World Journal of Advanced Engineering Technology and Sciences

eISSN: 2582-8266

Cross Ref DOI: 10.30574/wjaets

Journal homepage: https://wjaets.com/

(Research Article)

Combination of genetic algorithm and shifting bottleneck heuristics to solve a job shop scheduling problem in wood processing company

Diem-Thuy Vien-Nguyen [1, 2], Phi-Hung Tran [1, 2], Quynh-Lam Ngoc-Le [1, 2, 3] and Ngoc-Hien Do [1, 2, *]

[1] *Department of Industrial & Systems Engineering, Faculty of Mechanical Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam.*

[2] *Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Vietnam.*

[3] *VNU Center for Education Accreditation, Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Vietnam.*

World Journal of Advanced Engineering Technology and Sciences, 2024, 11(01), 037–047

Publication history: Received on 02 December 2023; revised on 17 January 2024; accepted on 20 January 2024

Article DOI: https://doi.org/10.30574/wjaets.2024.11.1.0017

Abstract

The scheduling problem has always been the concern of most companies in the flexible Job shop or Job shop environments because of its complexity. It is really a challenge for researchers to solve or find out the optimum solution. The lead time production at Teak Woods line in a wood processing company, a case study, faced with about 80% of orders which could not be done on time, although its capacity meet requirements. Therefore, it is very useful and necessary to set up a production scheduling model in order to reduce production time and synchronize the product's parts through efficient modulation in an optimum sequence of jobs on the shopfloor. A combination of Genetic Algorithm and Shifting Bottleneck Heuristics was used to find out high quality solutions. It shows that the productivity increased from 2.32 m^3/day to 4.3 m^3/day (up 98.27%); Processing time was reduced from 3248.6 minutes to 1352.1 minutes (down 58.38%); The sync time decreased from 7554.8 minutes to 350.8 minutes (down 95.36%).

Keywords: Genetic Algorithm; Shifting Bottleneck Heuristics; Combination of Genetic Algorithm and Shifting Bottleneck Heuristics; Job shop Scheduling Problem; Wood Processing Company.

1. Introduction

This paper would solve a wood processing company scheduling problem, in which about 80% of orders could not be done on time as the plan. Actually, different products would follow different production processes, but they have to share the limited resources such as machine, operator, equipment, and tool. Unfortunately, the priority of the jobs on the shop floor was set by manual way, which belongs to the manager experiment. In another hand, there are many scheduling algorithms or heuristics to solve the job shop models. Each of them has their own advantages and disadvantages. They are classified into three types of algorithms including experiment algorithms, optimal algorithms and nearest neighbor algorithms. Experiment algorithms show scheduling results in a short time, but they do not sure about the quality of the solutions. Optimal algorithms could find out optimal solutions but they would face with the very difficulty to solve the big scheduling problem. Nearest neighbor algorithm, which could be known as the improvement heuristics algorithms, could solve big problem with an acceptable quality solution. According to Hong Zhou, Waiman Cheung, Lawrence C. Leung [8], an algorithm combining local and global search would quickly find a better solution than a simple one. In this research, the company has to moderate a large and complex number of jobs in term of complicated shopfloor scheduling problem, NP-hard problem. The better solution could be achieved by applying the heuristics algorithms [5].

[*] Corresponding author: Ngoc-Hien Do

In the study, a combination of Genetic Algorithm (GA) and Shifting Bottleneck Heuristic (SBH) would be introduced to solve that problem.

2. Literature Review and Methodology

2.1. Scheduling

Scheduling is the process of arranging, controlling and optimizing work and workloads in a production process or manufacturing process. Scheduling is used to allocate plant and machinery resources, plan human resources, plan production processes and purchase materials [10].

There are five popular dispatching rules using on scheduling, which are earliest due date first (EDD), weighted shortest processing time first (WSPT), longest processing time first (LPT), with biggest weight first (WBW), shortest processing time first (SPT). Intended objective function of these rules are shown in Table 1, in which:

- C_{max}: completion time of the last job
- $\sum \omega_i C_i$: total weighted completion time
- $\sum C_i$: total completion time

Table 1 Objective function

Name	Objective function
EDD	C_{max}
LPT	C_{max}
WSPT	$\sum \omega_i C_i$
WBW	$\sum \omega_i C_i$
SPT	$\sum C_i$

2.2. Algorithm

2.2.1. Genetic Algorithm

Genetic Algorithm (GA) is a search-based optimization technique based on the principles of Genetics and Natural Selection [1]. It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a long time to solve. It is also used to solve optimization problems, in research, and in machine learning.

GAs have various advantages which have made them immensely more popular than other algorithms [1,2,5,14]:

- Provides a list of “good” solutions, not a single one.
- Useful in a large – space search and a large number of parameters involved.
- Uses probability changing rules, not determinism.
- Try to get objective function value without other information.

A GA begins by creating a random initial population then creates a sequence of new populations, computes its objective function value, fitness value, which is used to evaluate the quality of the candidate alternatives. The good alternatives, parents, are selected to generate new alternatives as children of the population. They are produced either by making random changes to a single parents-mutation or by combining the vector entries of a pair of parents-crossover. Some individuals in the population would be replaced by the new alternative generating the next generation of the population if the their fitness values were better. In that way, the next generation is always better than the previous generation because of taking good features from the parents. The algorithm stops when one of the stopping criteria is met.

Crossover operator

The crossover operator is analogous to reproduction and biological crossover. In this, more than one parents are selected and one or more off-springs are produced using the genetic material of the parents.

- *One Point Crossover:* A random crossover point is selected and the tails of its two parents are swapped to get new off-springs.
- *Multi Point Crossover:* A generalization of the one-point crossover wherein alternating segments are swapped to get new off-springs.
- *Uniform Crossover:* Essentially flip a coin for each chromosome to decide whether or not it would be included in the off-spring then bias the coin to one parents, to have more genetic material in the child from that parents.

Mutation Operators

- *Binary mutation:* the gene is randomly flipped with 0 changed to 1 and 1 changed to 0.
- *Reciprocal exchange mutation:* a subset of genes is chosen and their values are shuffled randomly.
- *Inversion mutation* selects a subset of genes like in reciprocal mutation, but invert the entire string in the subset instead of shuffling the subset.
- *Displacement mutation:* a random subset of gene values is assigned to a randomly chosen gene.
- *Insertion mutation:* a random value from the set of permissible values is assigned to a randomly chosen gene. (Special case of displacement mutation, only 1 gene be chosen)
- *Shift mutation:* randomly select a gene and then move it to a random position on the right or left of that gene's position.

Termination conditions

It's up to the problem's requirement to choose a rational termination condition.

- *Time:* determine how long will the process take first and then show the result in that time.
- *Result:* when reaching the optimal result.
- *Number of the generation*: stops right after the prescribed number of generations, ignores the result.
- *Combination:* Combine all the conditions above.

2.2.2. Shifting Bottleneck Heuristics

The Shifting Bottleneck Heuristics is a procedure intended to minimize the time that takes to do work, or specifically, the makespan in a job shop environment [17]. The makespan is defined as the amount of time, from start to finish, to complete a set of multi-machine jobs where machine order is pre-set for each job. Assuming that the jobs are actually competing for the same resources (machines) then there will always be one or more resources that act as a 'bottleneck' in the processing. This heuristics, or 'rule of thumb' procedure minimizes the effect of the bottleneck. The Shifting Bottleneck Heuristics is intended for job shops with a finite number of jobs and a finite number of machines.

Figure 1 shows a scheduling result of: = {(o1,1, M4: 0–16), (o1,2, M3: 21–33), (o1,3, M3: 33–51), (o1,4, M1: 51–69), (o2,1, M2: 0–16), (o2,2, M4: 94–112), (o2,3, M1: 112–136), (o2,4, M4: 136–148), (o3,1, M3: 0–21), (o3,2, M1: 21–45), (o3,3, M2: 45–68), (o3,4, M1: 69–105), (o4,1, M2: 16–32), (o4,2, M4: 32–62), (o4,3, M4: 62–94), (o4,4, M3: 94–118)}

Critical path: S-O$_{1,1}$-O$_{4,1}$-O$_{4,2}$-O$_{4,3}$-O$_{2,2}$-O$_{2,3}$-O$_{2,4}$-T and Gantt chart is shown in Figure 2.

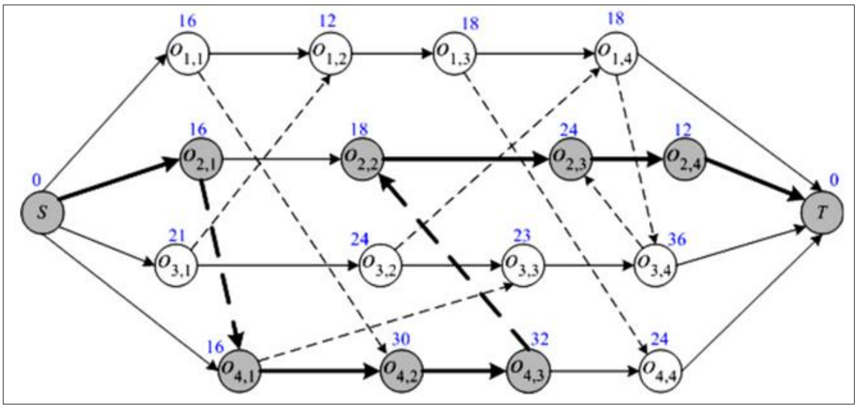

Figure 1 Disjunctive Graph in Job shop scheduling

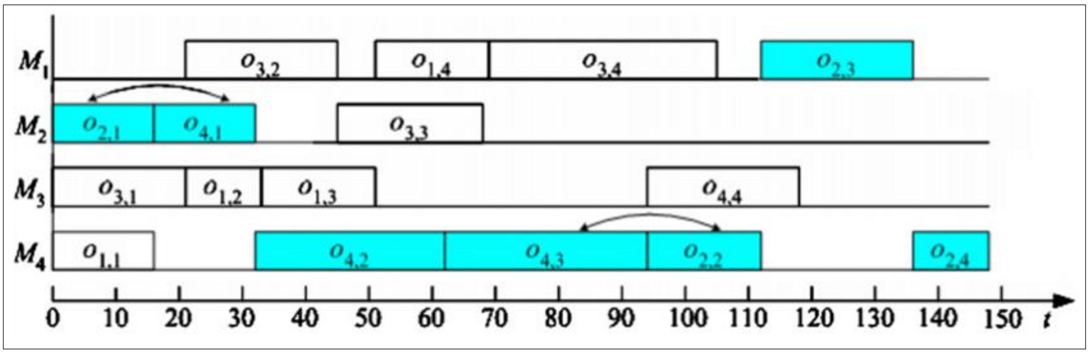

Figure 2 Gantt chart

2.2.3. Combination model of Genetic Algorithm and Shifting Bottleneck Heuristics

The GA would select parents with a better adapted function to produce offsprings. However, crossing two good individuals together would not make sure that the offsprings have a better adaptive value and solving problems by GA takes a long time. To speed up the approach to the optimal solution, this paper would integrate local search algorithms in genomic algorithms, so SBH was introduced to optimize the offsprings [23]. With a combination of GA and neighboring searches, a local optimal point would be created to contribute to global optimization.

3. Methodology

Based on the literature review, the scheduling model solving process was built with 6 steps below:

Step 1: Identify the problem by Pareto, Ishikawa diagram, Histogram chart,…

Step 2: Propose solutions. After identifying the problem and the main causes, this step would give a set of solutions. Those solutions are analyzed and evaluated specifically based on data proving the problem, the impact of the proposed solution on the case study. The result in step 2 is the problem-solving orientation through a clear and appropriate set of solutions. For example, the case study scheduling problem should be cleared on measurements as shown on Table 2.

Table 2 Scheduling Results

Serial	Volume	Task numbers	C_{max} (mins)	Average processing time (mins)	C-bar (mins)	Average sync (mins)	OF value
1	15.7	335.0	1961.0	945.3	1195.9	379.8	1032.7
2	21.9	487.0	2673.0	1006.2	1374.4	585.4	1216.4
3	28.0	678.0	2915.0	1352.1	1909.4	350.8	1591.7

Step 3: Collect data. The system data (system characteristics, basic data in settlement orientation, system results) is collected to serve the algorithm which will be built in step 4.

Step 4: Build algorithm and collect current system data. This step will use the algorithm and the current data to solve the system problem then show the algorithm's results.

Step 5: Compare the results in step 4 with the current system to know if it better or not.

Step 6: Conclude and request.

4. Case study

4.1. Mathematical model

Objective function: Minimize the weighted total between the average detail completion time - $\bar{C}$ (F1) and the synchronization time of a product (F2). Processing time, production time, is the time from the first station to the warehouse. Synchronization time is the time between receiving the first detail of the product and the final detail being completed. The greater the alpha coefficient, the more the objective function is affected by F1.

$$MinF(t) = \alpha F_1(t) + (1-\alpha)F_2(t)$$

$$= \alpha \frac{\sum_{i=1}^{n}\sum_{j=1}^{m_i}(te_{\mathrm{ijO_{ij}}-hy} - ts_{\mathrm{ij1}-hy})}{\sum_{i=1}^{n} m_i} + (1-\alpha)\frac{\sum_{i=1}^{n}(\max(te_{\mathrm{ijO_{ij}}-hy}) - \min(te_{\mathrm{ijO_{ij}}-hy})}{n} \qquad (1)$$

Criteria as followings:

- Maximum number of jobs on a machine.
- In the same detail, task n+1 just be executed after completing task n.
- Start time of the job must be later than finish time of the previous one.
- The number of machines, parts and processing stages is greater than 0.
- Alpha criteria falls between 0 and 1.

They are constraints of the model:

- $te_{\mathrm{ijO_{ij}}} \le ts_{\mathrm{ijO_{ij+1}}} : \forall i, j, h, y$
- $te_{\mathrm{ijO_{ij}}-hy} \le ts_{\mathrm{ijO_{ij+1}}-hy} : \forall i, j, h, y$
- $ts \ge 0 : \forall i, j, h, y$
- $te > 0 : \forall i, j, h, y$
- $n, m, o_{\mathrm{ij}}, h, y > 0$
- $\alpha \le 1$

Parameter:

n: number of scheduling product *(i = 1:n)*
m: m scheduling parts of each product *(j = 1:m)*
o: part processing
h: number of machine cells
y: y machines in each cell
t_s: start time
t_e: finish time
α: weighted of multi-objective function

Assumptions: Limited capacity: the number of machines is more than 1 and they are unbroken; the number of workers are known and stable; No more than 1 job on the same machine at the same time; the jobs have to be continuously and being performed after finishing the previous job (non-delay); every machine has to have its own worker at working time as a work center.

4.2. Scheduling Algorithm

Genetic coding needs to exhibit elements of the Jobshop scheduling model. A genetic coding proposal was made as Table 3.

Product A1 has 2 parts: A1.01 and A1.02, and every parts also has 3 task: 01, 02, 03. Task A1.01.01 is performed in machine M1 with 10 minutes of processing time, which are similar to other tasks. The process is encoded in the gene through the "Sequence processing" gene. The first-ranked tasks on sequence processing gene would be prioritized in scheduling model.

Table 3 Genetic coding

Task	Machine	Processing Time (mins)	Sequence Processing
A1.01.01	M1	10	A1.01
A1.01.02	M2	20	A1.02
A1.01.03	M3	15	A1.02
A1.02.01	M2	10	A1.01
A1.02.02	M1	30	A1.02
A1.02.03	M3	25	A1.01

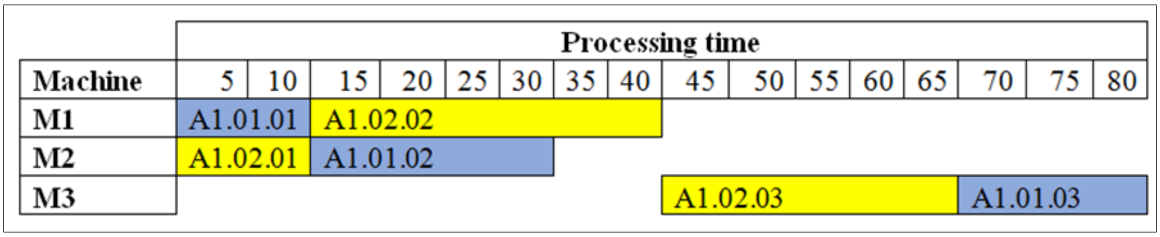

Figure 3 Gantt chart

On this scheduling model, objective function value would be affected by alpha coefficient, individual number and iteration number. In order to identify which is the most affective one, this paper keeps one of them and changes the others, try the sample products on Table 4 below.

Table 4 Sample product

Product code	Product name	Quantity
A9	Coffee Table CC1	28
BA	Dining Table	30

4.2.1. Alpha affected

Table 5 and Figure 4 show that objective function value is more sensitive with Alpha > 0.6 (reduces processing time, increases synchronizing time).

4.2.2. Individual number affected

A large number of original individuals will create the genetic abundance of the original population to reduce the number of iterations to reach a good solution as shown in Table 6 and on Figure 5. Therefore, the original number of individuals have to big enough to ensure the stability of the solutions.

4.2.3. Iteration number affected

Large number of iterations could take time to solve a large-scale problems, but it could find out the better alternatives as shown in Table 7 and Figure 6.

Table 5 Alpha affected

α	C_{max} (mins)	APT (mins)	Average C (mins)	Average sync (mins)	OF value
0.1	1104.1	666.3	775.4	137.1	200.9
0.2	1108.1	639.6	748.3	146.1	256.5
0.3	1057.8	616.3	662.9	150.1	292.9
0.4	1061.1	632.7	738.9	145.6	383
0.5	1103.5	643.8	750.5	166.6	458
0.6	1023.2	568.1	673.2	237.1	498.2
0.7	1059.1	591.5	704.3	223.5	557.8
0.8	1068.4	585.2	692	250.8	603.7
0.9	1049.2	488.6	626.6	523.5	616.5

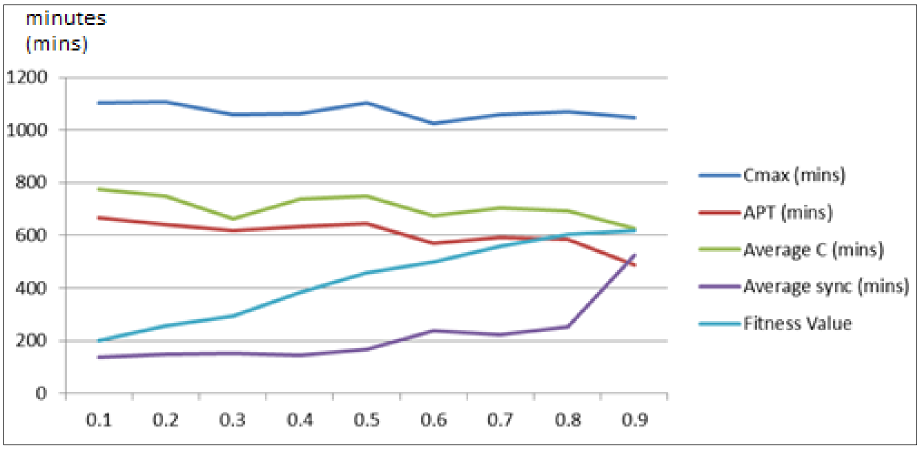

Figure 4 Alpha affected on objectives value

Table 6 Individual number affected

IN	C_{max} (mins)	APT (mins)	$\overline{C}$(mins)	Average sync (mins)	OF value
4	1039.6	580.2	690.4	333.2	618.5
6	1085.6	626.5	720.9	213	619.3
8	1041.4	592.8	701.2	299.6	620.8
10	1049.8	589.6	693.6	244	603.8
12	1010.4	556.6	659.9	302.2	588.3
14	1020.6	572.9	680.6	250.7	594.6
16	1019	555.8	666.8	302.2	593.9
18	1067.8	590.5	692.6	295.6	613.2
20	1065.2	613	715.7	303.1	633.1

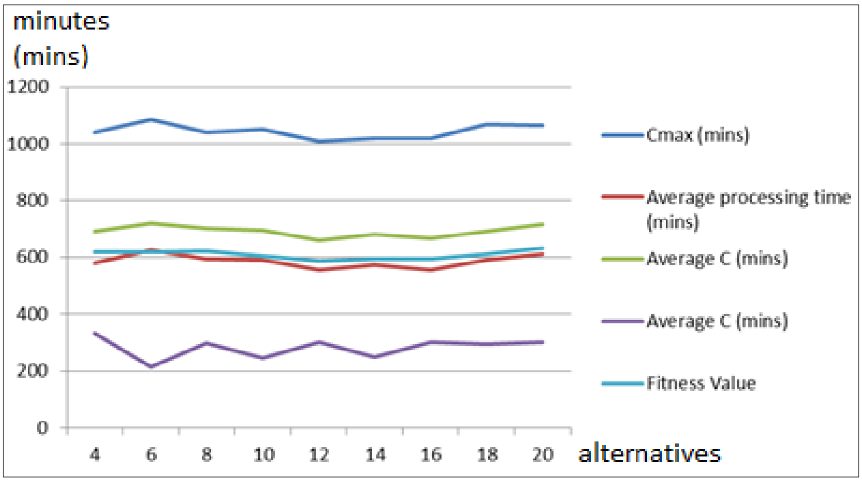

Figure 5 Individual number of alternatives (Fixed size) affection

Table 7 Iteration number affected

IN	C_{max} (mins)	APT (mins)	$\overline{C}$(mins)	Average sync (mins)	OF value
3	1109.6	603.2	726.9	301.3	646.6
6	1119.2	637	744.1	299.9	655.5
9	1049.8	589.6	693.6	244	603.8
12	1053.6	578.4	679.6	302.7	604.6
15	899.6	568.1	670.9	321.7	601.4
18	1068.6	571.3	674	367.5	613
21	1043.6	572	675.4	260.8	592.5
24	1003.8	577.9	687.3	346	618.9
27	1003.8	541.5	644	365.6	588.3

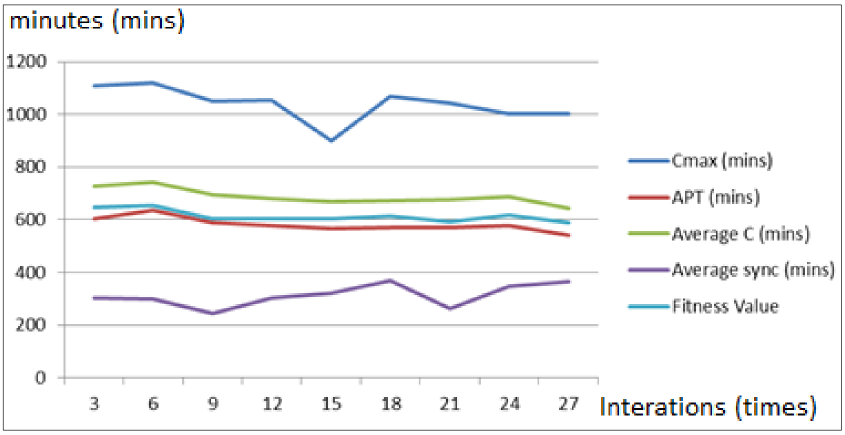

Figure 6 Iteration numbers affected on objectives values

4.3. Applying to actual schedule

In this part, the scheduling model was used to schedule 3 different cases with 3 different number of products, chosen randomly.

- Case 1: 5 products, 355 tasks, individual number is 20, alpha 0.8, iteration number is 50
- Case 2: 8 products, 487 tasks, individual number is 20, alpha 0.8, iteration number is 50
- Case 3: 12 products, 678 tasks, individual number is 20, alpha 0.8, iteration number is 50

Table 8 Case 1

Product code	Product name	Quantity
A1	KingSton Side chair	28
BA	Dining Table	30
A3	2-Seater sofa CC2	26
A7	Lounge chair CC1	27
BB	Dining ArmChair	36

Table 9 Case 2

Product code	Product name	Quantity
A1	KingSton Side chair	28
BA	Dining Table	30
A3	2-Seater sofa CC2	26
A7	Lounge chair CC1	27
BB	Dining ArmChair	36

Table 10 Case 3

Product code	Product name	Quantity
A1	KingSton Side chair	28
BA	Dining Table	30
A3	2-Seater sofa CC2	26
A7	Lounge chair CC1	27
BB	Dining ArmChair	36
A5	3 Seater Bench	35
A6	Coffee table-3 Seater bench	29
BC	Dining Table II	29

5. Results

Model input parameters that need to be noted are:

- Objective function value is more sensitive with Alpha > 0.6 (reduces processing time, increases synchronizing time).
- With the same scheduling volume, the higher the number of iterations would show the better solution.
- The scheduling volume needs to fix with the company capacity

On the current system, scheduling for 28m^3 with input coefficient alpha = 0.8, number of iterations 50 and initial number of 20. The productivity increased from 2.32m^3 /day to 4.3m^3 /day (up 98.3%), processing time was reduced from 3248.6

minutes to 1352.1 minutes (down 58.4%), the sync time decreased from 7554.8 minutes to 350.8 minutes (down 95.4%). Table 11 shows the evaluation between model and reality.

Thus, the scheduling model has solved the problems of product synchronization, processing time, workpiece productivity effectively by rational production moderation and machines scheduling. The scheduling results of the three cases as mentioned previously is shown on the Figure 7.

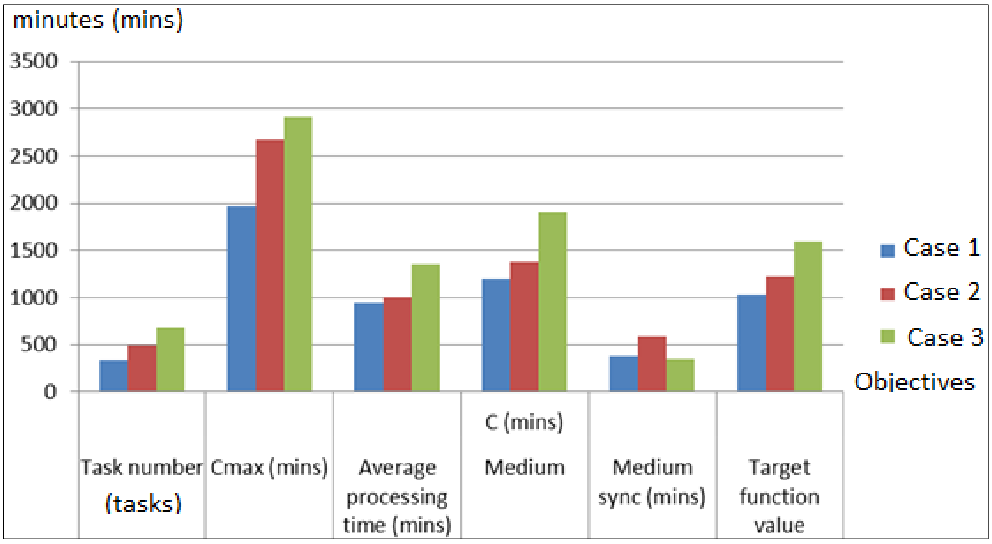

Figure 7 Scheduling results

Table 11 Model evaluation

	Average processing time(mins)	Medium sync	OF value
Reality	3248.6	7554.8	4109.9
Model	1352.1	350.8	1591.7
Changing	58.4%	95.4%	61.3%

6. Conclusion

A large volume of workloads, diverse and complex processes lead to difficulties in assigning work to production resources. This study introduced the combination of the GA and SHB to find out the quality solutions. Many wood furniture processing companies are faced with using the limited resources in the effective way to enhance the customer services and also reduce manufacturing costs. The profits would be made higher if the production costs were lower.

This study shows that the way to combine GA and SBH would bring good solutions for the case study. However, more complicated factors of the real systems, which would pay much affect on the solutions, would be considered in the future research. Besides, the alternative to reduce computer time should be focused on with the larger job shop scheduling problems.

Compliance with ethical standards

Acknowledgments

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] L. G. Guohui Zhang, A genetic algorithm for flexible job-shop, IEEE international conference on robotics and automation, Vol. 2, pp. 1120-1125, 1999.

[2] L. G. Y. S. Guohui Zhang, An effective genetic algorithm for the flexible job-shop scheduling problem, Expert Systems with Applications, Vol. 38, pp. 3563-3573, 2011.

[3] PPaz Perez-Gonzaleza, Victor Fernandez-Viagasa, Miguel Zamora Garcíab, Jose M. Framinana, Constructive heuristics for the unrelated parallel machines scheduling problem with machine eligibility and setup times, Computers & Industrial Engineering 131 (2019) 131-145, Elsevier.

[4] L. M. Oleh Sobeyko, Heuristic approaches for scheduling jobs in large-scale flexible job shops, Computers & Operations Research, Vol. 68, pp. 97-109, 2016.

[5] M. D.-M. R. Tavakkoli-Moghaddam, A computer simulation model for job shop scheduling problems minimizing makespan, Computers & Industrial Engineering , Vol. 48, pp. 811-823, 2005.

[6] W. C. L. Hong Zhou, Minimizing weighted tardiness of job-shop scheduling using a hybrid genetic algorithm, European Journal of Operational Research, Vol. 194, pp. 637-649, 2009.

[7] M. L.Pinedo, Scheduling-Theory, Algorithms, and Systems, New York: Springer Science + Business Media, 2008.

[8] Kexin Sun a b, Debin Zheng c, Haohao Song c, Zhiwen Cheng d, Xudong Lang a b, Weidong Yuan a b, Jiquan Wang, Hybrid genetic algorithm with variable neighborhood search for flexible job shop scheduling problem in a machining system, Expert Systems with Applications, Elsevier, Volume 215, 1 April 2023, 119359.

[9] M. G. L. S. X. Z. Jie Gao, A hybrid of genetic algorithm and bottleneck shifting formulate objective flexible job shop scheduling problems, Computers & Industrial Engineering, Vol. 53, pp. 149-162, 2007.

[10] Z. W. Weijun Xia, An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems, Computers & Industrial Engineering , Vol. 48, pp. 409-425, 2005.

[11] B. J. P.Brucker, A Brand and Bound Algorithm for job shop scheduling, pp. 107-127, 1994.

[12] E. P. J. Carlei, An Algorithm for solving the Jop-shop Scheduling problem, Management Science, pp. 164-176, 1989.

[13] A. G. C. S. M. Ballicu, Jop-shop scheduling with set-up time, pp. 95-100, 2002.

[14] N. S. A. V. E. Balas, Jop-shop scheduling with set-up time: deadlines and precedence constrains, Theory and Application, pp. 520-532, 2005.

[15] L. M. Oleh Sobeyko, Heuristic approaches for scheduling jobs in large-scale flexible job shop, Computers & Operations Research, Vol. 68, pp. 97-109, 2015.

[16] A. M. K. a. A. A. K. Imran A. Chaudhry, A Genetic Algorithm for Flexible Job Shop Scheduling, Proceedings of the World Congress on Engineering, Vol. I, 2013.

[17] M. M. A. B. M. L. Juan C. Vidal, Machine scheduling in custom furniture industry through neuro-evolutionary hybridization, Applied Soft Computing , Issue 13, pp. 1600-1613, 2011.

[18] P. S. P. a. O. K. Chetty, GA with Adaptive Probabilities for Lot Sizing in PRP.

[19] G. K. Seyda Topaloglu, A modified shifting bottleneck heuristic for the reentrant job shop scheduling problem with makespan minimization, Manufacture Technology, Vol. 794, Issue 44, p. 781, 2009.

[20] M. &. C. R. Gen, Genetic algorithms & engineering optimization, New York: Wiley, 2000.

[21] P. &. N. M. Moscato, A memetic approach for the traveling salesman problem: Implementation of a computational ecology, Proceedings of the international conference on parallel computing and, 1992.

[22] T. V. H. L. M. V. Nam, Data collection methods in research, VOER HaiNoi Spring.

[23] N. N. Phong, Manufacturing Resource Planning, 2012.

[24] T. I. M. Binchao Chen, A flexible dispatching rule for minimizing tardiness in job shop scheduling, 2012.

[25] P. R. D. E. Hart, Evolutionary scheduling: a review, Genetic Programing and Evolvable Machines, pp. 191-220, 2005.