The Scientific Journal of University of Benghazi

www.sjuob.uob.edu.ly

SJUOB 1988

Hybrid Genetic Simulated Annealing Algorithm for Job Shop Scheduling Problem

Sumaia E. Eshim [1*], Amer A. Boushaala [1], Mohammed M.Hamed [1]

[1] *Department of Industrial Engineering and Manufacturing Systems Benghazi University, Benghazi, Libya.*

Received: 23 / 11 / 2020; Accepted: 31 / 12 / 2020

الملخص:

مشكلة جدولة أعمال الورش (JSSP) هي مشكلة تحسين حسابي صعبة ومعروفة ، حيث صُنفت على أنها مشكلة صعبة ومعقدة (NP) وبالتالي لا يمكن لأي خوارزميات محددة حلها في فترة زمنية معقولة. الهدف الرئيسي من حل هذه المشكلة هو إيجاد التسلسل المناسب للوظائف على الآلات من أجل تحسين معايير الأداء.

في هذه الدراسة، قُدِمَ أسلوب بحثي meta-heuristic لحل مشكلة جدولة أعمال الورش (JSSP). ويستخدم هذا الأسلوب الخوارزمية الوراثية الهجينة التي اقترحت في دراسات سابقة لتوليد أفضل الحلول ومن ثم استخدام خوارزمية التلدين المحاكية لتحسين جودة وأداء أفضل الحلول من أجل إنتاج الحل الأمثل / شبه الأمثل. تبنت الدراسة الحالية حل (10) مشاكل معيارية استخدمت في الدراسات السابقة لتقييم أداء الخوارزمية المقترحة.

من خلال النتائج الحسابية، تم التحقق من صحة وجودة الخوارزمية المقترحة وذلك عن طريق حساب زمن انتهاء الوظائف على الآلات C_{max} .

الكلمات المفتاحية:

جدولة أعمال الورش، الخوارزمية الوراثية الهجينة ، الخوارزمية الوراثية، خوارزمية التلدين المحاكية.

Abstract

The job shop scheduling problem (JSSP) is a well-known difficult combinatorial optimization problem, as it is classified as NP-hard problem and therefore no deterministic algorithms can solve them in a reasonable amount of time. The main objective of solving this problem is to find suitable job sequences on machines to optimize the performance criteria. In this paper, a meta-heuristic approach for solving the job-shop scheduling problem (JSSP) is presented. This approach uses a hybrid genetic algorithm that is suggested by previous stuody, to generate the best solutions and then a simulated annealing algorithm to improve the quality and performance of the best solutions to produce the optimal/near-optimal solution. Ten benchmark problems adopted from the previous study are used to evaluate the performance of the proposed algorithm. The computational results validate the quality of the proposed algorithm; this is done by calculating the completion time C_{max} .

Keywords: Job shop scheduling; makespan; hybrid algorithm; genetic algorithm; simulated annealing.

1. INTRODUCTION

Scheduling is a decision-making practice used regularly in most manufacturing industries. It aims to optimize the objectives with the allocation of resources to tasks within the given time-periods. The resources and tasks in an organization can take a lot of different forms. The resources may be machines in a workshop, processing units in a computing environment and so on. The tasks may be jobs or operations in a production process, executions of computer programs, stages in a construction project, and so on. The objectives can take many different forms and one objective may be the minimization of total completion time of jobs (1).

Scheduling concerns with searching for an optimal or satisfying solution subject to several constraints. Achieving the best schedule may be very easy or very difficult depending on the shop environment, the process constraints, and the performance indicators. The scheduling problems depend on the shop floor environment, such as flow shop, job shop, and open shop. For each environment, a different scheduling approach is needed (2). Job shop scheduling is one of the most typical and complicated tasks in scheduling problems. It has attracted many researchers due to its wide applicability and inherent difficulty. As mentioned by Zandieh et al (3), (Nowicki and Smatnicki,1996) and (Jain and Meeran, 1999), defined the Job Shop Scheduling Problem (JSSP), is one of the hardest combinatorial optimization problems, as it is classified as NP-Hard problem and therefore no deterministic algorithms can solve them in a reasonable amount of time. Researchers from all over the world have conducted in-depth research on this issue. The approaches that are used to solve scheduling problems can be broadly classified under two categories: exact and approximate. The exact methods include mathematical techniques and branch-and-bound methods. The most common mathematical formulation of JSSP is integer programming, mixed-integer programming, and dynamic programming. The mathematical methods give optimal solutions but are generally applied to small problems. These methods are guaranteed to find the optimal solution, but the time required to solve the problem makes the solution highly impractical (4). All exact methods cannot solve larger problems in a reasonable time. Therefore, the focus of research shifted to approximation-based methods(5). Approximation methods, as evident from their name, provide no guarantee of an optimal solution, but they can provide the near-

*Correspondence:

Sumaia E. Eshim
sumaiaeshim@gmail.com

optimal solution for scheduling problems. Some common approximation methods are priority dispatching, local search, and metaheuristic methods.

In priority dispatching, methods specify which job should be selected for processing next from among a queue of jobs, whether scheduling or during real-time processing. When a machine or worker becomes available, the priority rule is applied and the next job is selected. Jobs can be scheduled on machines taking into consideration certain rules, depending on the final objective or purpose for which a schedule is being generated. But these methods do not focus on reducing makespan. Therefore, the efficacy of these methods greatly reduces larger problems [(6)]. Researchers in [(7,8)] claimed that priority dispatching rules can be successfully used in solving JSSPs and even other scheduling problems. Also, several dispatching rules present a significant optimization capacity.

Local search methods [(6)] are another approximation-based techniques to solve JSSP. The basic theme of these techniques is to modify existing solutions by searching in the neighborhood of this solution. A neighborhood operator is used to generate a new solution that is expected to be better than its seed.

Genetic Algorithm (GA) was first presented by Holland (1975) as mentioned by Zandeh et al [(3)], Genetic algorithm has also demonstrated potential for solving job-shop scheduling problem. Genetic algorithms (GAs) have proven their efficiency in solving high complexity combinatorial optimization problems, thus many researchers have applied them also to scheduling problems [(9,10,11)]. The implementation time of the GA can be defined as the time required by the algorithm to render an optimal or satisfying solution. This time reflects the solution quality comprising each generation. If the quality of the solutions is poor, i.e. the individuals are beyond the fitness function or imposed constraints, then the results seem to be hopeful, but, the GA will take more time to render or reach the best solution. Some techniques and operators are used to improve the solution quality. The performance of GAs solutions depends on the quality of the initial population [(12,13)]on which the quality and performance of the next population's generations will depend. All previous algorithms and approaches, there are positives and negatives of every optimization method. Therefore, combinations of two or more techniques are used to solve JSSP. These methods are called hybrid algorithms. Hybrid methods are frequently employed for solving JSSP for example hybrid GA and heuristic rules [(14)], hybrid GA and local search[(15)], and GA with simulated annulling [(10,16)]. The hybrid algorithms perform better than their corresponding individual counterparts as with hybridization convergence rate is usually high and it also helps in escaping local minima. All the algorithms developed for JSSP have their strengths and weaknesses [(6)]. Therefore, researchers are constantly in search of new algorithms and a lot of efforts have been put in to optimize and improve existing methods. This article focuses on this fact and tries to use a hybrid genetic algorithm proposed by (Boushaala, et. al., 2013) [(14)], for generating the best solutions for JSSP and then used these to generate solutions as an initial solution for the simulated annealing algorithm.

The rest of the paper is organized as follows; Section 2 covers the conventional problem definition. Section 3 provides details of the proposed hybrid method. Section 4 discusses the implementation of the algorithm and its results on benchmark problems followed by a conclusion in the last section.

2. PROBLEM DEFINITION

The job shop scheduling problem (JSSP) can be described as follows: a set of (n)jobs (J) to be scheduled on a set of (m) machines (M). Each job (j) visits the number of machines in a predetermined order. The processing times for each job at each machine are given and no machine can process more than one job at a time. If a job is started on a machine, then it cannot be interrupted. The problem is finding a schedule of the jobs on the machines. The assumptions of the present problem are:

- Every job has a unique sequence of (m) machines. There are no alternate routings;
- There is only one machine of each type in the shop;
- Processing times for all jobs are known and constant;
- All jobs are available for processing at time zero;
- Machine absences are not allowed;
- Transportation time between machines is zero;
- Each machine can perform only one job at a time;
- Each job visits each machine only once;
- An operation of a job can be performed by only one machine;
- The operation cannot be interrupted;
- A job does not visit the same machine twice;
- An operation of a job cannot be performed until its preceding operations are completed;
- There is no restriction on queue length for any machine;
- There are no limiting resources other than machines/workstations;
- The machines are not identical and perform different operations [(14)].

The objective of the scheduling job is to optimize a certain criterion. These criteria are used as a performance measure of the schedule. A common objective criterion is to minimize the makespan (C_{max}), which is the time needed to complete all the jobs.

3. Hybrid Genetic Simulated Annealing Algorithm

3.1 Overview of the HGSA Algorithm

The main purpose of the research is to explore a more effective and efficient approach to solve scheduling problems in job shops. The most prominent feature of our algorithm is a combination of a hybrid genetic algorithm (HGA) based on some recommended heuristics and developed heuristic rules as a first population for the genetic algorithm (used to generate the initial individuals) with simulated annealing, which can effectively guide and speed up the solution search. The resulted solutions obtained using a hybrid genetic algorithm (HGA) are considered the first initial solution of the presented simulated annealing algorithm is presented.

3.2 Hybrid Genetic Algorithm (HGA)

In this paper, a hybrid genetic algorithm (HGA) based on an integration of a genetic algorithm and some developed and recommended heuristic rules is presented. The flowchart of a hybrid Genetic algorithm is shown in Figure 1.

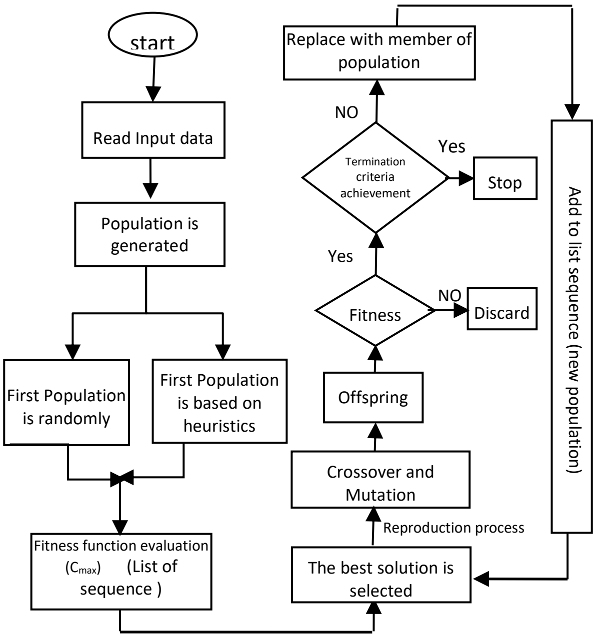

Fig:1 A hybrid Genetic algorithm flow chart

3.2.1. Solutions Representation

The first step in solving optimization problems with a genetic algorithm is the representation of problem solutions as a chromosome. The representation of a chromosome follows the genetic representation as mention by Boushaala et al [14], where each gene corresponds to one of (n) jobs and a chromosome corresponds to a solution (a sequence of jobs). The chromosome length is the sum of all the operations on all jobs. A string of triples (J, I, M) is used, one for each operation, where (J) is the job that operation belongs to, (I) is the progressive number of that operate within the job (j), and (M) is the machine assigned to that operation.

3.2.2. Solution Initialization Process

The first population currently set to (27) solutions are generated in two scenarios, in the first one, the initial population is randomly generated for each instance based on JSSP domain search space. In the second scenario, the initial population is generated based on the results obtained of twenty-seven heuristic rules. This allows the capability of the proposed genetic algorithm (GAs) for vastly obtaining a good initial set of feasible schedules (individuals). All the solutions in the first population have not the same sequence order to maintain the diversity of individuals.

3.2.3. Fitness function

Makespan is used to determine chromosome fitness. The chromosome with the least fitness function value is the most desired or considered as the best-rendered solution.

3.2.4. Selection Operator

The proposed genetic algorithm uses a ranking strategy for chromosome selection; the population is ranked according to the fitness values of its members. Two individuals based on minimum make span are selected from the population in the reproduction process. This is a process used to determine the number of trials for one particular individual used in reproduction. This means that all individuals in the population have the chances to reproduce. In each generation, a selection scheme is used to select the survivors for the next generation according to their fitness values.

3.2.5. Crossover Operator

a single-point crossover and multi-point crossover are applied in this paper. Both the applied crossover operators are selected randomly for each instance and remain constant. After the crossover is done, the new part chromosomes are checked for the validity of being the newcomer. If not, then, a repairing action is performed. This process is done to assure that the resulting string presents a complete sequence for processing the jobs.

3.2.6. Mutation Operator

To maintain the genetic diversity from one generation to the next, the offspring solutions are mutated. The reciprocal exchange mutation is applied for the GA with a probability of 0.3 in this paper.

3.2.7. Termination criterion

The most common method is to stop after a predetermined number of generations have been evaluated. Another commonly used method is to terminate when the solution crosses the desired threshold or fitness value. In this paper 200 generations are used as a stopping criterion where it was a balanced level and no more enhancements can be achieved. The final and the most important step in the proposed algorithm is to choose the individual to be replaced by the child. In this paper, if the child has the same fitness value as the member of the population, it is considered as the worst individual in the population. In this case and instead of dropping that child, it is replaced by the old one to give another chance for the older individual to reproduce. However, if the makespan for the child is less than the worst one and not equal to any member of the population, replace the worst individual with child'. If a member is having the same makespan value, then replace it by the member with the child.

3.3 Simulated annealing

Simulated Annealing (SA) is a local-search-based meta-heuristic that has exhibited some promise when it is applied to NP-hard problems. A typical SA starts from an initial solution and proceeds sequentially and slowly toward the area that might be far from the search area of the initial solution. More information in paper [18]. The flowchart of a hybrid genetic simulated annealing (HGSA) is shown in Figure 2.

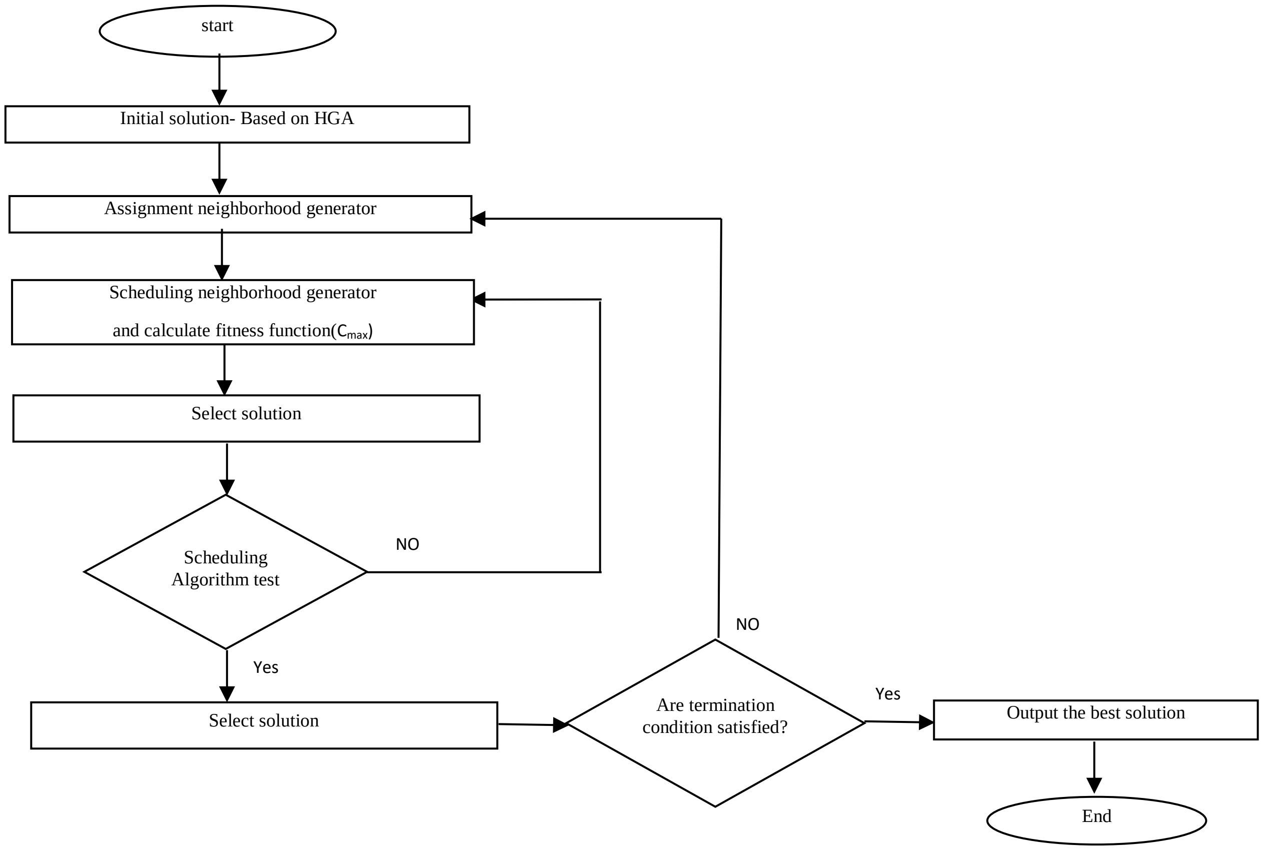

Fig:2. Algorithm flowchart of the hybrid genetic simulated annealing (HGSA) algorithm

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

To test the performance of the HGSA algorithm for solving JSSP, ten benchmark problems are tested and compared with other algorithms. The problems are of 5×5 (N×M represents N jobs and M machines). The experiment was conducted to illustrate the difference between the makespan obtained using the random GA and HGA as mentioned by Boushaala et al [(14)] The present experiment is conducted to improve the resulted solutions obtained using the hybrid genetic algorithm (HGA) according to some of benchmarks problem are considered the first initial solution of the presented simulated annealing algorithm. Two makespans of the experiment are shown in table 1. In this table, the first column presents the number of benchmark problems (BM). The second column presents a makespan from a genetic algorithm based on randomization population (GA-R) while the third column presents a makespan from a hybrid genetic based on a simulated annulling algorithm (HGSA). The makespan of the third is different from the second. The quality of the solution is better because is the makespan of the third is shorter than it in most the cases.

Table 1. Comparison of results of makespan

BM.	GA-R	HGSA
1	30	32
2	38	37
3	21	25
4	33	31
5	37	36
6	35	34
7	35	34
8	18	18
9	35	34
10	39	35

The above table demonstrates the comparison of the results obtained using the HGSA for benchmarks BM1–BM10 and genetic algorithm-based random. The analysis indicates that the results obtained using the HGSA algorithm were the most satisfactory.

The results obtained using HGSA and GA-R algorithms were compared, the results obtained using these two algorithms were identical in BM8. For BM1 and BM3 the makespan obtained using the GA-R algorithm were 30 and 21while using the HGSA were 32 and 25, respectively. Thus, the GA-R algorithm is superior to the HGSA in this scenario. However, for BM2, BM4, BM5, BM6, BM7, BM9, and BM10, the makespan obtained using the GA-R algorithm were 38, 33,37, 35,35,35 and 39 respectively, and those obtained using the HGSA were, 37, 31,36, 34, 34,34 and 35 respectively, these results indicating that the HGSA outperformed the GA-R algorithm.

5. Conclusion

The job shop scheduling problem (JSSP) is a well-known difficult combinatorial optimization problem, as it is classified as an NP-hard problem and therefore no deterministic algorithms can solve them in a reasonable amount of time. The main objective of this paper is to minimize the makespan of the job shop scheduling problem. A hybrid genetic simulated annealing algorithm (HGSA) for job shop scheduling problems is proposed in this paper. In the proposed algorithm, to obtain high-quality solutions, a hybrid algorithm based on an integration of a genetic algorithm and some developed and recommended heuristic rules, was used to generate the initial solution of the presented simulated annealing algorithm. The proposed algorithm is tested on a set of benchmark problems taken from a previous study and compared with another algorithm. It is found that the proposed algorithm can improve the solution obtained by a hybrid genetic algorithm.

6. References

1. Seda, M., “Mathematical Models of Flow Shop and Job Shop Scheduling Problems”, World Academy of Science, Engineering and Technology, 31, (2007).
2. Dorndorf, U., Pesch, E. , and Phan-Huy,T., “ Solving the Problem of Open Shop Scheduling” , Journal of Scheduling, Vol. 4, pp. 157 – 174, (2001).
3. Zandieh, M., Mahdavi, I., and Bagheri, A., “Solving the Flexible Job- Shop Problem by a Genetic Algorithm”, Journal of Applied, 8(24); PP.4650- 4655, (2008).
4. Lei, D., "Solving fuzzy job shop scheduling problems using random key genetic algorithm", Springer-Verlag London Limited 2009 Int J Adv Manuf Technol 49:253–262 DOI 10.1007/s00170-009-2379, (2010).
5. Akram ,K., Kamal, K., Zeb, A., "Fast simulated annealing hybridized with quenching for solving job shop scheduling problem", Applied Soft Computing, 49 , 510–523, (2016).
6. Qiu, X., Lau, H., "An AIS-based hybrid algorithm for static job shop scheduling problem" J. Intell. Manuf. 25 (3), 489–503, (2014).
7. Baker, K, Introduction to sequencing and scheduling, copyright by John Wiley & Sons, Inc. Duke university USA, ISBN 0-471-04555-1, (1974).
8. Kenneth R. Baker, Dan, Principle of Sequencing and Scheduling, Trietsch, Copyright by John Wiley & Sons, Inc. USA, ISBN 978-0-470-39165-5, (2009).
9. Rafinejad, S., Ramtin , F., Arabani , A., “A New Approach to Generate Rules in Genetic Algorithm Solution to a Job Shop Schedule by Fuzzy Clustering”, Proceedings of the World Congress on Engineering and Computer Science Vol. II, San Francisco, USA, (2009).
10. Tamilarasi, A., kumar, T, “An Enhanced Genetic Algorithm with Simulated Annealing for Job-Shop Scheduling”, International Journal of Engineering, Science and Technology Vol. 2, No.1, pp.144-151, (2010).
11. Kemppainen, K., “Priority Scheduling Revisited Dominant Rules, Open Protocols, and Integrated Order Management", Helsinki School of Economics, (2005).
12. Jain, A., Meeran, S., “Deterministic Job-Shop scheduling: past, present and future”, European Journal of Operational Research, 113 (2), 390−434, (1999).
13. Lin, S., Goodman, E., Punch, W., “A Genetic Algorithm Approach to Dynamic Job Shop Scheduling Problems”, Proceedings of the Seventh International Conference on Genetic Algorithms, PP. 481−488, (2003).
14. Boushaala, A. A., Shouman, M. A., and Esheem , S., “Genetic Algorithm based on Some Heuristic Rules for Job Shop Scheduling Problem” Proceedings of the 2012 International Conference on Industrial Engineering and Operations Management Istanbul, Turkey, (2012).
15. Kalshetty, R., Y., Adamuthe, R., Y., and Kumar, S., P., “Genetic Algorithms with Feasible Operators for Solving Job Shop Scheduling Problem”, Journal of Scientific Research, Volume 64, Issue 1, India, (2020).
16. Hongjing, W., Li, S., Jiang, H., Hu, J., Hu, J., “Hybrid Genetic Simulated Annealing Algorithm for Improved Flow Shop Scheduling with Makespan Criterion”, Appl. Sci. 8, 2621; doi:10.3390/app8122621, (2018).
17. Norozi, A., Ariffin, N., and Ismai l., N., “Application of Intelligence Based Genetic Algorithm for Job Sequencing Problem on Assembly Line ", ISSN 1941-7020, American J. of Engineering and Applied Sciences 3 (1): 15-24, (2010).
18. Alfakhri, M. M., "Solve job shop scheduling problem with sequence dependent setup times using a simulated annealing algorithm with different neighborhood search structure" Journal Libyan The seventh issue May, (2016).