

Luis Ferreirinha

School of Engineering, Polytechnic Institute
of Porto (ISEP/IPP)
Portugal

Sara Baptista

School of Engineering, Polytechnic Institute
of Porto (ISEP/IPP)
Portugal

Ângela Pereira

School of Engineering, Polytechnic Institute
of Porto (ISEP/IPP)
Portugal

A. S. Santos

School of Engineering, Polytechnic Institute
of Porto (ISEP/IPP)
Department of Production and Systems,
University of Minho (UM)
Portugal

J. Bastos

School of Engineering, Polytechnic Institute
of Porto (ISEP/IPP)
Portugal

A. M. Madureira

School of Engineering, Polytechnic Institute
of Porto (ISEP/IPP)
Portugal

M. L. R. Varela

Department of Production and Systems,
University of Minho (UM)
Portugal

An Industry 4.0 Oriented Tool for Supporting Dynamic Selection of Dispatching Rules Based on Kano Model Satisfaction Scheduling

Production scheduling is an optimizing problem that can contribute strongly to the competitive capacity of companies producing goods and services. A way to promote the survival and the sustainability of the organizations in this upcoming era of Industry 4.0 (I4.0) is the efficient use of the resources. A complete failure to stage tasks properly can easily lead to a waste of time and resources, which could result in a low level of productivity and high monetary losses. In view of the above, it is essential to analyse and continuously develop new models of production scheduling. This paper intends to present an I4.0 oriented decision support tool to the dynamic scheduling. After a first solution has been generated, the developed prototype has the ability to create new solutions as tasks leave the system and new ones arrive, in order to minimize a certain measure of performance. Using a single machine environment, the proposed prototype was validated in an in-depth computational study through several instances of dynamic problems with stochastic characteristics. Moreover, a more robust analysis was done, which demonstrated that there is statistical evidence that the proposed prototype performance is better than single method of scheduling and proved the effectiveness of the prototype.

Keywords Dynamic Production Scheduling, Single Machines, Decision Support Tool, Industry 4.0.

1. INTRODUCTION

The topic of combinatorial optimization (CO) consists of a set of key problems, primarily in the areas of mathematics, computer science and engineering. Developing efficient techniques to find either a minimum or maximum value of an objective function composed of several independent variables is objective of this research field. There is a set of categories where these problems can fit, depending on their characteristics, i.e., if they are continuous or discrete, restricted or not, single or multiobjective, static or dynamic, and so on. To find satisfactory solutions to these kinds of problems, heuristics and meta heuristics can be used [2]. In fact, due to the increase in complexity and the need for flexibilization of productive systems, which is increasingly becoming a more serious concern in the currently Industry 4.0 (I4.0) direction new heuristics are being continuously developed to produce acceptable results for combinatorial optimization problems [3]. Scheduling allocates the resources to activities and determines in which sequence the activities should be executed, in order to optimize a performance measure [4].

In view of the above, it is necessary to analyse more agile and flexible methods to solve these problems, which

not only contemplate a satisfaction of the interests of the client, but also the interests of the company.

The model proposed in [5] was revised and extended with an in-depth computational study to validate the concept and performance of the prototype through statistical evidence. The developed prototype tool is designed not only in a static environment but also in a dynamic one. In a static environment, the tool allows an analysis of several measures of performance simultaneously, which leads to a greater balance of the interests of the stakeholders [6]. Regarding dynamic scheduling, the prototype was developed with the purpose of not requiring any interaction with the user, the software itself alternates between priority rules according to the objectives in question. This article will only discuss the dynamic environment.

To validate the tool, multiple instances composed of a vast set of tasks with normally distributed stochastic characteristics were executed in a single machine environment. Then, a statistical analysis was conducted in order to find evidences of the effectiveness of the tool.

The remaining sections of this paper are organized as follows: in section 2 revises production scheduling. The developed prototype is presented in section 3 with the computational results. In section 4 is where the paper finally presents some conclusions and provides some ideas for future work.

2. SCHEDULING PROBLEMS

The production scheduling is preponderant for the survival of a company. This term can be defined as a deci-

sion function that concerns with the allocation of limited resources, like machines, operators, raw material, and others, to the operations, over a period of time, in order to optimize certain performance measure [3,7,8].

In scheduling problems, two types of feasibility constraints are usually found: the capacity limits of the machine and the technological constraints that condition the production order. The scheduling problems aim to answer two questions [9]: What resources should execute the operations and when should each operation be executed. In this perspective, the first question allows to define the allocation and the second one the sequence [10].

However, in many situations, the scheduling problems are reduced to a sequencing problem. That happens when there are no alternative resources to perform the operations or when the allocation problem has already been solved [4].

Although the main objective of these problems is to find an optimum schedule, the latter can vary according to the requirements of the foreman and the company. Sometimes, it will be better to choose a schedule that allows all the jobs to be done in the shortest period of time, other times a schedule that minimizes the average flow time will be better, or one minimizing the maximum tardiness of the jobs. These are just a few examples of alternative performance measures that the company has to face. It is more likely that one performance measure will not be enough, and the foreman will have to face a multi-objective optimization problem, for instance a scheduling problem often with conflicting objectives [8,11–13].

However, even when all the questions related to sequencing, allocation and preferred objective are answered, it does not mean a feasible and fixed schedule was achieved. In fact, due to unpredicted interruptions the idealized schedule becomes easily obsolete, and in need of fast modifications, which often leads to an average quality schedule. Such unexpected interruptions can occur for a variety of reasons, like related to the shop floor (e.g., breakdowns, rework), or for market or customer reasons (e.g., order cancelation, changes in delivery times, among others) [8].

With that stated, it becomes easily to understand how important the research in scheduling theory for industrial practice is.

2.1 Scheduling and production environments

The scheduling environment can be classified by three factors: the production environment, if the problem is static or dynamic, as well as the stochastic or deterministic nature of the scheduling variables [4].

The production environment is characterized by the organizational configuration of the production system itself resulting from the association between the transformation processes and the available processors that execute the transformation [10]. Generally, two classes of this type of environment are identified, the uni-operation, where the environment is characterized by the processing of jobs with a single operation, and the multi-operation, which concerns the processing of jobs with at least two operations [4].

2.2 Scheduling on single machines

The production scheduling on single machines is a one-operation environment consisting of a single processor, which executes all jobs. The apparent simplicity stems from the fact that the problem only requires the sequencing of the tasks, since there is only one machine to process them [14]. A schematic of this problem can be seen in Figure. 1.

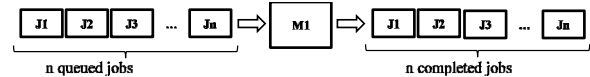


Figure 1. Single Machine Environment.

It is very common to decompose more complex environments into single machine ones [15–18].

2.3 Production Scheduling Methods

In a given problem, the determination and evaluation of each solution is only possible when the solution space is relatively small. However, as the dimension of the problem increases, the resolution may be rendered intractable. The methods of approaching scheduling problem can be divided into [7]:

- **Enumerative methods:** It is through implicit enumeration and comparison of all possible solutions that the optimal solutions are found. For complex problems, these methods do not allow optimal solution to be obtained at reasonable computing times. Dynamic programming, constrained programming, and Branch and X are examples of these methods;
- **Heuristic methods:** They make it possible to find solutions that are not optimal but are satisfactory in reasonable computational times. Among them, it is possible to identify local search heuristics, meta-heuristics and constructive heuristics. The latter are optimization techniques that start from an empty solution and sequentially build solutions without considering the impact of decisions in later phases;
 - **Priority rules:** These rules are usually used when there is only a need to sequence tasks. Basically, these rules sort the tasks in a sequence and determine by what order they should be executed. Such classification could be static or dynamic. In the first case, the position of the tasks within the sequence does not change with time, while in the second case, the tasks are sequenced whenever a decision must be made [9], [18], [19].

Multi agent Systems (MAS) are also considered as an approach to scheduling [20].

2.4 Dynamic Scheduling

In real world industry it is rare to schedule in a static environment, typically new tasks are launched during the implementation of a scheduling solution, making it immediately obsolete. In such scenario, it is necessary to adapt the scheduling solutions in order to incorporate

the jobs that arrived at the shop floor. In this type of scheduling, it is assumed that all tasks are not known at the beginning of the problem, that is, new tasks can be released during the implementation of a given scheduling solution.

A dynamic scheduling problem is handled through purely reactive models that respond to the launch of new tasks. Generally, dynamic scheduling reorders a queue of jobs, which have not yet been executed, whenever a new task is released [7,10,21,22].

Meta Heuristics and hyper-heuristics have become increasingly popular, due to their ability to solve real world optimization problems, such as scheduling problems in dynamic environments. A hyper-heuristic is a heuristic that seeks to automate the processes of selection, combination, generation or adaptation of several simpler heuristics. The most important characteristic of a hyper-heuristic is that they search a space of heuristics instead of directly searching the space of solutions [23–25].

Many papers have addressed the application of hyper-heuristics in either dynamic environments and in more complex scheduling problems, such as the hybrid flow shop (HFS) and job shop problems [23], [24], [26]. For example, in [23] it is proposed a new hybrid Dispatching Rule Based Genetic Algorithms (DRGA) which searches for the best sequence of dispatching rules and the number of operations to be handled by each dispatching rule simultaneously. It was used to solve different variants of the multi-objective job shop problem. In [24], the authors designed a framework that uses the genetic programming hyper-heuristic techniques to combine Palmer's and Gupta's algorithms, in order to obtain new and better heuristics to solve a flow shop scheduling problem. In [26] the authors propose a Genetic Algorithm (GA) for a Hybrid Flow Shop problem, which combines a meta-heuristic with a hyper-heuristic. At the same time the Human-Computer interaction for the control of Manufacturing systems are examined in [27].

What differentiates the present article from those presented above is that the developed tool decides between priority rules based on a real time monitoring of two performance criteria, while the majority of research is based on a metaheuristic (GA).

3. DYNAMIC SELECTION OF DISPATCHING RULES BASE ON THE KANO MODEL SATISFACTION SCHEDULING TOOL (DSDR-KMS-ST)

We resorted to the single machine scheduling problem for the analysis of the developed tool. This application, apart from allowing quick access to the insertion, removal, edition, and visualization of jobs, also gives the user the possibility to choose between static or dynamic environments.

3.1. Dynamic Environment

The DSDR-KMS-ST was designed to schedule the work in progress whenever new tasks enter the system, i.e., while a task is being processed on the machine the prototype reorders the tasks that are in the system thro-

ugh a constructive heuristic. When no other tasks enter the system, the schedule determined by the prototype is maintained. Basically, the developed prototype allows dynamic scheduling autonomously. Thus, the foreman does not need to schedule manually, the sole focus of the foreman is to define the optimization criterion.

In this case, the definition of the objectives of the schedule are classified through the degree of satisfaction of the Kano's Model. In this initial phase of the project the DSDR-KMS-ST presented here intends to portray a test of the Kano's Model concept with two performance measures, like Figure 2 shows. So, it is easily to see that the user has to deal with two objectives, where the average flow time was defined as a one dimensional criterion and the maximum tardiness was defined as a must be criterion of satisfaction. This is just an example of how the performance measures can be classified in the tool, where in this case, to run some tests with the model, the objectives were defined as described earlier. In the future it is the authors' intent that the user will be allowed to choose and classify whatever objectives from vast set of performance measures according to operational objectives of the organization.

Thus, for the defined objectives, it is assumed that the user does not want their maximum tardiness to exceed the set value and, at the same time, wants the average flow time to be as small as possible. As the Figure 2 shows, the maximum tardiness represents an obligatory attribute in the Kano's model, so if it is not fulfilled it results in an extreme dissatisfaction of the "client". As for the average flow time, it represents a proportional attribute, since it corresponds to a degree of satisfaction proportional to the degree of performance of the attribute, i.e., the smaller the average flow time, the higher the satisfaction of the "client" [28], [29].

The DSDR-KMS-ST also shows several useful performance indicators, at each instant, informing the user and giving him a chance to evaluate the performance of the system.

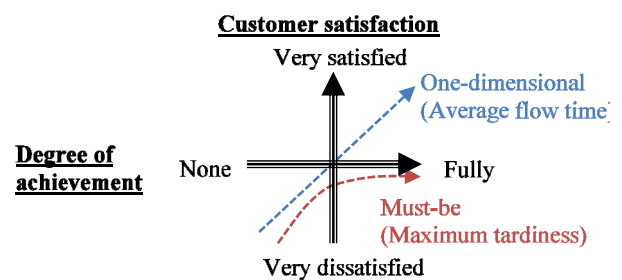


Figure 2 Kano's Model [28,29].

3.2. Computational Study

To validate the operation of the DSDR-KMS-ST, normal distributions were used to generate 100 jobs with the attributes shown in Table 1.

Table 1. Distributions used

	[1-20]	[21-40]	[41-60]	[61-80]	[81-100]
r_j	N(75,35)	N(165,45)	N(250,45)	N(340,45)	N(440,45)
$d_j[N(\mu,\sigma)+r_j+p_j]$	N(150,2)	N(205,2)	N(275,2)	N(365,2)	N(490,2)
P_j	N(10,2)				
W_j	N(10,3)				

The problem can be described as follows. Each of n jobs (numbered $1, \dots, n$) has to be processed without interruption on a single machine that can only perform one job at a time. From the Table 1., Job j ($j=1, \dots, n$) becomes available for processing at a stochastic release date r_j . Each job has an integer processing time p_j , a due date, d_j and a positive weight w_j . Whenever a schedule order is set, it is possible to compute the completion time C_j and the tardiness $T_i = \max \{C_j - d_j, 0\}$ of job i ($i=1, \dots, n$) and also the average flow time of the current schedule $Aft_s = \left(\sum_{j=1}^n c_j \right) / n$. The objective is to schedule the jobs in order to keep the maximum tardiness as small as possible, hopefully under the value set by the user, and whenever possible, minimize the average flow time in the shop floor. From the literature, it is known that the minimization of the maximum tardiness in single machines is achieved through the Earliest Due Date (EDD) rule, which consists of sequencing the jobs in ascending order of their delivery date. As for the minimization of the average flow time, is achieved through the Shortest Processing Time (SPT) rule, which consists of sequencing the job in ascending order of their processing time. With that in mind, an algorithm was developed to alternate between these two priority rules as time passes and new tasks are released. A brief outline of what has been described here is shown in Figure 3.

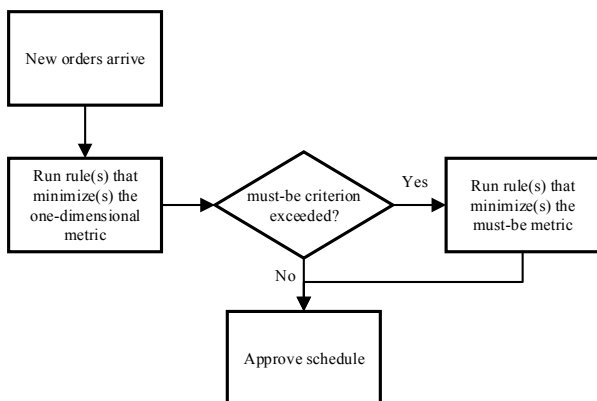


Figure 3 Diagram of the DSDR-KMS-ST system logic.

In figure 3 is shown the operating logic of the designed prototype. As soon as new tasks enter the system, the tool reschedules the set of jobs waiting to be processed, considering those that have entered the factory floor. According to the SPT rule in order to minimize the average flow time and if the maximum tardiness of the schedule does not exceed the value preset by the user. If not, the schedule is approved, and the tasks are processed accordingly. Otherwise, if the tool detects that the maximum tardiness in the system exceeds the set value, the prototype reschedules the tasks according to the EDD rule. Since the EDD rule minimizes the maximum tardiness, it is to be expected that it decreases and falls below the indicated value. Due to the characteristics of the tasks, of course there will be situations where neither with the tasks scheduled according to the EDD the maximum tardiness will be lower than the intended, so in these situations the tool will inform the

user of the situation but will keep the schedule found, since it is which minimizes the must be goal according to the Kano's model.

3.3. Computational Results

To test the performance of the tool it was defined that the maximum tardiness could not exceed the value of zero. Another feature of the prototype lies in the opportunity to set a certain margin for the objectives, in this case the tardiness value. Such a margin value will certainly help prevent the maximum tardiness from exceeding the target value since it will reschedule the tasks according to the EDD as early as the margin. For example, set the tardinessgoal value as T_g and the margin value as T_{mg} , if the user sets a $T_{mg} = 20$ temporal units, it means that as soon as the tool detects that through the SPT the scheduling plan has a tardiness higher than $T_g - 20$, it will reschedule tasks using EDD to decrease the tardiness, regardless of whether the SPT's plan did not exceed the value of T_g .

However, such a margin value will compromise the average flow time of the system, since schedules elaborated according to the SPT will be discarded, once the tool force the use of EDD to satisfy the tardiness margin. That said, it is possible to realize the impact that the value of the margin can cause in the system, if it is too high, it will fulfil the required tardiness, however, the user's satisfaction with the time average flow time will be lacking. On the other hand, the lower the value of the margin, or even zero, and depending on the characteristics of the tasks, the tool may not have the ability to prevent the plan from exceeding the maximum tardiness delineated, and then a must be criterion will not be met, which should be more detrimental to the quality of the scheduling solution. Therefore, a balance between the margin and the amount of tardiness actually intended should be considered.

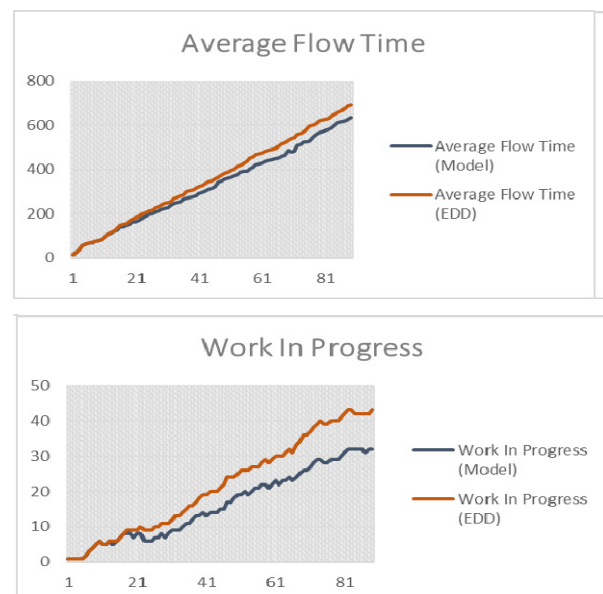


Figure 4 Obtained results.

At this initial design stage, the system reacts to the margin value as described above. But what would be interesting, and since what is intended to present in this pro

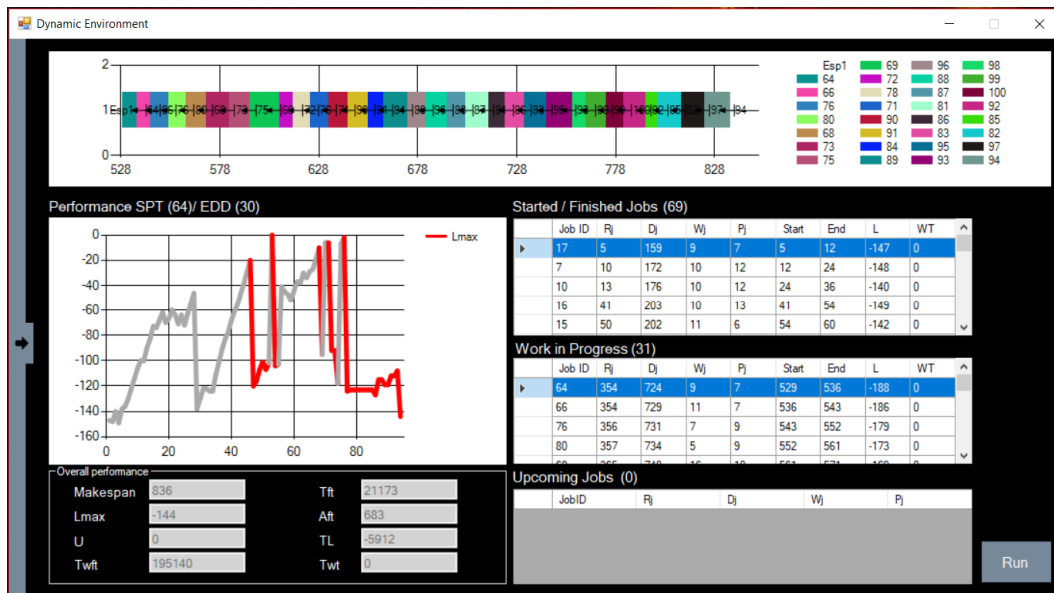


Figure 5. DSDR-KMS-ST.

ject is a totally autonomous tool adapted to the dynamic environment, would be the one dimensional margins autonomously adjusted to the system. This could be achieved in several ways, one of which could be a forecast model coupled with the tool, or even the company's own forecasting system, which anticipated the possible arrival of new tasks with shorter deadlines. This would signal the system to increase the margin value. Otherwise, the system would receive orders to reduce margin slack and thus provide more satisfaction to both the company and the customer.

To analyse the behaviour of the tool, the 100 tasks generated, and the tardiness value defined, mentioned before, were used and the results obtained were compared to a system where only EDD is used. These results are shown in Figure 4.

Figure 4 compares the average flow time and the work in progress (WIP) between the model and the system based only in EDD. As expected, the average flow time in the system based on EDD tends to get higher faster than the DSDR-KMS-ST since the rule focuses on minimizing the maximum tardiness. As for the work in progress, the system based only in EDD tends to create a bigger WIP than the DSDR-KMS-ST. This is due to the fact that EDD does not prioritize jobs with shorter processing times, which makes long tasks have the possibility of being processed first creating a bottleneck in the section and, in turn, generates high WIP faster.

Figure 5 shows the practical operation of the developed tool. In it, it is possible to verify its potentiality, since it allows the graphic perception of the schedule as well as the current performance measures of the system, giving the user a detailed set of useful information. The performance graph is where the behaviour of the maximum tardiness is found. In it, it is possible to verify the moments in which the tool uses the SPT rule, represented in grey, as well as the rule EDD, represented in red. As mentioned earlier, the tool begins to schedule the tasks that enter the system through the SPT, which causes that as the time elapses, the maximum tardiness begins to increase. As soon as the tool detects that it will not be possible to obtain a schedule with less than the

defined tardiness, the tool autonomously resorts to the EDD to minimize the maximum tardiness, hence the sharp decreases in the graph when the EDD is used.

Thus, Figure 5 shows the effectiveness of the tool, where the maximum tardiness was always less than zero and the average flow time was minimized whenever possible, reaching 683 in the final phase. As for the EDD based system, the average flow time of the final task set was 744.33, 64.33 units higher than DSDR-KMS-ST. It should also be noted that when the model ran, the SPT rule was used 64 times and the EDD rule was used 30. In the final stage of this primarily study there were 31 tasks in progress and 69 had been processed in DSDR-KMS-ST, which compared to the EDD based model, has less 11 tasks waiting in the system.

3.4. Discussion of results

In this subchapter we intend to analyse the performance of the developed tool. That is, we intend to find the statistical evidence that proves that the concept of the prototype achieves a better performance than single method of scheduling, in this particular case, in comparison with the EDD rule. For this, 30 instances of 100 tasks with the stochastic characteristics previously presented (Table 1) were generated. Figures 6 and 7 show the results obtained, by the tool and by continuous use of EDD, for each instance in relation to the average flow time and to work in process, respectively

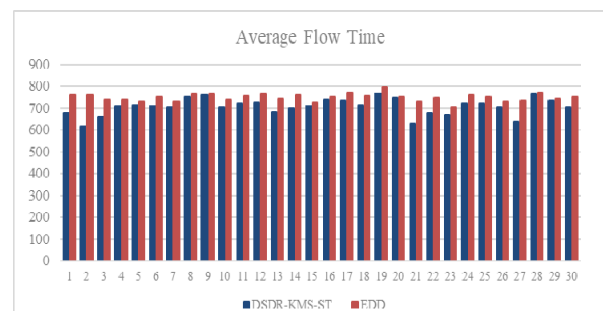


Figure 6. Results obtained from average flow time for each instance.

As can be seen from figure 6, the average flow time for all instances obtained by the DSDR-KMS-ST is always lower than the EDD rule. In fact, this result was already expected, since whenever the tool has opportunity it uses the SPT to minimize average flow time, unlike the EDD, which only focus in minimizing the maximum tardiness. From the results presented in figure 6 it is worth noting that, for DSDR-KMS-ST, the minimum value recorded was 618.44, compared to the maximum value of 765.98, and showed an average of 706.68-time units with a standard deviation of 37.74. As for the results of the EDD-based model, they recorded a minimum value of 702.38, a maximum value of 797.91 and a mean value of 749.23 time units, with a standard deviation of 17.48. With those results in mind, there appears that the tool has better performance than de EDD rule on a dynamic environment concerning the minimization of the maximum tardiness and the minimization of the average flow time. However, to verify that statement it requires a significant parametric t-test for independent samples, since both distributions are normal, with the hypothesis:

$$H0: \mu \text{ AFT_DSDR-KMS-ST} = \mu \text{ AFT_EDD}$$

$$H1: \mu \text{ AFT_DSDR-KMS-ST} < \mu \text{ AFT_EDD}$$

Since the equality of the variances is not assumed, it is verified that the average flow time through the DSDR-KMS-ST is lower than the results obtained by the EDD rule, $t(40,901)=-5,603$, $p\text{-value}=0,000$. That is, at the 5% level there is statistical evidence that the average flow time through the DSDR-KMS-ST is lower than that obtained through the EDD in a dynamic scheduling environment.



Figure 7 Results obtained from WIP for each instance.

In figure 7 the difference of the WIP between the DSDR-KMS-ST and the EDD is more noticeable, and similar to the average flow time, the WIP registered by the DSDR-KMS-ST is lower in all instances. Through the tool was registered a minimum value of 9 and a maximum value of 49 tasks in the system, as well as an average of 36.53 with a standard deviation of 8.33. For EDD, a minimum value of 38, a maximum value of 55, a mean of 45.30 units in the system with a standard deviation of 3.91 was verified.

Like the previous stated, to verify that the WIP in the system through the tool is lower than through the EDD on a dynamic environment schedule it requires a significant parametric t-test for independent samples, since both distributions are also normal, with the hypothesis:

$$H0: \mu \text{ WIP_DSDR-KMS-ST} = \mu \text{ WIP_EDD}$$

$$H1: \mu \text{ WIP_DSDR-KMS-ST} < \mu \text{ WIP_EDD}$$

Since the equality of the variances is not assumed, it is verified that the WIP through the DSDR-KMS-ST, is lower than the results obtained by the EDD rule,

$t(41,204)=-5,216$, $p\text{-value}=0,000$. That is, at the 5% level there is statistical evidence that the WIP through the DSDR-KMS-ST is lower than that obtained through the EDD in a dynamic scheduling environment.

4. CONCLUSION

In this paper we intend to demonstrate an approach, along with a proposed tool to support dynamic scheduling problems in the current context of Industry 4.0, without user intervention through the DSDR-KMS-ST. The developed prototype can schedule tasks dynamically, through the change between dispatch rules that best fit the fulfilment of the objectives outlined by the user.

The objectives for the schedule are classified according to the Kano's model, where at this initial phase of the project it was defined that the maximum tardiness could not exceed a certain value (must be criterion) and that the average flow time should be the least possible (one dimensional criterion). The logic of the tool operation is simple, as new tasks enter the system, they are scheduled according to the SPT. If the plan does not have a maximum tardiness higher than initially established, it will remain the same, otherwise the tool will automatically resort to the EDD to prevent the user from incurring in dissatisfaction.

To analyse the performance of the DSDR-KMS-ST, 30 instances with 100 stochastic tasks were generated and for both average flow time and WIP, there was statistical evidence, at the level of 5%, that the tool has better performance than the single EDD for a dynamic schedule environment with objectives classified by the Kano's model.

Regarding future work, it is intended that the user has the possibility to classify more performance criteria, by the degree of satisfaction, and that the tool dynamically alternates between more dispatch rules in order to optimize the customer satisfaction function. In the future, the tool should link to the MRP system, which would allow it to know when new tasks enter the system. It would be interesting if the margin value for the maximum tardiness changed dynamically depending on the performance of the system, once it was found that a fixed margin could have a great impact on the quality of the scheduling. Therefore, methodologies that allow the autonomous adjustment of the margin will also be analysed.

ACKNOWLEDGMENT

This work has been supported by FCT - Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC /00319/2019.

REFERENCES

- [1] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing.," *Science*, vol. 220, no. 4598, pp. 671–80, May 1983.
- [2] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Inf. Sci. (Ny)*, vol. 237, pp. 82–117, Jul. 2013.
- [3] A. Artiba and S. E. Elmaghraby, *The Planning and Scheduling of Production Systems: Methodologies and applications*. Springer US, 1996.

- [4] M. L. R. Varela, "Uma contribuição para o escalonamento da produção baseado em métodos globalmente distribuídos," Universidade do Minho, 2007.
- [5] L. Ferreirinha *et al.*, "A Dynamic Selection of Dispatching Rules Based on the Kano Model Satisfaction Scheduling Tool," Springer, Cham, 2019, pp. 339–346.
- [6] D. Shabtay and G. Steiner, "The single-machine earliness-tardiness scheduling problem with due date assignment and resource-dependent processing times," *Ann. Oper. Res.*, vol. 159, no. 1, pp. 25–40, Mar. 2008.
- [7] A. B. G. S. e Santos, "Análise do Desempenho de Técnicas de Otimização no Problema de Escalonamento," ISEP, 2015.
- [8] S. Goren and I. Sabuncuoglu, "Robustness and stability measures for scheduling: single-machine environment," *IIE Trans.*, vol. 40, no. 1, pp. 66–83, Jan. 2008.
- [9] K. R. Baker and D. Trietsch, "Principles of sequencing and scheduling, A John Wiley & Sons," *Inc. Publ.*, 2009.
- [10] M. Pinedo, *Scheduling: theory, algorithms, and systems*. 2012.
- [11] S. B. Akers and J. Friedman, "A Non-Numerical Approach to Production Scheduling Problems," *J. Oper. Res. Soc. Am.*, vol. 3, no. 4, pp. 429–442, Nov. 1955.
- [12] G. Campanella, R. A. Ribeiro, and L. R. Varela, "A Model for B2B Supplier Selection," Springer, Berlin, Heidelberg, 2011, pp. 221–228.
- [13] M. L. R. Varela, J. N. Aparício, and S. C. Silva, "A Web-based application for manufacturing scheduling," in *IASTED International Conference on Intelligent Systems and Control*, 2003, pp. 400–405.
- [14] S. C. Graves, "A Review of Production Scheduling," *Oper. Res.*, vol. 29, no. 4, pp. 646–675, Aug. 1981.
- [15] A. Madureira, I. Pereira, P. Pereira, and A. Abraham, "Negotiation mechanism for self-organized scheduling system with collective intelligence," *Neurocomputing*, vol. 132, pp. 97–110, May 2014.
- [16] L. R. Varela and T. Pinto, "Comparing Extended Neighborhood Search Techniques Applied to Production Scheduling The Romanian Review Precision Mechanics," 2010.
- [17] A. Madureira and I. Pereira, "Intelligent Bio-Inspired system for manufacturing scheduling under uncertainties," in *2010 10th International Conference on Hybrid Intelligent Systems*, 2010, pp. 109–112.
- [18] J. Blazewicz, K. H. Ecker, E. Pesch, G. Schmidt, and J. Weglarz, *Handbook on Scheduling: From Theory to Applications*. Springer Berlin Heidelberg, 2007.
- [19] M. Pfund, J. W. Fowler, and J. N. D. Gupta, "A survey of algorithms for single and multi-objective unrelated parallel-machine deterministic scheduling problems," *J. Chinese Inst. Ind. Eng.*, vol. 21, no. 3, pp. 230–241, Jan. 2004.
- [20] M. Petrović, Z. Miljković, and B. Babić, "Integration of process planning, scheduling, and mobile robot navigation based on TRIZ and multi-agent methodology," *FME Trans.*, vol. 41, no. 2, pp. 120–129, 2013.
- [21] D. Terekhov, D. G. Down, and J. C. Beck, "Queueing-theoretic approaches for dynamic scheduling: A survey," *Surv. Oper. Res. Manag. Sci.*, vol. 19, no. 2, pp. 105–129, Jul. 2014.
- [22] M. L. R. Varela and R. A. Ribeiro, "Distributed Manufacturing Scheduling Based on a Dynamic Multi-criteria Decision Model," Springer, Cham, 2014, pp. 81–93.
- [23] J. A. Vázquez-Rodríguez and S. Petrovic, "A new dispatching rule based genetic algorithm for the multi-objective job shop problem," *J. Heuristics*, vol. 16, no. 6, pp. 771–793, Dec. 2010.
- [24] C. E. Nugraheni, "On the Development of Hyper Heuristics Based Framework for Scheduling Problems in Textile Industry," *Artic. Int. J. Model. Optim.*, 2016.
- [25] C. Bharatiraja, G. Kasilingam, J. Pasupuleti, C. Bharatiraja, and Y. Adedayo, "Single Machine Connected Infinite Bus System Tuning Coordination Control using Biogeography-Based Optimization Algorithm," *FME Trans.*, vol. 41, pp. 120–129, 2013.
- [26] J. Antonio Vázquez Rodríguez and A. Salhi, "A Robust Meta-Hyper-Heuristic Approach," 2006.
- [27] V. Spasojević-Brkić, G. Putnik, V. Shah, H. Castro, and Z. Veljković, "Human-computer interactions and user interfaces for remote control of manufacturing systems," *FME Trans.*, vol. 41, no. 3, pp. 250–255, 2013.
- [28] C. Högström, M. Rosner, and A. Gustafsson, "How to create attractive and unique customer experiences," *Mark. Intell. Plan.*, vol. 28, no. 4, pp. 385–402, Jun. 2010.
- [29] M. Löfgren, L. Witell, and A. Gustafsson, "Theory of attractive quality and life cycles of quality attributes," *TQM J.*, vol. 23, no. 2, pp. 235–246, Mar. 2011.

**АЛАТ ЗА ОРИЈЕНТАЦИЈУ ИНДУСТРИЈЕ 4.0
ЗА ПОДРШКУ ДИНАМИЧКОМ ИЗБОРУ
ПРАВИЛА ДИСПЕЧИНГА НА ОСНОВУ
ПЛАНИРАЊА ЗАДОВОЉСТВА КАНО
МОДЕЛА**

**Л. Ферериња, С. Баптишта, А. Перейра,
А. С. Сантош, Ж. Баштош, А. М. Мадуреира,
М. Л. Р. Варела**

Програмирање производње је оптимизациони проблем који може пуно допринети капацитету конкурентности компанија које производе робу и услуге.

Начин да се промовише опстанак и одрживост организација у овој наредној ери Индустрије 4.0 (И4.0) је ефикасно коришћење ресурса. Потпуни неуспех да се задаци правилно изведу могу лако довести до губитка времена и ресурса, што може довести до ниског нивоа продуктивности и високих новчаних губитака. Имајући у виду горе наведено, неопходно је анализирати и континуирано развијати нове моделе програмирање производње.

Овај рад има за циљ да представи И4.0 оријентисан алат за подршку одлучивању за динамичко програмирање производње. Након што је развијено прво

решење, развијени прототип има способност да креира нова решења док задаци напуштају систем и стижу нови, како би се минимизирала одређена мера перформанси. Користећи сценарио/проблем једне машине, предложени прототип је проверен кроз детаљну рачунску студију кроз неколико случајева динамичких проблема са стохастичким карактеристикама. Штавише, направљена је и робуснија анализа која је показала да постоје статистички докази да су перформансе предложеног прототипа бољи од појединачног метода програмирања производње и ефективност прототипа је доказана.