



HAL
open science

When shop scheduling meets dominoes, eulerian and hamiltonian paths

Jean-Charles Billaut, Federico Della Croce, Fabio Salassa, Vincent t'Kindt

► **To cite this version:**

Jean-Charles Billaut, Federico Della Croce, Fabio Salassa, Vincent t'Kindt. When shop scheduling meets dominoes, eulerian and hamiltonian paths. 13th Workshop on Models and Algorithms for Planning and Scheduling Problems (MAPSP 2017), Jun 2017, Seon-Seebruck, Germany. hal-01703187

HAL Id: hal-01703187

<https://hal.science/hal-01703187v1>

Submitted on 7 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

When shop scheduling meets dominoes, eulerian and hamiltonian paths

Jean-Charles Billaut *

Federico Della Croce †

Fabio Salassa ‡

Vincent T'kindt §

1 Introduction

In this paper, we consider no-idle and no-wait shop scheduling problems with some standard configurations namely flow shop, job shop and open shop. We focus on the makespan, referred to as C_{\max} , as a performance measure. More precisely, we tackle four distinct problems which, using the general three-field notation [4], can be denoted as $F2|no-idle, no-wait|C_{\max}$ for 2-machine the flow shop, $J2|no-idle, no-wait|C_{\max}$ for the for 2-machine job shop and $O2|no-idle, no-wait|C_{\max}$ for the for 2-machine open shop. The m-machine flow shop problem is denoted as $F|no-idle, no-wait|C_{\max}$. We refer first to problem $F2|no-idle, no-wait|C_{\max}$: n jobs are available at time zero; each job j must be processed non-preemptively on two continuously available machines M_1, M_2 with integer processing times $p_{1,j}, p_{2,j}$, respectively. The processing order is $M_1 \rightarrow M_2$ for all jobs. Each machine processes at most one job at a time and operations of each job cannot overlap. For any sequence, $[j]$ denotes the job in position j . The objective is the minimization of the makespan. For the 2-machine jobshop problem, each job is required to strictly follow its own specific processing order and for the 2-machine openshop problem, the operations of a job can be processed in any order.

With respect to the literature, to the best of our knowledge, few works have dealt with the no-idle and no-wait constraints simultaneously. 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 relevant literature includes [3] where it is shown that minimizing the number of interruptions on the last machine is solvable in $O(n^2)$ time on two machines (the problem is denoted as $F2|no-wait|\mathcal{G}$) while it is NP -hard on three machines or more. We remark that problems $F2|no-wait|\mathcal{G}$ and $F2|no-idle, no-wait|C_{\max}$, even though close are not equivalent and an optimal solution with no interruption of problem $F2|no-wait|\mathcal{G}$ may be non-optimal for problem $F2|no-idle, no-wait|C_{\max}$. Consider a 2-job instance with processing times $p_{1,1} = i$, $p_{2,1} = i$, $p_{1,2} = i$, $p_{2,2} = j$, with $j > i$. Then, sequence 1-2 is no-idle, no-wait,

*billaut@univ-tours.fr. Université Francois-Rabelais de Tours, Laboratoire d'Informatique (EA 6300), ERL CNRS OC 6305, 64 avenue Jean Portalis, 37200 Tours, France.

†federico.dellacroce@polito.it. Corso Duca degli Abruzzi, 24, 10129 Torino, Italy.

‡fabio.salassa@polito.it. Corso Duca degli Abruzzi, 24, 10129 Torino, Italy.

§tkindt@univ-tours.fr. Université Francois-Rabelais de Tours, Laboratoire d'Informatique (EA 6300), ERL CNRS OC 6305, 64 avenue Jean Portalis, 37200 Tours, France.

has makespan $C_{\max}^{1-2} = 2j + i$ and is optimal for problem $F2|no - wait|\mathcal{G}$ as it has no interruption. However, it is not optimal for problem $F2|no - idle, no - wait|C_{\max}$ as sequence 2 - 1 is also no-idle, no-wait and has makespan $C_{\max}^{2-1} = 2i + j < 2j + i$.

In the next section we present some complexity results related to the four tackled shop scheduling problems.

2 Main results

We first point out that the *no - idle, no - wait* constraint is very strong as it forces consecutive jobs to share common processing times, namely, any feasible solution for $F2|no - idle, no - wait|C_{\max}$, requires that $\forall j \in \dots, n - 1, p_{2,[j]} = p_{1,[j+1]}$. Figure 1 provides an example of a feasible no-idle, no-wait schedule for a 2-machine flow shop with four jobs.

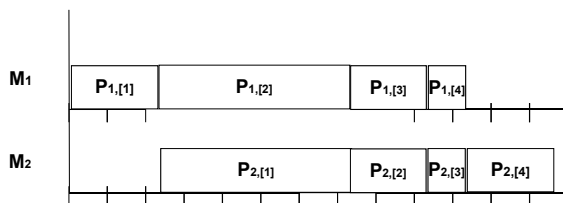


Figure 1: A no-idle no-wait schedule for a 2-machine flow shop

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 an 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.

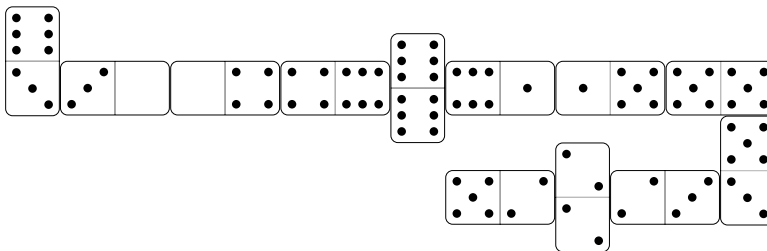


Figure 2: Solution of an SPD problem with 12 dominoes

We refer 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 $i \rightarrow j$ is allowed but not viceversa. It is easy to show that also problem *OSPD* is polynomially solvable as it can be seen as the solution of an eulerian path problem on a directed multigraph.

The following proposition holds.

Proposition 1 $F2|no-idle, no-wait|C_{\max} \propto OSPD$ and can be solved in $O(n)$ time.

Problem $F2|no-idle, no-wait|C_{\max}$ is also linked to a special case of the Hamiltonian Path problem on a connected digraph. Consider a digraph $G(V, A)$ that has the following property: $\forall v_i, v_j \in V$, either $S_i \cap S_j = \emptyset$, or $S_i = S_j$ where we denote by S_i the set of successors of vertex v_i . In other words, each pair of vertices either has no common successor or has all successors in common. Let indicate the Hamiltonian path problem in that graph as the Common/Distinct Successors Hamiltonian Directed Path (CDSHDP) problem.

The following proposition holds.

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

The analysis leading to Proposition 1 can be extended also to the m -machine case. The following proposition holds.

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

Finally, by reduction from the Numerical Matching with Target Sums (NMTS) problem, the following proposition holds.

Proposition 4 Problems $J2|no-idle, no-wait|C_{\max}$ and $O2|no-idle, no-wait|C_{\max}$ are NP-Hard in the strong sense.

References

- [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 Hohn, 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.*