When flow shop scheduling meets dominoes, eulerian and hamiltonian paths

Jean-Charles Billaut[1], Federico Della Croce[2], Fabio Salassa[2], Vincent T'kindt[1]

[1] Université François-Rabelais de Tours, ERL CNRS OC 6305, 37200 Tours, France
{billaut,tkindt}@univ-tours.fr
[2] Politecnico di Torino, 10129 Torino, Italy
{federico.dellacroce,fabio.salassa}@polito.it

Mots clés : *Flow shop scheduling, no-idle, no-wait, dominoes.*

1 Introduction

We consider the no-idle/no-wait two-stage flow shop according to the following specifications. There is a set of n jobs available at time zero ; each job j must be processed non-preemptively on two continuously available machines M_1, M_2 with known integer processing times a_j, b_j, respectively. The order of processing is $M_1 \rightarrow M_2$ for all jobs. Each machine can process at most one job at a time and the operations of each job cannot overlap. Also, for any given sequence, $[j]$ denotes the job in position j. We will focus primarily on the makespan as performance measure. Using the general three-field notation [4], the related two-machine flow shop problem is denoted by $F2|no-idle, no-wait|C_{\max}$. In [1], it is mentioned that both problems $F2|no-idle|\sum C_j$ and $F2|no-wait|\sum C_j$ are NP-hard. Similar consideration holds for problem $F2|no-idle, no-wait|\sum C_j$. The recent literature on $no-wait$ flow shop scheduling includes [3] where it is shown that minimizing the number of interruptions on any machine is polynomially solvable on two machines and NP-hard on three or more machines.

Notice that the $no-idle, no-wait$ requirement is very strong as it forces consecutive jobs to share common processing times. As an example, any feasible solution for $F2|no-idle, no-wait|C_{\max}$, requires that $\forall i \in \ ..., n-1$, $b_{[i]} = a_{[i+1]}$. Figure 1 provides an illustrative example of a feasible no-idle, no-wait schedule for a two-machine flow shop with four jobs.

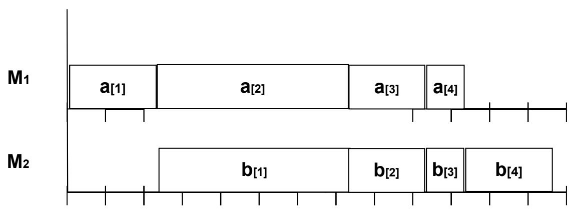

FIG. 1 – A no-idle no-wait schedule for a 2-machine flow shop

2 Main result

The peculiarity of the no-idle, no-wait effect strictly links the above mentioned flow shop problem to the game of dominoes. Dominoes are 1 x 2 rectangular tiles with each 1 x 1 square marked with spots indicating a number. A traditional set of dominoes consists of all 28 unordered pairs of numbers between 0 and 6. We refer here to the generalization of dominoes presented in [2] in which n tiles are present, each of the tiles can have any integer (or symbol) on each end and not necessarily all pairs of numbers are present. In [2], it is shown that the Single Player Dominoes (SPD) problem, where a single player tries to lay down all dominoes in a chain with the numbers matching at each adjacency, is polynomially solvable as it can be seen as the solution of a eulerian path problem on an undirected multigraph. Figure 2 shows the solution of an SPD problem with 12 tiles with numbers included between 0 and 6.

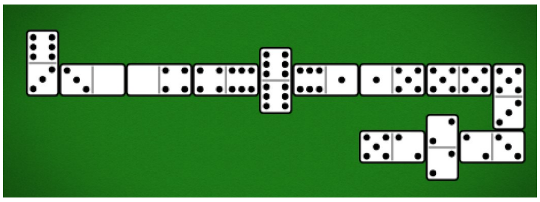

FIG. 2 – Solution of an SDD problem with 12 dominoes

We refer here to the oriented version of SPD called $OSPD$ where all dominoes have an orientation, e.g. if the numbers are i and j, only the orientation $i \to j$ is allowed but not viceversa. It is easy to show that also the OSDD problem is polynomially solvable as it can be seen as the solution of a eulerian path problem on a directed multigraph. The following proposition holds.

Proposition 1 *Problems $F2|no-idle, no-wait|C_{\max}$ and $OSDD$ are equivalent. Correspondingly, problem $F2|no-idle, no-wait|C_{\max}$ is polynomially solvable.*

The no-idle, no-wait 2-machine flow shop problem is also linked to a special case of the Hamiltonian Path problem. Consider a digraph $G(V, A)$ that has the following property : if $S_i \cap S_j \neq \emptyset$ then $S_i = S_j$ where we denote by S_i the set of successors of vertex i. In other words, each pair of vertices either has no common successors or has all successors in common. Let indicate the Hamiltonian path problem in that graph as the Common Successors Hamiltonian path (CSHP) problem. The following proposition holds.

Proposition 2 *$CSHP \propto F2|no-idle, no-wait|C_{\max}$. Correspondingly, problem $CSHP$ is polynomially solvable.*

By exploiting Proposition 2, the following proposition holds.

Proposition 3 *Problem $F|no-idle, no-wait|C_{\max}$ is polynomially solvable.*

Références

[1] I. Adiri and D. Pohoryles. Flowshop / no-idle or no-wait scheduling to minimize the sum of completion times. *Naval Research Logistics*, 29 : 495–504, 1982.

[2] Erik D. Demaine, Fermi Ma, and Erik Waingarten. Playing Dominoes Is Hard, Except by Yourself. *FUN 2014, LNCS*, 8496 : 137–146, 2014.

[3] Wiebke Höhn, Tobias Jacobs, Nicole Megow. On Eulerian extensions and their application to no-wait flowshop scheduling. *Journal of Scheduling*, 15 : 295–309, 2012.

[4] Lawler E.L., J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys (1993), "Sequencing and Scheduling : Algorithms and Complexity" in *S.C. Graves, A.H.G. Rinnooy Kan and P. Zipkin (Eds.) : Handbooks in Operations Research and Management Science vol 4 : Logistics of Production and inventory, North-Holland, Amsterdam.*