

Addressing idle and waiting time in short term production planning

*Original*

Addressing idle and waiting time in short term production planning / Pastore, E., Alfieri, A., Castiglione, C.. - 35:(2023), pp. 10-18. (XVI Convegno AITEM Napoli, 13-15 settembre 2023 Napoli, Italia 13/15 settembre 2023) [10.21741/9781644902714-2].

*Availability:*

This version is available at: 11583/2981784 since: 2023-11-14T16:58:36Z

*Publisher:*

Material Research Forum

*Published*

DOI:10.21741/9781644902714-2

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

## Addressing idle and waiting time in short term production planning

Erica Pastore<sup>1,a\*</sup>, Arianna Alfieri<sup>1,b</sup> and Claudio Castiglione<sup>1,c</sup>

<sup>1</sup>Department of Management and Industrial Engineering, Politecnico di Torino, Torino, Italy

<sup>a</sup>erica.pastore@polito.it, <sup>b</sup>arianna.alfieri@polito.it, <sup>c</sup>claudio.castiglione@polito.it

**Keywords:** Production Planning, Scheduling, Flowshop

**Abstract.** Production systems are facing the increase of economic and sustainability challenges in managing production resources, demand variability and variety, and the increasing shortage of materials. Thus, short-term production planning must include several aspects and consider multiple objective functions simultaneously. In this context, controlling and optimizing waiting and idle times might lead to various benefits, as they are among the main cost sources in production systems and can affect the feasibility of operations from a technological perspective. While waiting time is related to the work in process, idle time refers to a low utilization rate, and both may generate inefficiency and costs. This paper studies how different emphasis to waiting and/or idle time can affect the solution of short-term production planning with several industrially relevant objectives.

### Introduction

Short-term production planning deals with scheduling jobs to be produced in the shop floor to optimize one or more criteria. These operations might consider technological constraints related to the jobs to be produced, the processes, and the production resources; optimization criteria are related to costs and challenges the company faces. Among these costs, those related to idle and waiting times are largely important [1]. Long waiting times usually transfer in high work in process (WIP) levels, with consequent high inventory costs and low service level. Instead, long idle times usually imply low utilization rate, possibly due to resource over-sizing, and related costs.

Apart from costs, some industries and technological processes might avoid idle and/or waiting times. For instance, temperature or other characteristics of the materials might require that each operation immediately follows the previous one, thus not allowing any waiting time (*no-wait*) [2]. Similarly, resources might use materials or consumables that become unusable if the machine stays idle for too long (e.g., paint can dry); in such cases, idle time is not allowed (*no-idle*). In addition, interrupting some processes may generate high costs, thus limiting the number of interruptions on some stages can be beneficial, i.e., limiting the occurrences for the machine to move from busy to idle and vice-versa [3]. This occurs, for instance, in casting processes where interrupting a continuous production flow implies maintenance and extensive cleaning, which in turn causes extra costs and delays in the entire production [4].

Although the literature on short-term production addressing idle and waiting time is vast, usually such problems are addressed from the algorithmic point of view. On the contrary, the objective of this paper is to study how different approaches (i.e., different ways of addressing idle and waiting time) can affect the solution of short-term production planning with several industrially relevant objectives. The industrial/managerial implications are the focus of the paper, while the algorithmic side is out of the scope of the paper. The paper studies a permutation flow shop (PFS) production system (i.e., jobs undertake a set of operations on a set of workstations with the same order, and the job sequence on each machine is the same).

### Literature Review

The short-term planning of permutation flow shops has been widely studied in the literature. Many optimization criteria have been addressed, such as makespan, total flow time, total tardiness and

so on [e.g. 5,6]. As idle and waiting times are the focus of the paper, in the following, only the literature addressing these two performance measures is reviewed.

Most of the literature on short-term production planning addressing idle and waiting time focuses on developing heuristic and meta-heuristic algorithms to solve various problem variants; however, as this paper aims at investigating idle time at the industrial level, the reader is referred to [1,7,8,9] for reviews of the solution methods available in the literature.

When idle time is considered, depending on the technological processes of the shop floor and on the criteria to optimize, it has been assumed either to be avoided (totally or in part) or to be minimized. The *no-idle* PFS scheduling problem is NP-hard, and due to its complexity, many authors developed heuristic and meta-heuristic algorithms to solve it. This problem arises in many real-life production systems such as in the production of integrated circuits, where the costs of steppers are so high that idle time is not desired [7]; also, in fiberglass production and in foundries, some machines (e.g., furnaces, casting machines, ceramic roller kilns) cannot be easily turned off and restarted due to the long machine setup times [10]. Sometimes, only some of the workstations composing the flow shop must respect the no-idle conditions. For instance, in ceramic frit production, only the central fusing kiln has the no-idle constraint; also, in steel production, only the final casting phase needs not to stop (i.e., to be no-idle) while the previous operations can admit idle time [11]. In other cases, idle time can be allowed, but it is linked to higher costs (i.e., there is no technological constraint to avoid idle time); in these cases, some authors have addressed the problem of minimizing the total amount of idle time in the production system [1], or the problem of minimizing the number of interruptions [3]. In such cases, obviously, machines can be idle.

When waiting time is considered, depending on the technological process, jobs can be required not to wait between two consecutive operations. For instance, in steel manufacturing, after being heated to a specific temperature, hot slabs cannot wait before rolling operations; otherwise, their temperature would significantly drop [12]. In general, all the manufacturing processes that require the WIP to be pre-heated to a high temperature may need no-wait conditions [13,14]. Also, robotic cells, which provide a highly coordinated manufacturing process, need to avoid waiting time between consecutive operations [15]. In some cases, avoiding waiting time is an efficient strategy to reduce WIP-related costs [1]. In such cases, the no-wait condition may be downgraded to the minimization of the total waiting time of jobs in the system [1,16], which implies the possibility that jobs wait between consecutive operations to optimize some other performance measure.

### Problem formulation

This paper considers a flow shop production system. In this system, there is a set of  $J$  jobs to be processed, each requiring a set of  $M$  operations. Each operation is allocated to a single machine, and the order in which the operations have to be executed is the same for each job. Thus, the sequence of the operations is the same as the order of the machines, as depicted in Fig. 1. In the system, machines cannot perform more than one operation simultaneously, and jobs cannot be processed by more than one machine simultaneously. The operations, once started, cannot be interrupted (*non-preemption* assumption). For simplicity, a permutation flow shop is considered, in which the sequence of jobs is the same on each machine. Various problem variants are studied according to the assumptions made on idle and waiting time and on the performance measure.

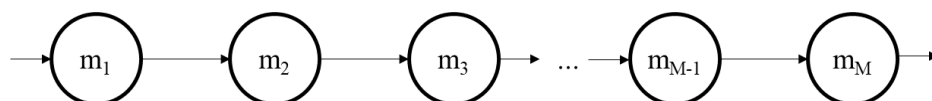


Fig. 1. Flow shop layout

Specifically, the paper addresses the following systems:

- flow shop with allowed idle and waiting times (*general*),
- flow shop with no-idle constraint (*no-idle*),
- flow shop with no-wait constraint (*no-wait*).

For each system, four solutions are found, each minimizing one among the following objective functions (OF):

- *makespan* ( $C_{max}$ ), i.e., the completion time of the last job on the last machine;
- *total completion time* ( $C_{tot}$ ), i.e., the sum of the completion times of all jobs in the sequence;
- *total core idle time* ( $C_{it}$ ), i.e., the sum of idle times in all machines between the start of the first job of the sequence and the end of the last job;
- *total core waiting time* ( $C_{wt}$ ), i.e., the sum of waiting times of all jobs between the first and the last operations.

In general, for any problem variant (i.e., for each combination system-OF), the aim is to decide how to process jobs (i.e., to find the job schedule) to optimize the selected OF. As the problem variants are mostly NP-hard, sub-optimal solutions are found in this paper by using a constructive heuristic algorithm. Specifically, the well-known NEH heuristic is adapted to the problem variants. The developed algorithm has the same general structure for all the problem variants, and only limited changes in the sorting rule (as explained in the following) are made to adapt it to each of them. The algorithm works as in the following.

1. Jobs are sorted according to a specific rule depending on which OF is minimized. For makespan and total completion time minimization ( $NEH_{C_{max}}$ ,  $NEH_{C_{tot}}$ ), jobs are sorted according to the decreasing sum of processing times; for waiting time minimization ( $NEH_{C_{wt}}$ ), jobs are sorted according to the index defined by [17] (that accounts for the variability of processing times); for idle time minimization ( $NEH_{C_{it}}$ ), jobs are sorted according to the descending order of the index defined by [18] (that accounts for the variability, skewness and kurtosis of processing times).
2. Each job in the sorted list is inserted in the solution in the position that minimizes OF, thus originating the final schedule.

For each solution of each problem variant, all the OFs are evaluated and compared. The comparison aims at evaluating the differences among solutions (in terms of objective functions) found by modelling idle and waiting times in different ways.

## Numerical results

The aim of the experiment is to assess the impact of the way idle and waiting times are modelled on the solution for short-term production planning. To this aim, the NEH is used to find solutions for all the problem variants previously described; the solutions are then compared with respect to their evaluated performance measures ( $C_{max}$ ,  $C_{tot}$ ,  $C_{it}$ ,  $C_{wt}$ ). The experiment investigates the trade-off between imposing no-idle/no-wait conditions and paying idle/waiting time. This is particularly relevant for systems characterized by high idle and waiting time costs.

Design of experiment. The Taillard benchmark [19] is used to determine processing times of jobs on the machines. The number of jobs varies between 20 and 500, while the number of machines between 5 and 20. For each problem, 10 instances are available. For each instance, the three systems (general, no-idle, no-wait) are considered, and for each of them, four solutions are found by applying the NEH algorithm with different OFs ( $NEH_{C_{max}}$ ,  $NEH_{C_{tot}}$ ,  $NEH_{C_{it}}$ ,  $NEH_{C_{wt}}$ ); note that in no-idle systems  $NEH_{C_{it}}$  cannot be used, as well as  $NEH_{C_{wt}}$  is not used in no-wait systems. Overall, 4800 experiments are run. For each solution, all the OF values are computed.

Results. Fig. 2 shows, for some of the considered problems, mean values and confidence intervals of some performance measures. Specifically, Fig.2 (a) shows the average values of  $C_{max}$ ,

$C_{tot}$ ,  $C_{it}$ ,  $C_{wt}$  of the solutions of the problems with 100 jobs. As an example, starting from the left part of the graph, the grey line shows the average completion time of solutions for no-idle systems found by minimizing  $C_{max}$ ,  $C_{tot}$  and  $C_{wt}$ , respectively; the second grey line shows the same for no-wait systems, and the third for general systems. Instead, Fig. 2 (b) shows the interval plot of makespan for all the problems with 500 jobs, grouped by system variant and NEH OF. As the figures show, for each system, minimizing different objective functions leads to difference system performance. For instance, in general systems, minimizing the waiting time leads to larger  $C_{max}$  values, which, in turns, implies having low utilization levels of machines and lower production rates. Moreover, how idle and waiting times are modelled has an impact on the system performance measures. As Fig. 2 (a) shows, no-idle systems tend to have a larger waiting time than a general system in which idle time is minimized (i.e., general system –  $NEH_{C_{it}}$ ). Obviously, if machines cannot be idle because of technological constraints, the consequence of increasing waiting times cannot be avoided. However, if machines can be idle but the idle time related cost is high, then minimizing idle time instead of imposing a no-idle condition can turn into lower waiting time (and, hence, WIP-related) costs. In this case, an economic trade-off should be evaluated.

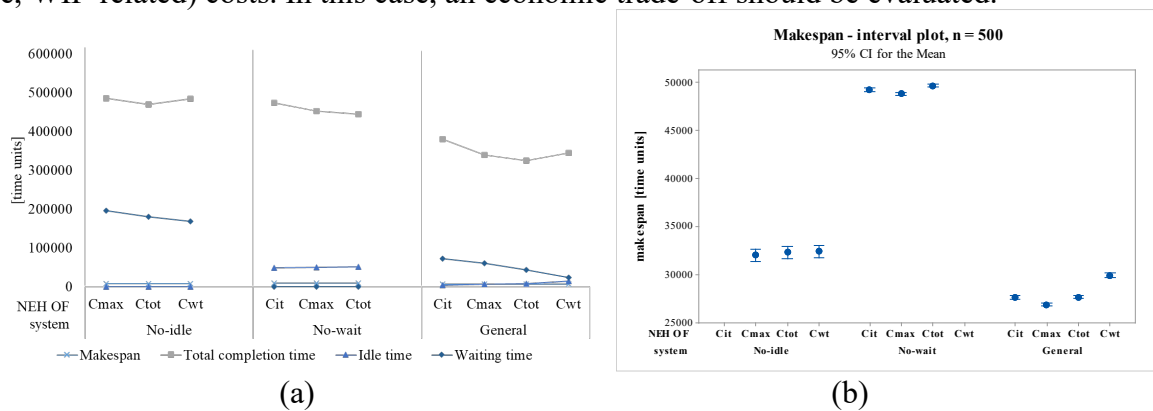


Fig. 2. (a) Average values of all performance measures for all problem variants with 100 jobs.  
 (b) 95% confidence intervals of makespan in all problem variants with 500 jobs.

Table 1 shows the average values of all the performance measures, grouped by system, NEH OF, number of jobs and machines. The table shows that, as expected, the larger the instance, the larger the values of each measure. Obviously, for each system, each evaluated performance measure has its lowest value in the solution that minimizes it: for instance, for general systems, the makespan is lower in the solutions of  $NEH_{C_{max}}$  than the one measured in the other solutions (the example refers to the rows with green borders of the table).

To assess the impact of the way idle and waiting times are modelled on the performance of the system, hypothesis tests are used for comparison. Starting from the *idle time*, it can be modelled either by imposing a no-idle condition, or by minimizing it as OF in a general system. The aim is to understand how the other performance measures ( $C_{max}$ ,  $C_{tot}$ ,  $C_{wt}$ ) change in these two cases. As an example, the blue rows of Table 1 display how makespan changes between a general system with  $NEH_{C_{it}}$  (denoted by  $C_{max_{Gen,Cit}}$ ) and a no-idle system with  $NEH_{C_{max}}$  (denoted by  $C_{max_{NI,Cmax}}$ ).

To evaluate the difference, the percentage difference  $\Delta_{C_{max}}$  is computed as:

$$\Delta_{C_{max}} = \frac{C_{max_{Gen,Cit}} - C_{max_{NI,Cmax}}}{C_{max_{Gen,Cit}}} \quad (1)$$

*Table 1. Average performance measures for all problem variants, grouped by system, NEH OF, number of jobs and of machines*

|         |                        | n. jobs = 20                         |       |       | 50    |        |        | 100    |        |        | 200     |         | 500      |
|---------|------------------------|--------------------------------------|-------|-------|-------|--------|--------|--------|--------|--------|---------|---------|----------|
|         |                        | 5                                    | 10    | 20    | 5     | 10     | 20     | 5      | 10     | 20     | 10      | 20      | 20       |
|         |                        | n. machines                          |       |       |       |        |        |        |        |        |         |         |          |
|         |                        | Makespan [time units]                |       |       |       |        |        |        |        |        |         |         |          |
| System  | NEH OF                 |                                      |       |       |       |        |        |        |        |        |         |         |          |
| No-idle | <i>C<sub>max</sub></i> | 1380                                 | 2007  | 3502  | 3015  | 3753   | 5877   | 5521   | 6887   | 9280   | 12174   | 15131   | 32032    |
|         | <i>C<sub>tot</sub></i> | 1413                                 | 1995  | 3564  | 3068  | 3774   | 5935   | 5650   | 6980   | 9309   | 12428   | 15313   | 32326    |
|         | <i>C<sub>wt</sub></i>  | 1492                                 | 2061  | 3548  | 3124  | 3903   | 6069   | 5750   | 7073   | 9412   | 12597   | 15527   | 32433    |
| No-wait | <i>C<sub>it</sub></i>  | 1494                                 | 2169  | 3302  | 3345  | 4578   | 6374   | 6381   | 8448   | 11361  | 16125   | 21225   | 49280    |
|         | <i>C<sub>max</sub></i> | 1433                                 | 2006  | 3043  | 3312  | 4386   | 6090   | 6344   | 8291   | 11081  | 15810   | 20770   | 48835    |
|         | <i>C<sub>tot</sub></i> | 1490                                 | 2092  | 3194  | 3440  | 4549   | 6263   | 6502   | 8585   | 11364  | 16114   | 21121   | 49696    |
| General | <i>C<sub>it</sub></i>  | 1402                                 | 1802  | 2516  | 2913  | 3349   | 4274   | 5433   | 6038   | 7007   | 11133   | 12269   | 27650    |
|         | <i>C<sub>max</sub></i> | 1261                                 | 1583  | 2319  | 2756  | 3135   | 3957   | 5272   | 5752   | 6634   | 10804   | 11753   | 26907    |
|         | <i>C<sub>tot</sub></i> | 1343                                 | 1695  | 2463  | 2889  | 3351   | 4120   | 5470   | 6039   | 6973   | 11106   | 12217   | 27688    |
|         | <i>C<sub>wt</sub></i>  | 1439                                 | 1814  | 2561  | 3127  | 3640   | 4440   | 5846   | 6512   | 7547   | 12055   | 13396   | 29959    |
|         |                        | Total completion time [time units]   |       |       |       |        |        |        |        |        |         |         |          |
| No-idle | <i>C<sub>max</sub></i> | 18308                                | 31712 | 60716 | 91810 | 130469 | 235129 | 310747 | 449960 | 697524 | 1446917 | 2089681 | 9931072  |
|         | <i>C<sub>tot</sub></i> | 17937                                | 30567 | 61106 | 87618 | 126870 | 233150 | 293060 | 435216 | 680733 | 1409108 | 2045126 | 9695973  |
|         | <i>C<sub>wt</sub></i>  | 19455                                | 31690 | 61326 | 92247 | 133463 | 240541 | 312290 | 449817 | 692068 | 1458294 | 2107398 | 9848739  |
| No-wait | <i>C<sub>it</sub></i>  | 17998                                | 26441 | 42571 | 95621 | 129115 | 184689 | 342539 | 452853 | 625838 | 1698083 | 2235262 | 12631925 |
|         | <i>C<sub>max</sub></i> | 16602                                | 24704 | 40486 | 87432 | 120392 | 176732 | 327895 | 430746 | 600214 | 1610098 | 2163203 | 12368030 |
|         | <i>C<sub>tot</sub></i> | 15670                                | 24252 | 39944 | 82172 | 118434 | 174832 | 302911 | 430587 | 599178 | 1586040 | 2179182 | 12578382 |
| General | <i>C<sub>it</sub></i>  | 18768                                | 25484 | 38795 | 90943 | 108210 | 147843 | 331945 | 365163 | 443562 | 1294577 | 1462361 | 7843045  |
|         | <i>C<sub>max</sub></i> | 15530                                | 22015 | 35586 | 77701 | 97664  | 134507 | 273004 | 328974 | 417138 | 1171820 | 1385930 | 7478628  |
|         | <i>C<sub>tot</sub></i> | 14607                                | 20945 | 34252 | 72579 | 92739  | 129430 | 264163 | 313461 | 398242 | 1129317 | 1314814 | 7173633  |
|         | <i>C<sub>wt</sub></i>  | 15472                                | 21687 | 34054 | 76418 | 97054  | 132443 | 284449 | 331726 | 417910 | 1202852 | 1403763 | 7603568  |
|         |                        | Total core idle time [time units]    |       |       |       |        |        |        |        |        |         |         |          |
| No-wait | <i>C<sub>it</sub></i>  | 1456                                 | 7258  | 25707 | 3362  | 16172  | 57887  | 6118   | 29888  | 109181 | 55958   | 203359  | 465365   |
|         | <i>C<sub>max</sub></i> | 1569                                 | 7503  | 26913 | 3702  | 16619  | 59145  | 6439   | 30527  | 109980 | 56117   | 204157  | 466044   |
|         | <i>C<sub>tot</sub></i> | 1665                                 | 7710  | 27848 | 4025  | 17508  | 60405  | 6829   | 32241  | 113159 | 58204   | 209402  | 481218   |
| General | <i>C<sub>it</sub></i>  | 140                                  | 953   | 4169  | 433   | 1312   | 6468   | 319    | 2377   | 8964   | 2707    | 11195   | 15356    |
|         | <i>C<sub>max</sub></i> | 420                                  | 2012  | 7134  | 660   | 3044   | 10961  | 707    | 3726   | 15108  | 4648    | 18806   | 22500    |
|         | <i>C<sub>tot</sub></i> | 625                                  | 2384  | 8401  | 970   | 3853   | 12511  | 1102   | 4978   | 17444  | 5874    | 22108   | 29321    |
|         | <i>C<sub>wt</sub></i>  | 1056                                 | 3488  | 10629 | 1894  | 6602   | 18320  | 2666   | 9374   | 28849  | 14480   | 44620   | 73747    |
|         |                        | Total core waiting time [time units] |       |       |       |        |        |        |        |        |         |         |          |
| No-idle | <i>C<sub>max</sub></i> | 4682                                 | 13360 | 32180 | 24955 | 49243  | 128670 | 62588  | 167571 | 357830 | 395643  | 920298  | 3420189  |
|         | <i>C<sub>tot</sub></i> | 4602                                 | 12842 | 32520 | 20391 | 46117  | 126908 | 45026  | 153468 | 343209 | 358420  | 872828  | 3155334  |
|         | <i>C<sub>wt</sub></i>  | 4154                                 | 12960 | 31769 | 15171 | 45069  | 128630 | 32280  | 136670 | 337244 | 305421  | 865828  | 2923396  |
| General | <i>C<sub>it</sub></i>  | 3880                                 | 7016  | 9717  | 13437 | 23579  | 40602  | 43084  | 67241  | 104676 | 175804  | 291344  | 1087660  |
|         | <i>C<sub>max</sub></i> | 2329                                 | 4421  | 6684  | 13030 | 17163  | 30105  | 39687  | 55440  | 86463  | 168936  | 262840  | 1177607  |
|         | <i>C<sub>tot</sub></i> | 1714                                 | 3556  | 5772  | 7460  | 12927  | 25641  | 19296  | 41601  | 67913  | 101469  | 182405  | 713631   |
|         | <i>C<sub>wt</sub></i>  | 899                                  | 2295  | 4195  | 2926  | 7309   | 16820  | 8203   | 19182  | 42068  | 45134   | 100114  | 340582   |

The tested hypothesis is  $\Delta_{Cmax} = 0$ . Over all the experiments, the average  $\Delta_{Cmax}$  is -0.1663, and the T-test of the tested hypothesis has a p-value equal to zero, thus the mean percentage difference cannot be considered equal to zero. In practice, the makespan of solutions of no-idle systems in which the  $NEH_{Cmax}$  is used ( $Cmax_{NI,Cmax}$ ) is larger than that of a general system that minimizes the idle time ( $Cmax_{Gen,Cit}$ ). This means that imposing a no-idle condition on one hand avoids idle time related costs but, on the other hand, it increases the utilization related costs. Fig. 3 graphically displays the trade-off between these two performance measures. If the no-idle systems (red diamonds) do not have any cost related to idle times, the costs related to the makespan are larger than the general systems (blue circles).

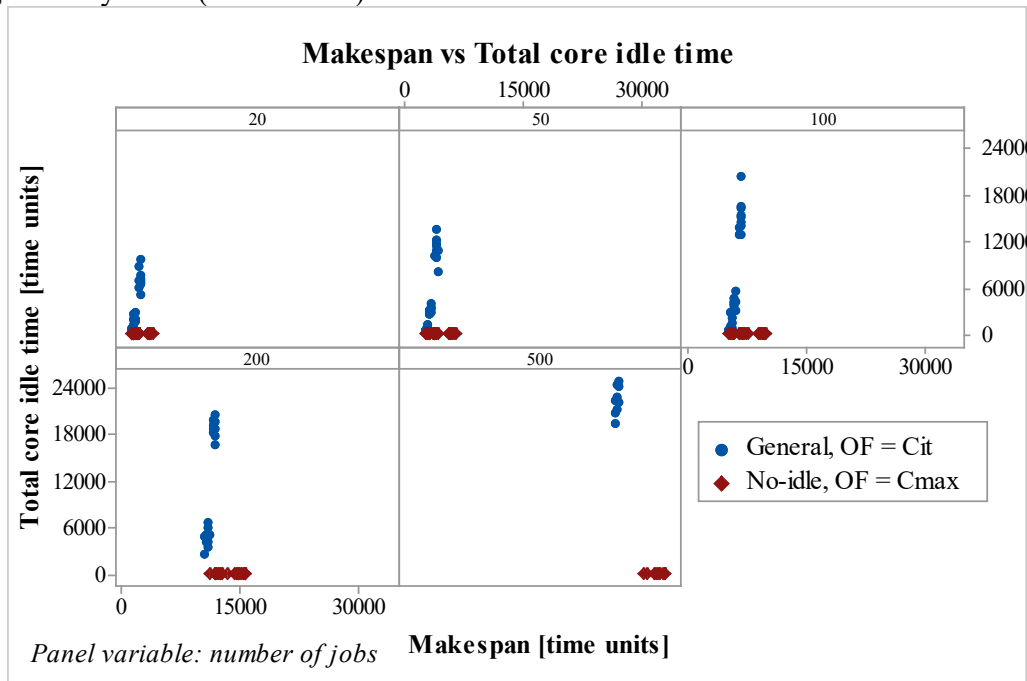


Fig. 3. Dispersion of  $cmax$  (x-axis) and  $cit$  (y-axis), grouped by number of jobs. Compared systems: general with  $NEH_{Cit}$  (blue circles), no-idle with  $NEH_{Cmax}$  (red diamonds).

The same analysis has been performed for total completion times and waiting times. In both cases, the difference between the two systems when such performance measures are considered is statistically significant.

The same comparison has been made to address the *waiting time* modelling. The considered systems are: general system with waiting time minimization ( $Gen, Cwt$ ), no-wait systems with their OFs ( $NW, Cmax - NW, Ctot - NW, Cit$ , alternatively). As an example, let consider the total completion time as the performance measure to be evaluated; then, the compared systems are the ones written with the purple color in Table 1. The percentage difference is computed as:

$$\Delta_{Ctot} = \frac{Ctot_{Gen,Cwt} - Ctot_{NW,Ctot}}{Ctot_{Gen,Cwt}} \quad (2)$$

Over all the experiments, the average  $\Delta_{Ctot}$  is equal to -0.2708, and the T-test for the hypothesis  $\Delta_{Ctot} = 0$  has a p-value equal to zero, thus the mean percentage difference differs from zero. In practice, minimizing the total completion time in a no-wait system leads to larger completion times than minimizing the total core waiting time in a general system. Fig. 4 graphically shows how the two measures are distributed.

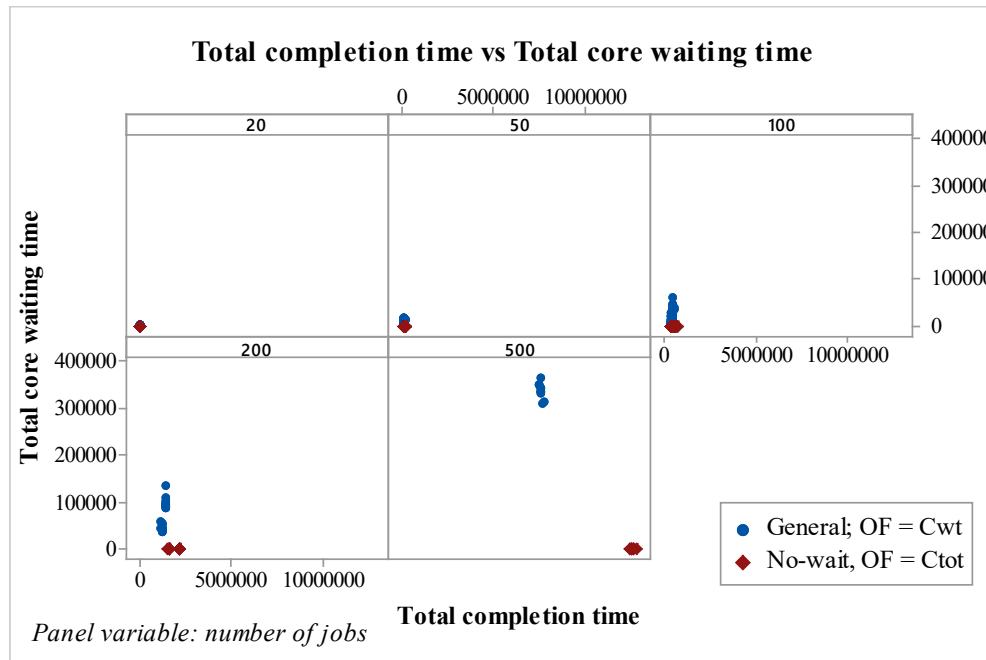


Fig. 4. Dispersion of  $C_{tot}$  (x-axis) and  $C_{wt}$  (y-axis), grouped by number of jobs. Compared systems: general with  $NEH_{C_{wt}}$  (blue circles), no-wait with  $NEH_{C_{tot}}$  (red diamonds).

As the figure shows, imposing the no-wait condition increases the total completion time, leading to larger costs related to flow time, WIP and service level reduction.

### Conclusions

Idle and waiting times are very relevant performance measures in production systems, as they are critical in some technological processes, and they both generate costs. In short-term production planning, they can be avoided by imposing no-idle and no-wait conditions, or they can be optimized to reduce them. This paper studies how different ways to model them can generate different schedules with different total costs, thus affecting the solution of short-term production planning. The addressed systems are permutation flow shops with and without no-idle/no-wait conditions, in which several performance measures are optimized. Numerical results on benchmark problems available in the literature show that the way idle and waiting times are modelled significantly impacts on other performance measures such as makespan and total completion time. These are in turns related to utilization, WIP, and flow time costs.

If the technological characteristics of the process or the materials impose no-idle/no-wait conditions, the other performance measures will suffer from these constraints, but no actions can be implemented to improve them. For all the other cases in which idle/waiting times can occur but with high costs, the economic trade-off of allowing some idle/waiting time but reducing makespan and/or total completion time related costs should be considered. The numerical results of the paper specifically show that sometimes allowing (and thus paying) idle/waiting times is beneficial to reduce other cost sources such as utilization and/or flow time, WIP, etc.

Finally, as the used heuristic algorithm, developed for different problems (even though adapted for the considered problem), could have had some effect on the performed comparisons, future research will be devoted to developing more sophisticated ad hoc solution algorithms that include the economic trade-off in finding the optimal solution.

## References

- [1] A. Alfieri, M. Garraffa, E. Pastore, F. Salassa, Permutation flowshop problems minimizing core waiting time and core idle time. *Comp. & Ind. Eng.*, 108983. (2023) <https://doi.org/10.1016/j.cie.2023.108983>
- [2] H. Aydilek, A. Aydilek, M. Allahverdi, A. Allahverdi, More effective heuristics for a two-machine no-wait flowshop to minimize maximum lateness. *Inter. J. of Ind. Eng. Comp.*, 13(4), (2022) 543-556. <https://doi.org/10.5267/j.ijiec.2022.7.002>
- [3] H. Öztop, M.F. Tasgetiren, L. Kandiller, Q.K. Pan, Metaheuristics with restart and learning mechanisms for the no-idle flowshop scheduling problem with makespan criterion. *Comp. & Oper. Res.*, 138, 105616. (2022) <https://doi.org/10.1016/j.cor.2021.105616>
- [4] W. Höhn, T. Jacobs, N. Megow, On Eulerian extensions and their application to no-wait flowshop scheduling. *J. of Sched.*, 15(3) (2012) 295-309. <https://doi.org/10.1007/s10951-011-0241-1>
- [5] M. de Fátima Morais, M.H.D.M. Ribeiro, R.G. da Silva, V.C. Mariani, L. dos Santos Coelho, Discrete differential evolution metaheuristics for permutation flow shop scheduling problems. *Comp. & Ind. Eng.*, 166, 107956. (2022) <https://doi.org/10.1016/j.cie.2022.107956>
- [6] R.g. Saber, M. Ranjbar, Minimizing the total tardiness and the total carbon emissions in the permutation flow shop scheduling problem. *Comp. & Oper. Res.*, 138, 105604. (2022) <https://doi.org/10.1016/j.cor.2021.105604>
- [7] R. Ruiz, E. Vallada, C. Fernández-Martínez, Scheduling in flowshops with no-idle machines. in: *Computational intelligence in flow shop and job shop scheduling*, (2009) 21-51. [https://doi.org/10.1007/978-3-642-02836-6\\_2](https://doi.org/10.1007/978-3-642-02836-6_2)
- [10] P.J. Kalczyński, J. Kamburowski, A heuristic for minimizing the makespan in no-idle permutation flow shops. *Comp. & Ind. Eng.*, 49(1) (2005) 146-154. <https://doi.org/10.1016/j.cie.2005.05.002>
- [1] Q.K. Pan, R. Ruiz, An effective iterated greedy algorithm for the mixed no-idle permutation flowshop scheduling problem. *Omega*, 44 (2014) 41-50. <https://doi.org/10.1016/j.omega.2013.10.002>
- [12] H. Yuan, Y. Jing, J. Huang, T. Ren, Optimal research and numerical simulation for scheduling no-wait flow shop in steel production. *J. of Appl. Math.* (2013) <https://doi.org/10.1155/2013/498282>
- [13] R. Alvarez-Valdés, A. Fuertes, J.M. Tamarit, g. Giménez, R. Ramos, A heuristic to schedule flexible job-shop in a glass factory. *Eur. J. of Oper. Res.*, 165(2) (2005) 525-534. <https://doi.org/10.1016/j.ejor.2004.04.020>
- [14] B. Na, S. Ahmed, G. Nemhauser, J. Sokol, A cutting and scheduling problem in float glass manufacturing. *J. of Sched.*, 17 (2014) 95-107. <https://doi.org/10.1007/s10951-013-0335-z>
- [15] A. Agnetis, D. Pacciarelli, Part sequencing in three-machine no-wait robotic cells. *Oper. Res. Letters*, 27(4), (2000) 185-192. [https://doi.org/10.1016/S0167-6377\(00\)00046-8](https://doi.org/10.1016/S0167-6377(00)00046-8)
- [16] A.P. De Abreu, H.Y. Fuchigami, An efficiency and robustness analysis of warm-start mathematical models for idle and waiting times optimization in the flow shop. *Comp. & Ind. Eng.*, 166, (2022) 107976. <https://doi.org/10.1016/j.cie.2022.107976>

- [8] H. Singh, J.S. Oberoi, D. Singh, Multi-objective permutation and non-permutation flow shop scheduling problems with no-wait: a systematic literature review. *RAIRO-Oper. Res.*, 55(1), (2021) 27-50. <https://doi.org/10.1051/ro/2020055>
- [9] A. Allahverdi, A survey of scheduling problems with no-wait in process. *Eur. J. of Oper. Res.*, 255(3), (2016) 665-686. <https://doi.org/10.1016/j.ejor.2016.05.036>
- [17] K. Maassen, A. Hipp, P. Perez-Gonzalez, Constructive heuristics for the minimization of core waiting time in permutation flow shop problems. In 2019 International Conference on Industrial Engineering and Systems Management (IESM) (pp. 1-6). IEEE. (2019) <https://doi.org/10.1109/IESM45758.2019.8948147>
- [18] W. Liu, Y. Jin, M. & Price, A new Nawaz-Enscore-Ham-based heuristic for permutation flow-shop problems with bicriteria of makespan and machine idle time. *Eng. Opt.*, 48(10), (2019) 1808-1822. <https://doi.org/10.1080/0305215X.2016.1141202>
- [19] E. Taillard, Benchmarks for basic scheduling problems. *Eur. J. of Oper. Res.*, 64(2), (1993) 278-285. [https://doi.org/10.1016/0377-2217\(93\)90182-M](https://doi.org/10.1016/0377-2217(93)90182-M)