A linear programming model for the parallel non-related machines problem, in the drying area of a chilean sawmill

Un modelo de programación lineal para el problema de máquinas paralelas no relacionadas en el área de secado de un aserradero en Chile

Um modelo de programação linear para o problema de máquinas paralelas não relacionadas na área de secado de uma serraria no Chile

Nayaret Cifuentes*
Gustavo Gatica**
Rodrigo Linfati***

Fecha de recepción: 20 de enero de 2017
Fecha de aceptación: 13 de agosto de 2017

Abstract

Scheduling of activities in manufacturing and service enterprises should perform efficiently, since it impacts both productivity and competitiveness. This study analyzes a real case of green wood dryers in a sawmill in Chile, with a set of ten parallel machines with three different technologies, with 161 jobs, on a monthly planning horizon. The methodology considered two stages: first, the products were grouped by density and fiber type; second, a mathematical model was proposed based on linear programming, which was modeled with AMPL software. In addition, we conducted a statistical analysis to evaluate the solution quality and the computing times, using the CPLEX and GUROBI commercial solvers. The results of the computational experiment showed a reduction in the makespan of 8.5 %, allowing us to conclude that the solver CPLEX is better than the solver GUROBI, regarding CPU time and number of instances optimally solved in 59.3 % of the analyzed cases. The most influential parameters for computing time were GUROBI cuts (evaluated at 0), CPLEX mipcuts (evaluated at 2), and repeatpresolve (evaluated at 0). The time difference in the latter parameter was statistically significant.

Keywords: Parallel machines; Parameterization; Scheduling of the production.

Resumen

La programación de actividades en empresas manufactureras y de servicios debe funcionar de manera eficiente, ya que afecta la productividad y la competitividad. Este estudio analiza un caso real de programación en secadores de madera verde en un aserradero de Chile, con un conjunto de 10 máquinas paralelas con tres tecnologías diferentes, con 161 trabajos, en un horizonte de planificación mensual. La metodología considera dos etapas: en primer lugar,

* Ms.C. Universidad del Bío-Bío (Nayaret, Chile).
** Ph. D. Universidad Andrés Bello (Santiago de Chile, Chile). ggatica@unab.cl. ORCID: 0000-0002-1816-6856.
*** Ph. D. Universidad del Bío-Bío (Nayaret, Chile). rlinfati@ubiobio.cl. ORCID: 0000-0002-7659-383X.

los productos se agrupan por densidad y tipo de fibra, y en segundo lugar, se propone un modelo matemático basado en la programación lineal, que es modelado con el software AMPL. Se realiza un análisis estadístico sobre la calidad de la solución y el tiempo de cómputo, con los programas comerciales CPLEX y GUROBI. Los resultados del experimento computacional permiten reducir el makespan en un 8,5 %, concluyendo que el solver CPLEX resultó ser mejor que el solver GUROBI, respecto al tiempo de CPU y al número de instancias resueltas al óptimo, en el 59,3 % de los casos analizados. Los parámetros más influyentes para el tiempo de cálculo fueron: “cuts” en GUROBI (evaluados en 0), “mipcuts” en CPLEX (evaluados en 2) y repeatpresolve (evaluados en 0). La diferencia en tiempo de este último parámetro es estadísticamente significativa.

Palabras clave: Máquinas paralelas; Parametrización; Programación de la producción.

Resumo

A programação de atividades nas empresas manufatureiras e de serviços deve funcionar de maneira eficiente, já que afeta a produtividade e a competitividade. Este estudo analisa um caso real de programação em secadores de madeira verde em uma serraria do Chile, com um conjunto de 10 máquinas paralelas com três tecnologias diferentes, com 161 trabalhos, em um horizonte de planificação mensal. A metodologia considera duas etapas: em primeiro lugar, os produtos se agrupam por densidade e tipo de fibra, e em segundo lugar, propõe-se um modelo matemático baseado na programação linear, que é modelado com o software AMPL. Realiza-se uma análise estatística sobre a qualidade da solução e o tempo de cômputo, com os programas comerciais CPLEX e GUROBI. Os resultados do experimento computacional permitem reduzir o makespan em um 8,5 %, concluindo que o solver CPLEX resultou ser melhor que o solver GUROBI, a respeito do tempo de CPU e do número de instâncias resolvidas ao ótimo, em 59,3 % dos casos analisados. Os parâmetros mais influentes para o tempo de cálculo foram: “cuts” em GUROBI (avaliados em 0), “mipcuts” em CPLEX (avaliados em 2) e repeatpresolve (avaliados em 0). A diferença em tempo deste último parâmetro é estatisticamente significativa.

Palavras chave: Máquinas paralelas; Parametrização; Programação da produção.

I. Introduction

Scheduling of activities in manufacturing and service enterprises should perform efficiently, due to its impact on productivity and competitiveness. Scheduling should contribute to have a greater control of the productive system operations to achieve the proposed goals, and thus to gain a competitive advantage in the market [1-6]. Regarding scheduling, difficulties arise from the lack of balance between the production capacity and the capacity of meeting the client's demands in an opportune way [2, 7-11].

Production planning is an optimization problem [5]. The literature presents cases of activity programming in areas like textile, transport, office, aluminum metallurgy, semiconductor production, molding plastics, and printed circuit boards, among others [12]. Further, optimization contributes to the allocation, transport, cutting of materials, plant distribution, forecasts, classification of inventories, production planning, and production scheduling [2, 3, 13]. For sequencing and programming tasks in automated cells, scheduling of storage with muffling in intermediate stages, operations sequence in an assembly system, or for scheduling of tasks in parallel machines with deadlines, it is necessary to minimize the total processing time (makespan), the total time of the system tasks flow, the satisfactory delivery times, the efficient use of resources, the idle time of servers or machines, and the completeness of orders, among others [14].

In industrial production processes, it is possible to work in one or more of the productive stages with parallel machines, which contributes to improve the work execution and to distribute a job between several resources or type of products. All problems of allocation and sequencing in parallel machines, which seek a proximate way to assign and perform jobs on different machines, belong to the NP-Hard class and are fundamental in manufacturing industries [1, 4, 9, 12-13, 15-19].

The literature depicts heuristics solutions, metaheuristics, and exact methods that consider Branch and Bound, Branch Price algorithms, Columns Generation, Benders and Danzig Wolf decomposition, linear and lagrangian relaxation, Giffler and Thompson (for active problems and without machines delay), Johnson (for flow shop problems of 2 and 3 machines), Johnson (for job shop problems of 2 machines and multiple jobs), Page's method, CDS algorithm, NEH algorithm , Gupta method, and IG algorithm [2, 8] .

Our case study considered ten drying chambers (machines) in parallel, of three different technologies, with different temperatures, capabilities and times. More than 40 different products that entered the dryers as requirements were differentiated for avoiding the mix up of density and type of fiber (central or lateral).

In this study, a mathematical model, based on linear programming, is proposed using AMPL software. In addition, a statistical analysis is conducted to evaluate the quality of the solutions provided by the CPLEX and GUROBI commercial solvers. The following section describes the two-phase procedure used in the wood drying industry to formulate the model, which is subsequely validated with two commercial solvers. Additionally, the computing time is improved through a process of parameter analysis, and validated by an exhaustive statistical analysis.

II. Methods and materials

The methods were divided into two stages. In the first stage, the products were grouped by density and fiber type, considering a tolerance range of 3 mm of difference in density; the data were obtained from the SAP system of the company; and the loads were composed according to the maximum capacity of the dryer, with each load consisting of 30 lots of the same density and type; however, in some cases, due to optimization of sawmill production, it was not possible to form a load, hence mixtures were formed. In the second stage, the mathematical model was formulated, and sets, parameters, variables, and constraints were defined.

The first stage consisted in grouping products and adjusting loads. To group products, we obtained the history of the sawmill production, which accounted for a total of 69 products. We classified these products into 45 groups, which we assigned to the adjustment of the loads, according to quality conditions.

III. Formulation of the mathematical model

The requirements to formulate the model in the drying area considered ten drying chambers, with a volume dependent on the product type. Chamber technologies were classified into three types: three dryers of UHT, six of ACTH, and one of ACTH-DT type. Dryer technologies were differentiated by temperature, processing time, and capacity; and the 69 products, by type and density. The idea is that the maximum capacity of the dryer contains just one type of product to avoid a wood mix up that may alter the drying process. The processing times, according to the type of product and drying, were provided by the management and validated by the times of the drying control system. Products delivered by sawmills cannot be exposed to environmental conditions for more than three days, i.e., the maximum date to enter the dryer is ($k+3$), where k represents the product's storage day by drying area. The programming horizon was monthly.

The model was formulated to propose a solution to the problem of production scheduling in the drying area, focused on making the decision of what, when, and how much product to program, and on which machine perform the operation. Sets, parameters, objective variables function, and restrictions were defined.

A. Sets

The model has two sets M and N, where $\boldsymbol{M} = \{1,..,m\}$ represents the set of available parallel machines, and $\boldsymbol{N} = \{1,..,n\}$, the set of jobs, which fluctuates according to the monthly load.

B. Parameters

A set of four parameters were identified for the model. $\boldsymbol{P_{ij}}$ is the process (drying) time of job j in machine i. $\boldsymbol{B_{ij}}$, the possibility of processing job j in machine i; this paramenter is established by the technology of the dryer, via a compatibility index. Each job has a reception time, denoted by $\boldsymbol{K_j}$ that corresponds to the time at which the product is stored in the courtyard prior to starting the drying process. Jobs have an expiration time defined as $\boldsymbol{D_j}$, since, according to the quality standard, the products cannot be exposed for more than three days to the environmental conditions.

C. Variables

$$\boldsymbol{X_{ij}} = \begin{cases} 1, if\ job\ j\ is\ processed\ in\ machine\ i \\ 0,\ otherwise. \end{cases}$$

The binary type variable X_{ij} takes a value of 1 if it assigns job j to machine i.

$\boldsymbol{C_{max}}$: Maximum time of completion for all jobs

The continuous decision variable determines the maximum time of completion of the last job j processed on machine i.

$\boldsymbol{f_{ij}}$: Entering time of job j in machine i

The continuous variable determines the starting date of job j in the dryer.

D. Objective Function

$$Min\ = C_{max}$$

The Objective Function establishes that the maximum time of completion of those processes should be minimized.

E. Restrictions

$$\sum_{i=1}^{m} X_{ij} = 1\ \forall\, j\ \in N\ (1)$$

$$X_{ij}\ \leq\ B_{ij}\ \ \forall i \in M\ \wedge\ \forall j \in N\ (2)$$

$$\sum_{j=1}^{n} P_{ij} \times X_{ij} \leq C_{máx}\ \forall\ i \in\ M\ (3)$$

$$f_{ij} +\ P_{ij} \times X_{ij}\ \leq\ f_{i(j+1)}\ \forall\ i \in\ M\ \wedge\ \forall j \in N\ (4)$$

$$K_j\ \leq\ f_{ij} + L\left(1 -\ X_{ij}\ \right) \forall\ i \in\ M\ \wedge\ \forall j \in N\ (5)$$

$$f_{ij}\ \leq\ D_j + L\left(1 -\ X_{ij}\ \right) \forall\ i \in\ M\ \wedge\ \forall j \in N\ (6)$$

$$X_{ij}\ \in \{0, 1\}\ \forall i \in M\ \wedge\ \forall j \in N\ (7)$$

$$f_{ij} \geq 0 \; \forall i \in M \; \wedge \; \forall j \in N \; (8)$$

$$C_{máx} \geq 0 \; (9)$$

Restriction (1) ensures that each job j can only be scheduled and processed by a single machine i of m available machines. Restriction (2) ensures compatibility and possibility to process job j on each machine i, according to the conditions of each type of dryer technology, i.e., relating each machine i with jobs j. Restriction (3) establishes the maximum competition time (C_{max}) as the greatest amount of processing time on each machine. Restriction (4) establishes sequencing of loads, i.e., the time at which job j enters into machine i plus its processing time, which must be less than the entering time of consecutive jobs j into the same machine i for all machine i and every job j. Restriction (5) refers to the entry time of job j into machine i, establishing when job j is assigned to machine i; the entering time of this job j into machine i must be greater than or equal to the time when j was received; this is for all i belonging to the set M of machines and all j belonging to the set N of jobs. Restriction (6) assures that time of job j entering into machine i does not exceed job j maturity time, when assignation occurs; this is for all i belonging to the set M of machines and for all j belonging to the set N of jobs. Restriction (7) defines the nature of the decision variable, being of binary type. Restrictions (8) and (9) establish the non-negativity of the variables. The L parameter corresponds to a very large, but finite number according to the Big M method.

IV. Results

All experiments were processed on a computer with 4 GB RAM, processor Intel Core i5, model 3570, 3.4 GHz, Windows 7 Professional 32-bit. The model was implemented in AMPL [20], executing 108 combinations of instances and parameters.

A. Scheduling of parallel machines

Manually programming of the ten dryers, in a horizon of six days, resulted in a makespan of 132.23 hours; whereas, programing with the proposed model resulted in a shorter makespan of 121 hours, that is, an improvement of 8.5 %. Moreover, the model decreased the dead time from 454 to 203 hours, and increased the total loads from 31 to 38 loads for the first week of the horizon analyzed.

B. Solver analysis

To analyze the parameters that impact the computing time, nine instances were generated, with computing times ranging from 300 to 5000 (s) and from 6 to 11000 (s), in CPLEX version 12.6.3.0 and GUROBI 6.5.0, respectively. Table 1 presents the instance, objective function value (Z Value), time resolution (T (s)), and last improvement time (T_l(s)) for both solvers.

Table 1
Summary Gurobi and CPLEX Solver

	CPLEX				GUROBI		
Test No.	Z	T(s)	T_l(s)		Z	T(s)	T_l(s)
1	122.25	2618.83	610.92		122.25	3669.64	75.00
2	121.00	327.64	317.48		121.00	2960.18	2930.00
3	121.00	5648.53	5609.27		121.00	366.59	366.00
4	134.00	2524.16	2510.42		134.00	5.99	5.00
5	134.00	421.77	413.72		134.00	28.97	28.00
6	122.25	1669,52	1065.08		122.25	176.89	11.00
7	122.25	400.72	395.07		122.25	272.14	109.00
8	122.25	319.93	241.46		122.25	11590.47	11590.00
9	131.00	806.87	17.58		131.00	353.98	1.00

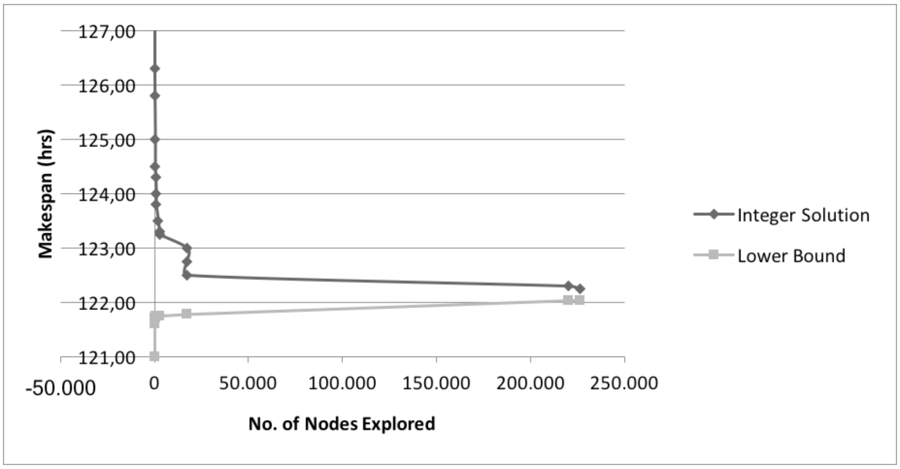

Fig. 1. Graph Instance Resolution No. 1.

Fig. 1 shows the evolution of the upper dimension (Whole Solution) and the lower dimension (Best Bound) for instance No. 1. Upper and lower domensions slowly converge, accounting for most of the improvements for the whole solution (Incumbent) in the first explored nodes, after which, it is only necessary to enumerate nodes up to achieving to close the difference between both levels (GAP), and thus obtain the optimal certificate for the solution, producing only minor improvements in the last nodes.

C. Analysis of parameters with CPLEX and GUROBI

108 instances and parameter combinations were executed to record times, and to comparatively study the GUROBI and CPLEX parameters. In CPLEX, the following parameters were assessed: *Mipcuts* evaluated at (-1), i.e., does not generate cuts, and *Mipcuts* evaluated at (2), i.e., aggressively generates cuts. *Mipemphasis* (0) emphasizes feasibility over optimality, and *Mipemphasis* (2) emphasizes optimality over feasibility. *Repeatpresolve* (0) deactivates, *represolve* (3) repeats pre-processing of the root node and allows new cuts. In GUROBI, *Cuts* evaluated at (0) does not generate cuts, and *Cuts* evaluated at (3) aggressively generates cuts. *Mipfocus* (1) emphasizes feasibility over optimality, and *Mipfocus* (2) emphasizes optimality over feasibility. *Presolve* (0) disables the preprocessing of the model, and *Presolve* (2) pre-processes the model in a more aggressive way.

The computing times (s), according to each parameter, are shown in Table 2; tests are bounded by 3600 (s), and thus avoid significant deviations stemming from excessive times.

Table 2

Computing Times for Parameters Studying the CPLEX and GUROBI Solver

	CPLEX							**GUROBI**						
Test No.	**Normal Model**	***MIP-CUTS (-1)***	***MIP-CUTS (2)***	***MIP-EMPHASIS (1)***	***MIP-EMPHASIS (2)***	***REPEAT-PRESOLVE (0)***	***REPEAT-PRESOLVE (3)***	**Normal Model**	***CUTS* (0)**	***CUTS* (3)**	**MIP-FOCUS (1)**	**MIP-FOCUS (2)**	***PRE-SOLVE* (0)**	***PRE-SOLVE* (2)**
1	**2618.83**	605.96	366.76	***3600.00**	***3600.00**	695.58	2222.23	3696.64	224.47	***3600.00**	***3600.00**	1573.59	1024.77	362.12
2	**327.64**	40.95	***3600.00**	199.98	145.83	67.21	***3600.00**	2960.18	*3600.00	***3600.00**	***3600.00**	*3600.00	*3600.00	*3600.00
3	**5648.53**	15.94	16.40	2192.33	9.44	16.38	208.77	366.59	12.09	***3600.00**	3549.01	19.41	***3600.00**	28.83
4	**2524.16**	18.58	12.39	1809.24	9.75	606.41	***3600.00**	5.99	37.92	27.63	23.40	18.10	***3600.00**	9.41
5	**421.77**	375.51	12.99	174.32	0.92	4.18	11.19	28.97	85.29	7.61	3.87	10.92	67.11	7.63
6	**1669.52**	1068.75	516.44	***3600.00**	***3600.00**	694.03	***3600.00**	176.89	369.17	3483.35	***3600.00**	3172.01	1069.16	397.46
7	**400.72**	***3600.00**	1443.23	***3600.00**	174.64	158.15	***3600.00**	272.14	854.98	286.57	***3600.00**	***3600.00**	1250.73	1260.25
8	**319.93**	***3600.00**	568.13	***3600.00**	***3600.00**	457.47	***3600.00**	11590.00	818.88	***3600.00**	***3600.00**	1005.87	857.50	528.73
9	**806.87**	43.32	1787.60	637.17	827.87	1027.89	***3600.00**	353.98	430.98	***3600.00**	***3600.00**	2494.82	***3600.00**	467.17
Average	**1637.55**	**1041.00**	**924.88**	**2157.00**	**1353.35**	**414.14**	**2671.35**	**2161.26**	**714.86**	**2422.80**	**2797.36**	**1721.64**	**2074.36**	**740.18**

D. Statistical analysis

The statistical analysis was performed via hypothesis testing, using the difference between the base model and each parameter; the null and alternative hypotheses were defined as $H_0 = \mu_1 \leq \mu_2$ and $H_1 = \mu_1 > \mu_2$, respectively.

Table 3 summarizes the statistical analysis. The only case in which the parameter mean is less than the base model is *repeatpresolve* valuated at zero for the rest of the cases. The null hypothesis is accepted, therefore, there is no statistical evidence that the averages of the parameters in the evaluated models have less times than the basic model of each *solver.*

TABLE 3

STATISTICAL ANALYSIS PARAMETERS STUDYING CPLEX AND GUROBI SOLVER

Model 1	Model 2	α	Statistic t	t critical	Conclusion
Base Model CPLEX	*mipcuts (-1)*	0.05	0.7737	**1.7459**	H_0 NOT rejected
	mipcuts (2)	0.05	1.0044	**1.7459**	H_0 NOT rejected
	mipemphasis (1)	0.05	-0.6682	**1.7459**	H_0 NOT rejected
	mipemphasis (2)	0.05	0.3421	**1.7459**	H_0 NOT rejected
	repeatpresolve (0)	**0.05**	**2.0330**	**1.7459**	$\boldsymbol{H_0}$ **rejected**
	repeatpresolve (3)	0.05	-1.3300	**1.7459**	H_0 NOT rejected
Base Model GUROBI	*cuts (0)*	0.05	1.0968	**1.7459**	H_0 NOT rejected
	cuts (3)	0.05	-0.1880	**1.7459**	H_0 NOT rejected
	mipfocus (1)	0.05	-0.4645	**1.7459**	H_0 NOT rejected
	mipfocus (2)	0.05	0.3221	**1.7459**	H_0 NOT rejected
	presolve (0)	0.05	0.0640	**1.7459**	H_0 NOT rejected
	presolve (2)	0.05	1.0764	**1.7459**	H_0 NOT rejected

E. GUROBI vs. CPLEX analysis

All instances were resolved with feasible solutions for the problem of scheduling parallel machines. CPLEX solved more instances to the optimum. The time of the parameter of CPLEX *repeatpresolve* is emphasized; in deactivated mode, it records the lowest time in most of instances.

F. Analysis of cuts with CPLEX and GUROBI

The average time of the *mipcuts* parameter evaluated at -1 was 1041.00 (s), and two of the nine instances failed to be resolved to the optimum. When evaluating the parameter at 2, the average time was 924.88 (s), and only one of the nine instances failed to be resolved to the optimum. As for time and number of decisive instances, conducting aggressive cuts yielded better results.

In GUROBI, the *cuts* parameter in deactivated mode worked slightly better, recording a time average of 714.86 (s) with only a single instance not resolved. Evaluating the parameter at 3 resulted in an average time of 2422.80 (s), with five instances failing to be resolved to the optimum.

G. Analysis of mipemphasis and mipfocus with CPLEX and GUROBI

In CPLEX, with the *mipemphasis* parameter evaluated at 1, the average time increased by 31.72 %, and four instances did not converge to the optimum. When evaluating the parameter at 2, the time average was reduced by 17.3 %, and three instances did not obtain the optimum. On GUROBI, with the *mipfocus* parameter evaluated at 1, the average time increased by 29.43 %, in addition, nine of six instances failed to resolve to the optimal. For the case of evaluating the parameter at 2, the time average fell by 20.37 %, and two instances did not resolve to the optimum.

Notably, with the *mipemphasis* parameter evaluated at 2, for a group of instances (2, 3, 4, 5, 7), times declined considerably. Moreover, in instances 3, 4, and 5, the problems resolved almost instantly.

H. Analysis of presolve with CPLEX and GUROBI

The parameter *repeatpresolve* evaluated at zero recorded an average of 414.14 (s) and all instances resolved to the optimum; the best being the basic CPLEX model, with a decrease of 75 % in the resolution average time. For this case, with a level of significance of 0.05, the null hypothesis is rejected, statistically supporting that the basic CPLEX model has times of resolution higher than the *repeatpresolve* (0) model. Evaluating the parameter at 3, genereated 6 instances that did not resolve to the optimal, and an average computing time that raised to 2671.35 (s).

In GUROBI, evaluating the *presolve* parameter at zero generated six unresolved instances to the optimum, and an average time of 2074 (s). When assessing the parameter at two, the aggressive reformulations and cuts on the root node helped solving the problem, resulting in a single instance that could not be resolved to the optimum, and a considerable decrease in the average time to 740.18 (s).

V. Conclusions

Regarding the execution of the model, we obtained favorable results for the input of 38 loads. Applying the model, the dead times decreased to 203 hrs and a total of 38 loads for the first week of December, according to the data of the dry area. With manual programming, however, based on the data downloaded from the control system of the dryers, a dead time of 454 hrs and a total of 31 loads were obtained. We can conclude that implementing a model of production programming positively influences the productive process.

When the size of the instances is superior to 38 loads, the model increased the computing time; quick solutions of good quality to resolve these cases exist; however, the greater amount of time is used to demonstrate that this is the optimal solution. We conducted an experiment to find out which parameters can improve this behavior; for this, we analyzed two types of solvers, CPLEX and GUROBI.

Of the108 instances executed in AMPL and the 54 CPLEX instances, 60 % experienced improvements regarding time. As for the 54 instances in GUROBI, only 33 % improved the behavior of the model. Furthermore, CPLEX achieved more instances resolved to the optimum, which leads us to conclude that the solver CPLEX is better than the solver GUROBI, in terms of the number of instances resolved.

Regarding the average of each parameter, the only one that presented improvements was *repeatpresolve* in deactivated mode, which considerably reduced the computing times. Also, in the hypothesis testing, *repeatpresolve* in deactivated mode was the only parameter whose average differences were significant.

VI. Recommendations and future work

The model showed symmetry given more than one machine with the same technology, and therefore, it presented equal processing times. We suggest exploring new formulations based on representatives and cut planes that actively exploit this condition to improve computing times. In the future, heuristic techniques based on mathematical models (math-heuristics) could help to reduce computing time and efficiently explore the solution space.

Acknowledgements

NC acknowledges partial support from Magíster en Ingeniería Industrial. GG acknowledges partial support from Universidad Andres Bello. RL acknowledges partial support from CONICYT FONDECYT through grant 11150370, and the Grupo de Logística y Transporte of Universidad del Bío-Bío. This support is gratefully acknowledged.

Author's contributions

Every author contributed in an equal manner to this research.

References

[1] E. Pérez, I. Pérez, and M. Jiménez, "Algoritmo genético para secuenciación de pedidos en taller de mecanizado con máquinas en paralelo, recirculación y tiempos de preparación," *Ing. Ind. Actual. Nuevas Tend.*, vol. 4 (12), pp. 38-53, 2014.

[2] P. Ballesteros, and C. Duque, "Aplicación del fraccionamiento de operaciones en una heurística constructiva en programación secuencial para

asignación de varios trabajos a varias máquinas en paralelo," *Sci. Tech.*, vol. 18 (4), pp. 664-671, 2013.

[3] P. P. B. Silva, D. P. B. Riveros, and J. E. B. Bolívar, "Aplicación de una heurística constructiva en programación secuencial para asignación de varios trabajos a varias máquinas en paralelo," *Sci. Tech.*, vol. 18 (1), pp. 120-125, 2013.

[4] H. Gámez, "Asignación de trabajos en máquinas en paralelo mediante un modelo de programación entera," *Sci. Tech.*, vol. 2 (52), pp. 73-77, 2012.

[5] J. Osorio, O. Castrillón, J. Toro, and J. Orejuela, "Modelo de programación jerárquica de la producción en un Job shop flexible con interrupciones y tiempos de alistamiento dependientes de la secuencia," *Ing. E Investig.*, vol. 28 (2), pp. 72-79, 2008.

[6] A. J. Ruiz-Torres, J. H. Ablanedo-Rosas, N. Alomoto, and D. Jadan, "Minimización de la tardanza en problemas de programación de tareas en maquinas paralelas con deterioro de los recursos,", *Rev. Ing. Ind.*, vol. 13 (1), pp. 51-75, 2014.

[7] E. Salazar-Hornig, and S. J. C. Medina, "Minimización del makespan en máquinas paralelas idénticas con tiempos de preparación dependientes de la secuencia utilizando un algoritmo genético," *Ing. Investig. Tecnol.*, vol. 14 (1), pp. 43-51, 2013.

[8] E. Salazar Hornig, and B. Figueroa Morales, "Minimización de la tardanza para el flowshop flexible con setup utilizando heurísticas constructivas y un algoritmo genético," *Ingeniare Rev. Chil. Ing.*, vol. 20 (1), pp. 89-98, Apr. 2012. DOI: http://doi.org/10.4067/S0718-33052012000100009.

[9] E. Salazar Hornig, and R. Sarzuri Guarachi, "Algoritmo genético mejorado para la minimización de la tardanza total en un flowshop flexible con tiempos de preparación dependientes de la secuencia," *Ingeniare Rev. Chil. Ing.*, vol. 23 (1), pp. 118-127, Jan. 2015. DOI: http://doi.org/10.4067/S0718-33052015000100014.

[10] J. Lopez, J. Giraldo, and J. Arango, "Reducción del tiempo de terminación en la Programación de la Producción de una Línea de Flujo Híbrida Flexible (HFS)," *Inf. Tecnológica*, vol. 26 (3), pp. 157-172, 2015.

[11] E. Vallada, C. Maroto, R. Ruiz, and B. Segura, "Problemas de programación de la producción en el sector cerámico español," in *27 Congreso Nacional de Estadística e Investigación Operativa*, vol. 27, pp. 638-676, 2003.

[12] F. Muñoz, and R. Moraga, "Heurística constructiva visionaria para el problema de máquinas paralelas no relacionadas con tiempos de setup dependientes de la secuencia," *Rev. Ing. Ind.*, vol. 5 (1), pp. 43-59, 2006.

[13] J. Caballero, and G. Mejía, "Redes de Petri y algoritmos genéticos, una propuesta para la programación de sistemas de manufactura flexible," *Ing. Univ.*, vol. 10 (1), pp. 0-27, 2010.

[14] E. Delgado, C. Cortés, and Ó. Duarte, "Aplicación de algoritmos genéticos para la programación de tareas en una celda de manufactura," *Ing. E Investig.*, vol. 25 (2), pp. 24-31, 2005.

[15] J. Torres, R. Montoya, C. Arboleda, D. Paternina, and Y. Frein, "Minimización del tiempo total de flujo de tareas en una sola máquina: Estado del arte," *Ing. Desarro.*, No. 12, pp. 118-129, 2002.

[16] L. Ramírez, "Algoritmo heurístico para resolver el problema de programación de operaciones minimizando el tiempo total de ejecución ponderado de las actividades (WJCJ) en un taller de flujo flexible con restricciones de fechas de liberación (RJ), alistamiento (SJK) y máquinas proporcional (QM) en las estaciones," *Prospectiva*, vol. 9 (2), pp. 59-64, 2011.

[17] F. Baesler, R. Moraga, and O. Cornejo, "Introducción de elementos de memoria en el método simulated annealing para resolver problemas de programación multiobjetivo de máquinas paralelas," *Ingeniare Rev. Chil. Ing.*, vol. 16 (3), pp. 428-437, Dec. 2008. DOI: http://doi.org/10.4067/S0718-33052008000300006.

[18] F. Baesler, L. Ceballos, and M. Ramírez, "Programación multiobjetivo de máquinas moldureras a través de algoritmos meméticos," *Maderas Cienc. Tecnol.*, vol. 8 (3), pp. 183-192, 2006.

[19] P. P. B. Silva, D. P. B. Riveros, and M. E. B. Loaiza, "Programación disyuntiva para *n* tareas en *m* máquinas en producción por pedidos," *Sci. Tech.*, vol. 1 (30), pp. 273-278, 2006.

[20] R. Fourer, D. Gay, and B. Kernighan, "A modeling language for mathematical programming," *Manag. Sci.*, vol. 36 (5), pp. 519-554, May. 1990. DOI: http://doi.org/10.1287/mnsc.36.5.519.