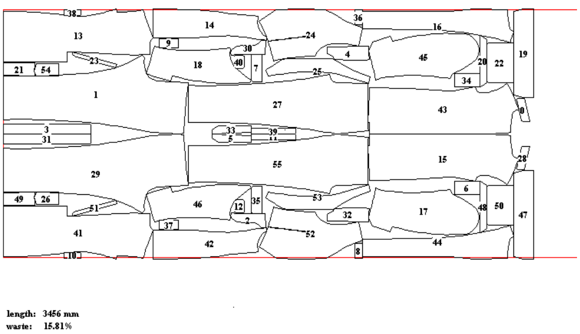

Fig. 2. A batch of 55 cutouts.

A set–up time occurs on a machine, when switching from one cutout to another, corresponding to the time required by the cutter to move from the previous to the next assigned cutout. Let $s_{jh,kp}$ be the set–up time required by a machine for moving from the cutout j of the batch h to the cutout k of the batch p (possibly $h = p$). Cutouts must be processed by only one cutting machine within their time windows.

The problem (referred as ASP) consists in assigning and sequencing cutouts of each batch to the cutting machines, in such a way that each cutout is assigned for cutting to exactly one machine, within the respective time window. Hence, if the cutout f_{jh} is processed by machine M_i, denoting with t_{jh} its starting processing time, the time windows constraints can be formulated as $r^i_{jh} \leq t_{jh} + p_{jh} \leq D^i_{jh}$. The objectives are minimizing the number of machines required for cutting and balancing the machines workloads. Note that release dates and deadlines depend on both the cutout and the machines.

Since a limited amount of time is available for assigning and scheduling a batch from its release date, fast heuristics algorithms, based on different assigning and scheduling rules, are proposed.

In Section 2, a problem formulation and some related literature results are reported. The basic steps of the algorithms for assigning and sequencing cutouts to the cutting machines are presented in Section 3.

2. PROBLEM FORMULATION AND RELATED RESULTS

The problem of assigning and sequencing cutouts can be formulated as a parallel machine scheduling problem.

In a parallel machine scheduling problem, a set of independent jobs, $J_1, \ldots, J_n$, have to be processed on m parallel machines, $M_1, \ldots, M_m$. Machines can handle at most one job at the time, and, if preemption is not allowed, each job can be executed on at most one machine at time without interruption.

In fact, in the ASP problem a batch of cutouts corresponds to a set of jobs that must be assigned for cutting to exactly one cutting machine.

A related problem, is the machine scheduling problem with real time constraints, in which every job asks to be scheduled for a given duration between a release date and a deadline. If the request is accepted by a machine, a given benefit is obtained. The objective is to maximize the benefit obtained from the scheduled jobs. This is an old NP–hard scheduling problem (Garey and Johnson, 1979).

Very recently the first constant approximation algorithms has been proposed for this problem (Bar–Noy *et al*, 1999) both on single and parallel machines. In general, the time windows in which a job can be scheduled may be much larger than its processing time. Arkin and Silverberg (1998) showed that the special case where there is no slack time can be solved optimally in polynomial time, even for multiple machines.

Another special case of this problem, that was considered earlier in the literature is the case in which all jobs are released at the same time (or, similarly, the case in which all deadlines are the same). This special case remains NP–hard even for a single machine (in which it reduces to minimizing the weighted number of tardy jobs). However Sahni (1976) gave a fully polynomial approximation scheme for this case.

In the ASP problem, the release date and the deadline of each cutout, i.e., the job real time constraints, can assume general values. However, ASP substantially differs from the machine scheduling problem with real time constraints for the following aspects:

- Batches information are not available at the beginning of the production period.
- A set–up time occurs on a cutting machine, when switching from one cutout to another (i.e., a cutout scheduling on each machine is required).

In the following Section, heuristic algorithms for assigning and sequencing cutouts to the machine are presented.

3. ASSIGNING AND SEQUENCING ALGORITHMS BASIC STEPS

Since short computational times are required for scheduling cutouts of a batch, when a new batch is released for cutting, cutouts of the previous batches are not involved in the assignment and sequencing of the current batch.

All the proposed algorithms are based on the following two phases:

ph_1 Cutout assignment.
ph_2 Cutout sequencing.

In the *Cutout assignment* phase, cutouts of a batch B_h are iteratively assigned to a machine. Two different approaches, called *push* and *pull* respectively, for the assignment are proposed.

In both the approaches, the assignment is iteratively performed by selecting a not yet assigned cutout and a cutting machine able to process it. In the push strategy, first a cutout f_{jh} is selected according to some rule, and then a cutting machine is chosen. On the other hand, in the pull strategy, first a machine M_i is selected, and then, machine M_i *chooses* a cutout.

Let S_i be the set of the cutouts currently assigned to machine M_i. The completion time C_i of machine M_i is defined as the time required by M_i for processing all the cutouts in S_i. The working time W_i is defined as $\sum_{j \in S_i} p_{jh}$. Let t^i_{jh} be the starting time of the cutout f_{jh} on the machine M_i.

The following cutout/machine selection rules are employed by the algorithms:

- cutout selection;
  - LPT select the cutout f_{jh} with the longest processing time p_{jh};
  - EDD select the cutout f_{jh} with the earliest deadline D^1_{jh};
  - ERD select the cutout f_{jh} with the earliest release date r^1_{jh};
  - $waste_c$ given a machine M_i, let f_{kp} be the last cutout sequenced on it. Select the cutout f_{jh}, not yet assigned, such that $t^i_{jh} - t^i_{kp} - p_{kp}$ is minimum.
- machine selection;
  - $\min C_i$ select the machine with the minimum completion time;
  - $\min W_i$ select the machine with the minimum working time;
  - $waste_m$ select the machine M_i such that $t^i_{jh} - t^i_{kp} - p_{kp}$ is minimum, in which f_{kp} is the last cutout sequenced on M_i, and f_{jh} is a given cutout not yet assigned.

Note that, the selection rule $waste_c$ ($waste_m$), given a machine M_i (cutout f_j), selects the cutout f_j (machine M_i) that minimizes the difference between the completion time of the last scheduled cutout on M_i and the starting time of f_j.

In the *Cutout sequencing* phase, the selected cutout is scheduled on the assigned machine. The following rules are performed, in this phase:

s_1 schedule cutout in EDD order;
s_2 schedule cutout at the end of the machine cutout sequence.

After that all the cutouts of the current batch are assigned and scheduled, a cutout interchange procedure, based on a local search, is applied, on the cutouts of each machine. The aim is to minimize the overall set–up time. Only cutouts of the current batch are involved in this procedure.

In the assignment phase, in order to find a feasible cutout assignment (i.e., in such a way that the cutout time window constraints are respected), set–up times are considered by increasing the cutout processing times of an *average* set–up time.

Different heuristics have been developed based on different assignment and sequencing rules. Since phases ph_1 and ph_2 require a fixed machine number m, in order to minimize the overall number of the machines, multiple executions, with different values of m, of the two phases are performed.

In the next section, a computational study on a set of real life instances is presented, in which both the quality of the solution and the computational times of the proposed algorithms are compared.

4. COMPUTATIONAL EXPERIENCE

On the basis of the rules reported in Section 3, different push and pull algorithms have been developed and tested on real instances. In particular, five push and three pull heuristics have been considered. In Tables 1 and 2, the details of the push and the pull algorithms are respectively reported.

Table 1. The push algorithms.

Name	Cutout selection	Machine selection	Sequencing rule
Ps1	LPT	$\min W_i$	s_1
Ps2	EDD	$waste_m$	s_2
Ps3	EDD	$\min C_i$	s_2
Ps4	EDD	$\min W_i$	s_2
Ps5	ERD	$\min W_i$	s_1

Table 2. The Pull algorithms.

Name	Machine selection	Cutout selection	Sequencing rule
Pull 1	$\min C_i$	EDD	s_2
Pull 2	$\min C_i$	$waste_c$	s_2
Pull 3	$\min C_i$	ERD	s_2

The computational results are related on two real life instances referred in the following as R_1 and R_2. The first instance is composed of 12 batches and 624 cutouts, while instance R_2 is composed of 896 cutout grouped in 16 batches.

Let $B = \{B_1, \ldots, B_q\}$ be a set of q batches and n cutouts, and let $f_j \in B$ and $B_h \in B$ be respectively the cutout j and the batch h of the set B.

For testing the performances of the heuristics, a lower bound on the number of the machines is computed off-line (i.e., assuming all instance data known at the beginning of the production period) as follows. Let $r^1_{min} = \min_{f_j \in B}\{r^1_j\}$ and $D^1_{max} = \max_{f_j \in B}\{D^1_j\}$. $D^1_{max} - r^1_{min}$ is an upper bound on the maximum machine completion time on the set B. A lower bound on the minimum number of the cutting machines is then:

$$LB = \left\lceil \frac{\sum_{j \in B}(p_j + (\min_{k \in B, k \neq j} s_{jk}))}{D^1_{max} - r^1_{min}} \right\rceil \quad (1)$$

Table 3. The Push algorithms on instance R_1.

	Ps1	Ps2	Ps3	Ps4	Ps5
$\bar{E}_{min}$	1.73	7.03	5.92	6.21	1.44
$\bar{E}$	10.11	11.73	13.55	11.42	9.87
$\bar{C}_{max}$	31.47	28.22	28.39	29.25	30.86
$\bar{C}$	27.25	27.06	27.03	27.06	27.2
T	331.54	325.23	325.4	326.51	328.23
m	6	6	6	6	6
sec.	<0.1	<0.1	<0.1	<0.1	<0.1

Table 4. The Pull algorithms on instance R_1.

	Pull 1	Pull 2	Pull 3
$\bar{E}_{min}$	6.89	7.47	5.92
$\bar{E}$	11.26	14.92	13.55
$\bar{C}_{max}$	28.55	29.13	28.39
$\bar{C}$	27.08	26.97	27.03
T	325.23	325.23	325.4
m	6	7	6
sec.	<0.1	<0.1	<0.1

In Tables 3,4 and 5,6 the computational results related to the instances R_1 and R_2 are respectively reported. For each heuristic, the following performance indexes are reported. Let $E_j = D^k_j - (t^k_j + p_j)$ be the earliness of the cutout f_j, $f_j \in B$, where M_k is the machine on which cutout f_j is processed. We indicate with $\bar{E}_{min} = (\sum_{B_h \in B} \min_{f_j \in B_h}\{E_j\})/q$ the average minimum earliness of the instance, and with $\bar{E}$ the average earliness of the instance (i.e., $\bar{E} = \sum_{f_j \in B} E_j/n$).

Moreover, let C^h_i be the machine completion time of machine M_i related to the cutouts of the batches $\{B_1, \ldots, B_h\}$, and let $\Delta C^h_i = C^h_i - C^{h-1}_i$, i.e., the amount of time required for cutting the cutouts of the batch B_h on machine M_i . Let $C^h_{max} = \max_{i=1,\ldots,m} \Delta C^h_i$, and let $\bar{C}_{max}$ be the average maximum time for cutting a batch, i.e., $\bar{C}_{max} = \sum_{B_h \in B} \Delta C^h_{max}/q$, and $\bar{C}$ be the average machine completion time related to a batch (i.e., $\bar{C} = \sum_{i=1}^{m}(\sum_{B_h \in B} C^h_i/q)/m)$.

In Tables 3–6, in lines 2 and 3 $\bar{E}_{min}$ and $\bar{E}$ are respectively reported, and in lines 4 and 5, the machine completion times $\bar{C}_{max}$ and $\bar{C}$ are contained.

Note that, for the problem under study, a cutout schedule with bigger values of $\bar{E}_{min}$ and $\bar{E}$ will be in general preferred. This is due to the fact bigger is the earliness of a cutout more is the available time to process it, and, hence, to manage system failures.

T, in line 6, is the total time required for processing all the cutouts of the instance. Finally in line seven, the computational time of each algorithm is reported.

Table 5. The Push algorithms on instance R_2.

	Ps1	Ps2	Ps3	Ps4	Ps5
$\bar{E}_{min}$	1.27	6.46	6.24	5.36	0.95
$\bar{E}$	10.7	12.41	14.16	12.1	10.46
$\bar{C}_{max}$	34.99	29.26	29.37	32.36	34.7
$\bar{C}$	28.01	27.91	27.87	27.91	28.06
T	453	447.7	447.07	450.45	454.24
m	7	7	7	7	7
sec.	<0.1	<0.1	<0.1	<0.1	<0.1

Table 6. The Pull algorithms on instance R_2.

	Pull 1	Pull 2	Pull 3
$\bar{E}_{min}$	6.34	6.17	6.24
$\bar{E}$	11.44	14.34	14.16
$\bar{C}_{max}$	29.45	30.11	29.37
$\bar{C}$	27.94	27.85	27.87
T	448.16	446.25	447.07
m	7	7	7
sec.	<0.1	<0.1	<0.1

The lower bound computed as in (1), both for R_1 and R_2, is of six machines. On instance R_1, almost all push and pull algorithms found a solution requiring the minimum machine number (excepted **Pull 2** that provides a solution with seven machines), while on instance R_2 all the algorithms require seven machines.

Among the push heuristics **Ps2** and **Ps3** have in general a good behavior both for the earliness and the machine completion time indexes. The pull algorithms have similar performances.

Computational times are small in according to the problem requirements.

REFERENCES

Arkin, E.M. and Silverberg, E.B. (1998). Scheduling jobs with fixed start and end times *Discrete Applied Mathematics*, 18, 1–8.

Bar–Noy, A., Guha, S., Naor, J., and Schieber, B. (1999). Approximation the throughput of multiple machines under real–time scheduling. In *Proceedings of the thirty–first annual ACM symposium on theory of computing*, 622–631, Atlanta.

Garey, M.R., Johnson, D.S. (1979). Computers and intractability - A guide to the theory of NP–Completeness. *Freeman* San Francisco.

Sahni, S. (1976). Algorithms for scheduling independent tasks *Journal of ACM*, 23, 116–127.