Indonesian Journal of Electrical Engineering and Computer Science
Vol. 27, No. 1, July 2022, pp. 496~503
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v27.i1.pp496-503

Hybrid metaheuristic to solve job sequencing problem

Gustavo Erick Anaya-Fuentes[1], Héctor Rivera-Gómez[2], Oscar Montaño-Arango[3], José Ramón Corona-Armenta[4], Enrique Martínez-Muñoz[5]
Department of Industrial Engineering, Faculty of Engineering Science, Autonomous University of the state of Hidalgo, Hidalgo, México

Article Info

Article history:

Received Jan 4, 2022
Revised May 30, 2022
Accepted Jun 5, 2022

Keywords:

Closest neighbor
Iterated local search
Job shop scheduling
Makespan
Metaheuristics

ABSTRACT

In this article, the job shop scheduling problem was solved, which is considered as an NP-Hard problem, so it is difficult to solve. The proposed algorithm consists in the combination and adaptation of the nearest neighbor metaheuristics and iterated local search (ILS) to find optimal solutions. The proposed algorithm was evaluated by comparing the results with those presented in different instances and sizes of the literature. The results of the proposed method were close in high percentages to the best results and averages obtained by other methods in the literature and in some cases, they were improved. The above represents the feasibility of a new combination of efficient methods in solving task scheduling problems.

Corresponding Author:

Gustavo Erick Anaya-Fuentes
Department of Industrial Engineering, Faculty of Engineering Science,
Autonomous University of the state of Hidalgo
Pachuca-Tulancingo km 4.5, Carboneras, Mineral de la Refoma Hidalgo, CP 42184, México
Email: gustavoerick_anay@hotmail.com

1. INTRODUCTION

Business organizations are immersed in a process of constant competition to maintain and increase market share, offering quality products at a competitive price. This can be achieved through process standardization and cost minimization strategies. The reduction of costs in the production processes is the purpose of one of the classic problems of the industry, known as job shop scheduling problem (JSSP) arises from the interest in studying the advantages offered by the use of a computer with many processors in parallel, despite the fact that a set of tasks can be processed in less time by this type of multiprocessing [1], certain anomalies were identified that affect the time required to process a set of tasks [2], classifying it as a NP-hard type combinatorial optimization problem [3]. This type of problem had already been approached using exhaustive exploration methods, such as linear programming, in the search for optimal solutions [4], however, they did not provide the expected results, due to the computational nature of the problem. The nature of the problem classified with NP – hard. Later, heuristic methods and techniques were used in the exploration of feasible solutions, showing that heuristics did not ensure optimal values, but with the advantage of finding solutions in a feasible computational time for the industry. The complexity of the problem lies in its approach, as described:

The JSSP seeks to program a finite set J of n jobs in a finite group M of m machines. Several operations are considered in each job, in addition, the order of the machines for each job is fixed and predefined [5]. The objective of the JSSP is to find an appropriate sequence, represented in a schedule that schedules the appropriate operation for all jobs that can minimize the makespan or one that minimizes the idle time of the machines [6]. The formulation of the model is as follows:

N: Number of jobs.
M: Number of machines.
Pij: Processing time for job j on m/c, j=1, 2…, N and 1, 2…M
Dj: Job expiration date j.
SEQ: Processing Sequence Matrix.
NUMT: Number of machines for each job
NUMJ: Number of jobs per machine J.
DISJ: Disjunction matrix.
Decision variables:
Cj: Completion time of work j.
Sji: Start time of job j on machine i.
Fji: Completion time of job j on machine i.
Pij: Processing time for job j on m/c, j=1, 2…, N and i=1, 2…M.
The objective function consists of minimizing the makespan by means of the equation.

$$Makespan = MAX(Cj), \forall\ j\ \in N \tag{1}$$

Subject to restrictions:
Restriction to avoid task overlaps

$$(S_{hi} - S_{hj}) \geq P_{hj} - MY_{hij}, \forall\ i, j\ \in N, \forall\ h\ \in M \tag{2}$$

Constraint to satisfy the precedence of operations

$$(S_{hj} - S_{hi}) \geq P_{hi} - M(1 - Y_{hij}), \forall\ i, j\ \in N, \forall\ h\ \in M \tag{3}$$

$$\sum_{h \in M} (S_{SEQ(j,l),j} + P_{SEQ(i,l),j)} \geq \sum_{h \in M} S_{SEQ(j,l+1),j}, \forall j\ \in N, \forall l\ \in M - 1 \tag{4}$$

In this problem, jobs J contain the elements $\{J_1, J_2, \ldots J_n\}$ and a set of machines M contains $\{M_1, M_2, \ldots, M_m\}$ [7]. Each job Jn must be done by m machines to finish the job, as seen in Figure 1. Each job consists of a set of operations and the order of operation of the machines is predefined. Each operation uses one of the machines to complete its work during a fixed time interval [8]. Once an operation is processed on a particular machine, it cannot be interrupted before the procedure ends. The sequence of operations for a job must be predetermined and can be different for any job. Each job has a sequence of operations. Each machine can process only one operation during the time interval. The goal of the JSSP is to find an appropriate sequence for all jobs, minimizing makespan or machine downtime [6].

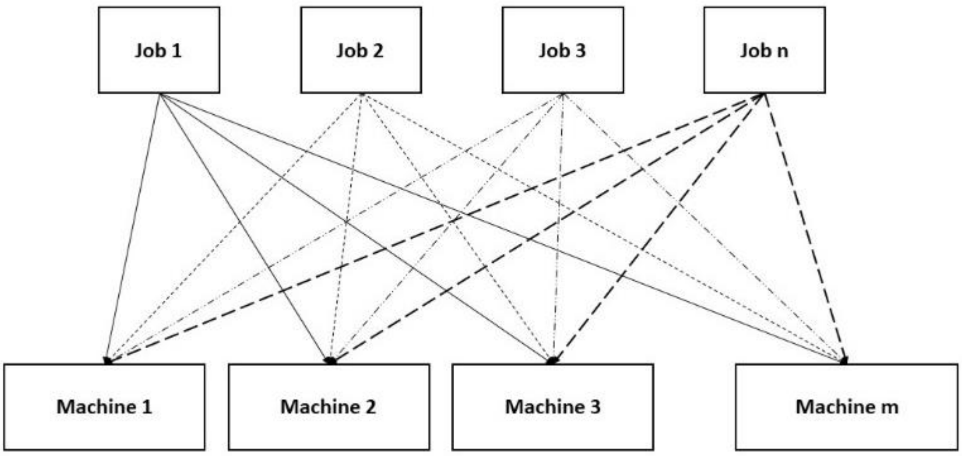


Figure 1. Job shop scheduling problem

Different methods have been proposed to solve the JSSP [9]–[13], using genetic algorithms [14], as well as the shifting bottleneck heuristic [15], which found better results in 20.69% compared to the solution provided by the first-come, first-served priority rule [16], in addition, the bat algorithm [17] is an approximation method that shows better results with respect to the particle swarm heuristic [18]–[21], when comparing the speed of convergence and the precision. Likewise, it justifies the use of approximation methods in this problem classified as NP-hard because, although exact methods guarantee optimal solutions for small problems, they are not useful for large problems [22]. The method is based on echolocation characteristics of bats, in the search to optimize the problem. Furthermore, the JSSP was solved using the neighborhood search variable method variable neighborhood search (VNS) [23] to find feasible solutions to the problem within a limit of solutions called neighborhood, showing that VNS outperforms results to other metaheuristics with which it was compared, such as genetic algorithms [24] and tabu search method [25].

In addition, the golden ball algorithm metaheuristic was used, adapting it to solve the JSSP, this technique uses soccer concepts, in the search for feasible solutions to the problem, relating concepts such as player with schedule, team with group of schedules, among others, proposing a treatment for the populations generated in which the players are assigned (Schedule) to the teams (group schedules), competing among them and using stoppage criteria. This algorithm proves to be competitive in its results when compared to other algorithms in the literature [26]. The above shows a trend in the search for new methods that provide better performance in solving the problem. In this context, this work proposes an adaptation of the nearest neighbor [27] and iterated local search (ILS) metaheuristics to find feasible solutions to the JSSP, testing the appropriate instances that allow comparing the results of this proposal with other methods from the literature [14], [17], [26], [28].

Some investigations have used the nearest neighbor methods in conjunction with repair techniques to block JSSP problems and with the intention of evaluating, generating a statistical analysis of a large set of generated neighbor solutions to detect critical characteristics of the repair scheme and the space search in general [29]. To solve the multi-objective JSSP, the nearest neighbor method was used to evaluate the density of solutions and guide the search more efficiently [30]. The closest neighbor was used to find not only the closest neighbor of distance 1 but also the closest neighbor of distances n where n>1. To do this, neighbors that are connected by routes of length n were considered [31].

The iterated local search [32] has been widely used to solve problems such as the traveling agent [33], the flow shop scheduling problem [34], inventory routing problem [35], transportation problems [36], the location problem optimal service units [37], vehicle routing problem [38] and the JSSP [39]. The main contribution of the paper is the adaptation of the closest neighbor heuristics with the Iterated Local Search in determination of efficient and feasible solutions to the indicated problem. In section 2 the research methodology is presented, in section 3 is presented the results and discussion, in section 4 the conclusion is presented.

2. RESEARCH METHOD

The present research proposes the adaptation of the nearest neighbor metaheuristics [27] and iterated local search to find feasible solutions to the job shop scheduling problem (JSSP). Unlike previous works in which a local iterated search is carried out in a neighborhood [23], using a variable, this work adapts two metaheuristics, applying the nearest neighbor method to obtain a good feasible solution used as the initial solution of the iterated local search technique, which is later applied to obtain feasible solutions to the JSSP. Diagram alir dari algoritma yang diusulkan as shown in Figure 2.

The pseudocode of the proposed algorithm is as follows:

A. Pseudocode: start
   - Encrypt the instance data.

B. Pseudocode: the closest neighbor.
   - Randomly take an instance operation.
   - Select the next operation, the one that represents the lowest cost.
   - Repeat the previous procedure until obtaining a feasible sequence.
   - Store the solution in a stack of size n.
   - Generate an initial population of size n-1.
   - Add the new population to the stack.
   - Determine the makespan of each element in the stack

C. Pseudocode: Iterated Local Search
   - Define numIter = number of iterations

- For i: 1: numIter
- Generate a new population of n individuals
- Calculate the makespan of each individual in the new population.
- Compare the makespan of each element in the new population with that of each element in the stack
- If makespan stack element> makespan population element
- Substitute the best in the pile.
- End if
- End For

D. Print to the best sequence with your makespan

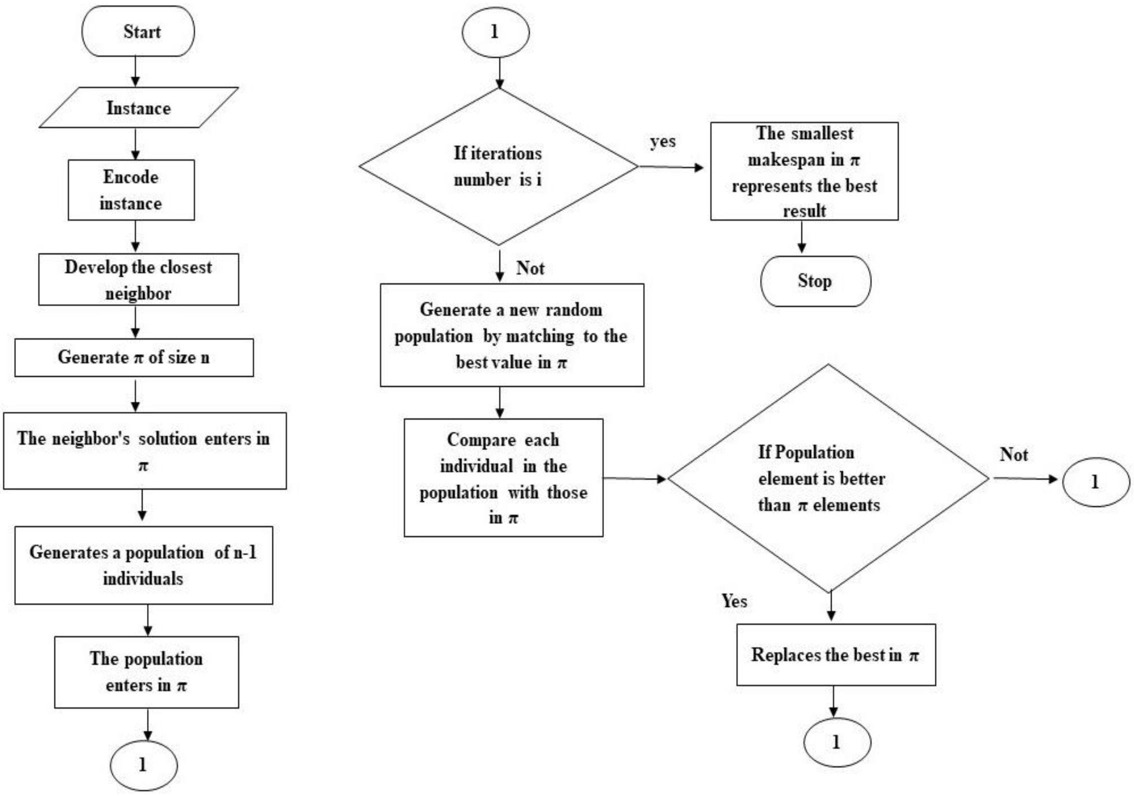


Figure 2. Flow diagram of the proposed algorithm

To compare the performance of the algorithm, 30 trials were carried out in each of the tested instances, with the intention of comparing the results obtained by other methods and techniques that are available in the literature for each instance.

3. RESULTS AND DISCUSSION

The method described in the previous section is used to solve instances of the JSSP resolved by other methods in the literature [14], [26], [28], [40]–[42]. With the intention of comparing its results with those obtained by The method proposed in the present investigation, for which 30 tests were carried out for each instance, using a computer with the following characteristics: Intel (R) Core (TM) i5-8265U processor CPU @ 1.60GHz 1.80 GHz, installed RAM 8.00 GB. The results obtained are shown in 2 tables, Table 1, presents the best results of the proposed method compared with the best results obtained by methods in the literature, while Table 2, shows the averages of each instance obtained by the proposed method, comparing them with those of literature [14], [26], [28], [40]–[42].

The results obtained through the proposed algorithm show to be competitive with those of the literature, as seen in Table 1, where the approximation of the results of the proposed algorithm was compared with the best makespan in the literature, in percentage terms, reaching 100% proximity in instances LA01, LA03 and FT06, with an average percentage of 92.5% of total algorithm approximation for the instances evaluated. The worst results reach an approximation of 76.7, 76.8 and 78.8%, for instances LA17, LA18 and FT20, respectively. On the other hand, the makespan averages of the 30 trials in each instance are compared with the averages of the instances of the different methods in the literature, as presented in Table 2, in which a 100% approximation is found in the FT06 instance, high percentages in instances such as LA02, LA03, LA05 and FT20, with an average of 87.79%. The previous results allow us to identify a high performance of the proposed algorithm, in comparison with methods proposed in the literature, being viable and feasible for its use in solving problems related to the sequencing of activities.

Table 1. Better results found in the literature compared to the results of the proposed method

Instance	Abdalla, [14]	Hegazy, [40]	Hegazy, [17]	Sayoti, [26]	Hernández, [41]	Lamos, [42]	Muthian, [28]	Proposed	Approach percentage
LA01	733	681	666	666	666	666	658	666	100.0
LA02	724	694	655	926	655	655	653	655	99.7
LA03	671	633	621	597	---	597	597	597	100.0
LA04	658	611	610	590	---	590	590	599	98.5
LA05	611	593	593	593	---	593	593	598	99.2
LA06	---	926	926	926	---	926	868	983	93.8
LA07	---	890	890	890	---	890	890	930	95.5
LA08	---	---	---	863	---	863	863	971	87.5
LA09	---	---	---	951	---	951	951	1004	94.4
LA10	---	---	---	958	---	958	958	983	97.4
LA11	---	---	---	1222	---	1222	1222	1283	95.0
LA12	---	---	---	1039	---	1039	1039	1144	89.9
LA13	---	---	---	1150	---	1150	1150	1264	90.1
LA14	---	---	---	1292	---	1292	1292	1302	99.2
LA15	---	---	---	1207	---	1207	1207	1263	95.4
LA16	---	---	---	945	945	945	941	1079	85.8
LA17	---	---	---	784	---	784	784	967	76.7
LA18	---	---	---	848	---	848	848	1045	76.8
FT06	---	55	55	55	55	55	56	55	100.0
FT10	---	1007	1004	946	938	930	925	1032	89.0
FT20	---	1242	1203	---	1165	1165	1035	1254	78.8

Table 2. Instance averages

Instance	Hegazy, [40]	Hegazy, [17]	Sayoti, [26]	Hernández, [41]	Proposed	Approach percentage
LA01	705.0	695.9	666.0	666.0	685	97.1
LA02	729.7	696.7	655.0	---	678	96.5
LA03	657.5	633.5	604.7	---	617.8	97.8
LA04	648.1	633.3	593.2	---	673.1	86.5
LA05	601.1	599.8	593.0	---	632.1	93.4
LA06	940.2	938.5	926.0	926.0	1037.4	88.0
LA07	940.1	934.9	890.0	---	981.9	89.7
LA16	---	---	952.0	950.2	1135.8	80.5
LA17	---	---	784.5	---	1016.3	70.5
LA18	---	---	856.1	---	1116.8	69.5
FT06	58.9	56.8	55.0	55.0	55.0	100.0
FT10	1086	1076.6	962.1	945.0	1123.0	81.2
FT20	1296	1283.7	---	1170.5	1280.7	90.6

The results obtained in some instances are shown below, analyzing the improvements obtained in the first 100 iterations. In Figure 3 it is possible to identify that the makespan decreases as the number of iterations of the proposed algorithm grows, reducing from 1420 to 850 units in 100 iterations, which represents a reduction of 40.14% in the LA04 instance, which seeks to minimize the makespan in sequencing 10 jobs on 5 machines.

The results of the LA12 instance are shown in Figure 4 identifying improvements in each iteration going from 2080 to a makespan of 1420 in 100 iterations, this represents an improvement of 31.73%. Figure 5 shows the makespan improvements for the LA18 instance, in this Figure an improvement from 2290 to 1400 units is observed in 100 iterations, which represents a reduction of 38.86%. The FT20 instance, which seeks to minimize the makespan of the sequencing of 20 jobs and 5 machines, reduces the results from 2330 to 1550 in 100 iterations, representing an improvement of 33.43% as shown in Figure 6.

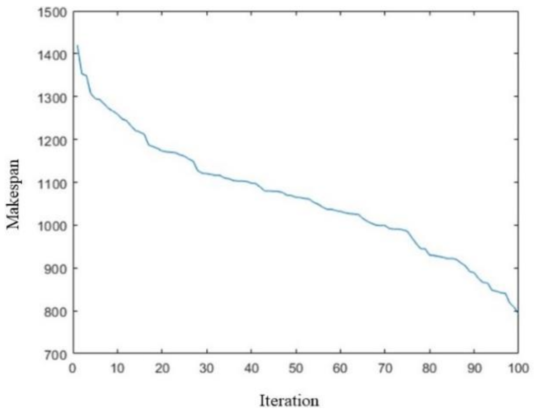


Figure 3. Makespan of the iterations in instance LA04

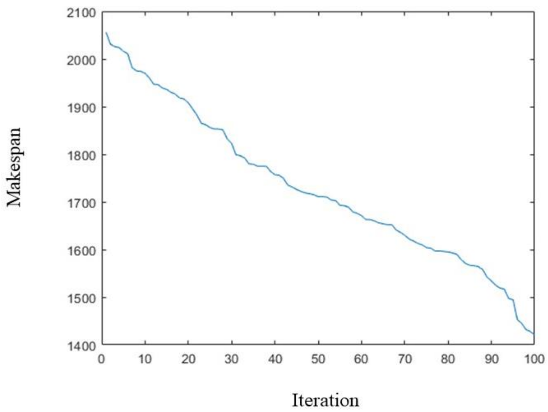
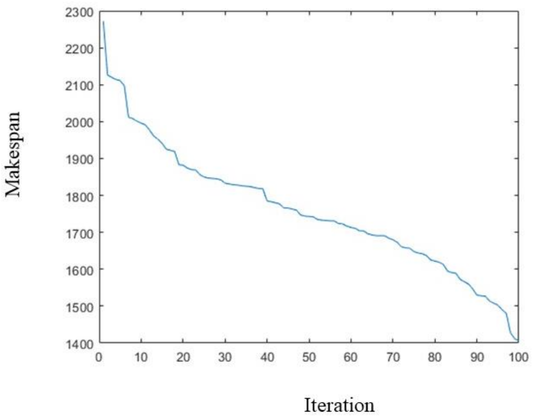


Figure 4. Iteration´s makespan in instance LA12



Fugure 5. Iteration´s makespan in instance LA18

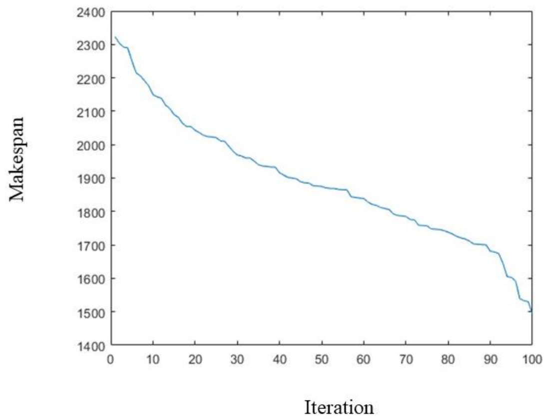


Figure 6. Iteration´s makespan in the FT20 instance

4. CONCLUSION

The results of the present investigation allow us to conclude that the proposed method is competitive with respect to different methods in the literature, because high percentages of approximation were found in accordance with the results shown in Tables 1 and 2, being better those corresponding to the minimum values found, compared with the average, in addition, it is possible to observe that as the number of iterations in the algorithm grows, the results continue to improve until finding a local optimum or a general optimum without being able to identify whether it is one or another of the above, because this method does not ensure optimization. The best results are presented in small instances, when the problem is less complex, therefore, the metaheuristics still do not ensure the optimization of the problems, but they provide good feasible results. Therefore, it is necessary to continue in the search for techniques and tools that allow greater efficiency in the results. The proposed algorithm could be improved by incorporating operators such as elitism within the process of the so-called stack, as well as optimizing the parameters according to the size of the instance to be evaluated, such as an optimal value in the population size in each iteration and the number of iterations, it is also possible to adapt the algorithm proposed with others from the literature, in the search to improve the results.

REFERENCES

[1] Baer, “Multiprocessing systems,” *IEEE Trans. Comput.*, vol. C–25, no. 12, pp. 1271–1277, 1976, doi: 10.1109/TC.1976.1674594.

[2] R. Graham, “Bounds for certain multiprocessing anomalies,” *Bell Syst. Tech. J.*, vol. 45, no. 9, pp. 1563–1581, 1966, doi: 10.1002/j.1538-7305.1966.tb01709.x.

[3] P. Bruker, “Complexity of shop-scheduling problems with fixed number of jobs: a survey,” *Math. Methods Oper. Res.*, vol. 65, no. 3, pp. 461–481, 2007, doi: 10.1007/s00186-006-0127-8.

[4] G. Dantzig, R. Fulkerson, and S. Johnson, “Solution of a large-scale traveling-salesman problem,” *J. Oper. Res. Soc. Am.*, vol. 2, no. 4, pp. 393–410, 1954, doi: 10.1287/opre.2.4.393.

[5] A. S. Jain and S. Meeran, “Deterministic job-shop scheduling: Past, present and future,” *Eur. J. Oper. Res.*, vol. 113, no. 2, pp. 390–434, 1999, doi: doi.org/10.1016/S0377-2217(98)00113-1.

[6] L. K. Sahu and S. K, "Job shop scheduling with heuristic algorithms," *Int. J. Res. Innov. Sci. Technol.*, vol. 2, no. 1, pp. 39–45, 2015.
[7] W. Wang and P. Brunn, "An effective genetic algorithm for job shop scheduling," in *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.*, 2000, vol. 214, no. 4, pp. 293–300, doi: 10.1243/0954405001517685.
[8] P. Sharma and A. Jain, "A review on job shop scheduling with setup times," *Inst. Mech. Eng. Part B J. Eng. Manuf.*, 2015.
[9] E. Gafarov and F. Werner, "Two-machine job-shop scheduling with equal processing times on each machine," *Math. MDPI*, vol. 7, no. 301, pp. 1–11, 2019, doi: 10.3390/math7030301.
[10] J. Alzahrani, "Job-shop scheduling optimization with stochastic processing times," *Int. J. Eng. Technol. Manag. Res.*, vol. 6, no. 1, pp. 73–83, 2019, doi: 10.29121/ijetmr.v6.i1.2019.347.
[11] B.-A. Han and J.-J. Yang, "Research on adaptive job shop scheduling problems based on dueling double DQN," *Inst. Electr. Electron. Eng.*, vol. 8, pp. 186474–186495, 2020, doi: 10.1109/ACCESS.2020.3029868.
[12] S. Bourki, M. Essaid, and F. Chebihi, "Memetic chicken swarm algorithm for job shop scheduling problem," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 3, pp. 2075–2082, 2019, doi: 10.11591/ijece.v9i3.pp2075-2082.
[13] S. Sundari, "Makespan minimization in job shop scheduling," *Int. J. Eng. Manag. Res.*, vol. 11, no. 1, pp. 228–230, 2021, doi: 10.31033/ijemr.11.1.31.
[14] A. O. Dagroum, H. Elaswad, and A. Jabuda, "Genetic algorithm for minimizing the makespan in job shop scheduling with machine setup issue," *Int. Res. J. Eng. Technol.*, vol. 8, no. 2, pp. 1994–1997, 2021.
[15] J. Kassu and B. Eshetie, "Job shop scheduling problem for machine shop with shifting heuristic bottleneck," *Glob. J. Res. Eng. J Gen. Eng.*, vol. 15, no. 1, pp. 20–26, 2015.
[16] H. Tesfaye, "Optimization of job shop scheduling problem," *J. Mech. Civ. Ind. Eng.*, vol. 1, no. 1, pp. 14–20, 2020.
[17] H. Zaher, M. El, N. R. Saied, and H. Sayed, "Bat algorithm for job shop scheduling problem," *J. Multidiscip. Eng. Sci. Technol.*, vol. 4, no. 2, pp. 6758–6763, 2017.
[18] D. Lei, "A Pareto archive particle swarm optimization for multi-objective job shop scheduling," *Comput. Ind. Eng.*, vol. 54, no. 4, pp. 960–971, 2008, doi: 10.1016/j.cie.2007.11.007.
[19] T.-L. Lin *et al.*, "An efficient job-shop scheduling algorithm based on particle swarm optimization," *Expert Syst. Appl.*, vol. 37, no. 3, pp. 2629–2636, 2010, doi: 10.1016/j.eswa.2009.08.015.
[20] L. Zhigang, J. Bin, and G. Xingsheng, "A similar particle swarm optimization algorithm for job-shop scheduling to minimize makespan," *Appl. Math. Comput.*, vol. 183, no. 2, pp. 1008–1017, 2006, doi: 10.1016/j.amc.2006.05.168.
[21] P. Pongchairerks, "A self-tuning PSO for job-shop scheduling problems," *Int. J. Oper. Res.*, vol. 19, no. 1, pp. 96–113, 2014, doi: 10.1504/IJOR.2014.057848.
[22] K. Kosovskii, T. Kosovskaya, and N. Kosovskii, "NP completeness conditions for verifying the consistency of several kinds of systems of linear diophantine discongruences," *Mathematics*, vol. 49, no. 1, pp. 25–31, 2016, doi: 10.3103/S1063454116010088.
[23] M. Sevkli and E. Aydin, "Variable neighbourhood search for job shop scheduling problems," *J. SOFTWARE*, vol. 1, no. 2, pp. 34–39, 2006, doi: 10.4304/jsw.1.2.34-39.
[24] A. Rahmani, J. Vahidi, B. Saemi, A. Sangaiah, and M. Elhoseny, "Extended genetic algorithm for solving open-shop schedulingproblem," *Soft Comput.*, vol. 23, pp. 5099–5116, 2018, doi: 10.1007/s00500-018-3177-y.
[25] É. Taillard, "Parallel taboo search techniques for the job shop scheduling problem," *J. Comput.*, vol. 6, no. 2, pp. 108–117, 1994, doi: 10.1287/ijoc.6.2.108.
[26] F. Sayoti, M. Riffi, and H. Labani, "Optimization of makespan in job shop scheduling problem by golden ball algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 4, no. 3, pp. 542–547, 2016, doi: 10.11591/ijeecs.v4.i3.pp542-547.
[27] N. Bhatia and A. Vandana, "Survey of nearest neighbor techniques," *Int. J. Comput. Sci. Inf. Secur.*, vol. 8, no. 2, pp. 302–305, 2010.
[28] A. Muthiah and R. Rajkumar, "A novel algorithm for solving job-shop scheduling problem," *MECHANIKA*, vol. 23, no. 4, pp. 610–616, 2017, doi: 10.5755/j01.mech.23.4.14055.
[29] J. Lange and F. Werner, "On neighborhood structures and repair techniques for blocking job shop scheduling problems," *Algorithms MDPI*, vol. 12, no. 11, pp. 1–28, 2019, doi: doi:10.3390/a12110242.
[30] M. Frutos and F. Tohmé, "Evolutionary multi-objective scheduling procedures in non-standardized production processes," *Dyna rev.fac.nac.minas*, vol. 79, no. 172, pp. 101–107, 2012, [Online]. Available: http://www.scielo.org.co/pdf/dyna/v79n172/a11v79n172.pdf.
[31] R. Gupta, "On the Spaced repetition scheduling problem," *Int. J. Res. Eng. Sci. Manag.*, vol. 2, no. 10, pp. 231–233, 2019.
[32] H. Lourenço, M. Oliver, and S. Thomas, "Iterated local search," in *Handbook of Metaheuristics*, pp. 321–353, 2003.
[33] C. Rey, P. Toth, and D. Vigo, "An iterated local search for the traveling salesman problem with pickup, delivery and handling costs," *2020 39th Int. Conf. Chil. Comput. Sci. Soc.*, 2020, pp. 1–8, doi: 10.1109/SCCC51225.2020.9281164.
[34] J. Belabid, S. Aqil, and K. Allali, "Solving permutation flow shop scheduling problem with sequence-independent setup time," *J. Appl. Math.*, vol. 2020, pp. 1–11, 2020, doi: 10.1155/2020/7132469.
[35] A. Alvarez, P. Munari, and R. Morabito, "Iterated local search and simulated annealing algorithms for the inventory routing problem," *Int. Trans. Oper. Res.*, vol. 25, no. 6, pp. 1785–1809, 2018, doi: 10.1111/itor.12547.
[36] O. Cosma, P. Pop, and C. Sitar, "An efficient iterated local search heuristic algorithm for the two-stage fixed-charge transportation problem," *Carpathian J. Math.*, vol. 35, no. 2, pp. 153–164, 2019, doi: 10.37193/CJM.2019.02.04.
[37] R. Ríos and R. Dagoberto, "Optimal location of capacitated service units through a metaheuristic optimization approach," *Nov. Sci.*, vol. 9, no. 19, pp. 153–164, 2017.
[38] J. Walker, G. Ochoa, M. Gendreau, and E. Burke, "Vehicle routing and adaptive iterated local search within the HyFlex hyper-heuristic framework," *Learn. Intell. Optim. 6th Int. Conf.*, vol. 7212, 2012, doi: 10.1007/978-3-642-34413-8_19.
[39] N. Nouri and T. Ladhar, "An efficient iterated greedy algorithm for the makespan blocking flow shop scheduling problem," *Polibits*, no. 53, pp. 91–95, 2016, doi: 10.17562/PB-53-10.
[40] H. Zaher, M. El-Sherbieny, N. Ragaa, and H. Sayed, "Bat algorithm for job shop scheduling problem," *J. Multidiscip. Eng. Sci. Technol.*, vol. 4, no. 2, pp. 6758–6763, 2017.
[41] L. Hernández-Ramírez, J. Frausto-Solis, G. Castilla-Valdez, J. J. González-Barbosa, D. Terán-Villanueva, and L. Morales-Rodríguez, "Problem, a hybrid simulated annealing for job shop scheduling," *Int. J. Comb. Optim. Probl. Informatics*, vol. 10, no. 1, pp. 6–15, 2019.
[42] H. Lamos-Díaz, K. Aguilar-Imitola, Y. T. Pérez-Díaz, and S. Galván-Nuñez, "A memetic algorithm for minimizing the makespan in the job shop scheduling problem," *Rev. Fac. Ing.*, vol. 26, no. 44, pp. 113–123, 2017, doi: 10.19053/01211129.v26.n44.2017.5776.

BIOGRAPHIES OF AUTHORS

Gustavo Erick Anaya-Fuentes obtained bachelor's degree as Industrial Engineer, master's degree in industrial engineering and PhD degree in Industrial Engineering at Autonomous University of the State of Hidalgo in 2007, 2013 and 2017, respectively. He is a Research Scientist at the Autonomous University of Hidalgo in the Department of Engineering. He is a member of SNI, the National System of Researchers of Mexico. His research interest includes optimization, productivity, logistics and production planning. He can be contacted at email: gustavoerick_anay@hotmail.com.

Hétor Rivera-Gómez received the Ph.D. degree in Engineering from the Ecole du Technologie Superieure, Montreal, Canada. He is a Research Scientist at the Autonomous University of Hidalgo in the Department of Engineering. He is the author of several papers in international journals. He is a member of SNI, the National System of Researchers of Mexico, and the SMIO, the Mexican Society of Operational Research. His research interest includes performance analysis, simulation and control of production systems, and dynamic programming in hybrid (discrete and continuous state). He can be contacted at email: hriver06@hotmail.com.

Oscar Montaño-Arango received a doctorate degree in engineering with a specialty in Planning Systems in 2007, at the National Autonomous University of Mexico, from that year he began to work as a Professor-Researcher at the Autonomous University of the State of Hidalgo. Since 2013 it belongs to the National System of Researchers. His current research interests are in systems analysis and diagnostics, maturity models, and supply chains. He has published articles in national and international journals in the fields of planning, analysis and optimization of organizational systems. He can be contacted at email: omontano@uaeh.edu.mx.

José Ramón Corona-Armenta obtained bachelor´s degree as Civil Engineer at Technological Institute of Pachuca (I.T.P.). Master's degree in engineering, specialization in Operations Research, from the National Autonomous University of Mexico (U.N.A.M.). PhD in Industrial Engineering Systems at National Polytechnic Institute of Lorraine (I.N.P.L.) in France. He is a Research Professor at Autonomous University of the State of Hidalgo since 2005. He has held positions as a civil servant within the same University. His research projects revolve mainly in the use of multi-criteria techniques for decision-making in organizational processes, in technological innovation, planning, project evaluation, sustainable development, management of hydraulic resources and foresight. He has been members of various scientific and research associations. He has various research works including congresses, conferences, lectures, scientific and popular articles, books and book chapters, among others. He can be contacted at email: jrcorona@uaeh.edu.mx.

Enrique Martínez-Muñoz obtained bachelor's degree as Industrial Engineer at Autonomous University of the State of Hidalgo in 1985, he obtained a Master's degree in Organization Administration from the National Autonomous University of Mexico (UNAM) in 2008, he is a PhD in Administrative Sciences from the Faculty of Accounting and Administration of the Autonomous University of Tlaxcala FCA / UAT in 2012, he has carried out diplomas in the Government of Spain among them: Strategic Planning, Communication and Political Marketing, Diploma in Public Management, and in the Institute of Public Administration (IAPH). He is a Research Scientist at the Autonomous University of Hidalgo in the Department of Engineering. He belongs to the National System of Researchers Conacyt 2020-2022, has the recognition of the desirable profile 2021-2023, Specialist in Administration and Business Management Systems. He can be contacted at email: emmunoz@uaeh.edu.mx.