

ISSN 2185-3118

INTERNATIONAL JOURNAL OF INTELLIGENT ENGINEERING AND SYSTEMS

Volume 14 Issue 2 April 2021



Intelligent Networks and Systems Society

DAFTAR ISI

1. Halaman pengesahan
2. Detail jurnal
3. Artikel final yang sudah dipublikasi
4. *Cover letter* jurnal
5. Komentar editor terkait penerimaan artikel untuk dipublikasikan
6. Review pertama dari reviewer jurnal
7. Lampiran Jawaban penulis untuk Review pertama dari reviewer jurnal
8. Lampiran Tracing result perbaikan artikel
9. Lampiran Hasil pengecekan plagiasi artikel
10. Review kedua dari reviewer jurnal
11. Email dari editor terkait *Acceptance Letter* Artikel
12. Lampiran *Acceptance Letter*
13. Lampiran tagihan pembayaran publikasi artikel
14. *Copyright Statement* penulis
15. *Invoice* pembayaran publikasi artikel
16. Email dari editor terkait artikel yang siap dipublikasikan

1

**HALAMAN
PENGESAHAN**

**HALAMAN PENGESAHAN
HASIL PENELITIAN**

Judul : Performance Evaluation of Various Heuristic Algorithms To Solve Job Shop Scheduling Problem (JSSP)

Penulis : Admi Syarif, Ade Pamungkas, Renaldi Kumar, Mitsuo Gen

NIP : 196701031992031003

Instansi : Jurusan Ilmu Komputer, FMIPA, Universitas Lampung

Publikasi : International Journal of Intelligent Engineering and Systems (IJIES)
Vol. 14, No. 2, pp. 334-343, April 2021

Alamat Web (Link) : <https://doi.org/10.22266/ijies2021.0430.30>
<http://repository.lppm.unila.ac.id/id/eprint/>

Penerbit : Intelligent Networks and Systems Society

ISSN : 2185-3118 (eISSN), 2185-310X (pISSN)

Indeks : Scopus: Q2 (2019), SJR 2019 : 0.2

Jenis Publikasi : Jurnal Internasional Bereputasi



Dekan Fakultas MIPA Unila

Dr. Eng. Supto Dwi Yuwono, M.T.
NIP. 197407052000031001

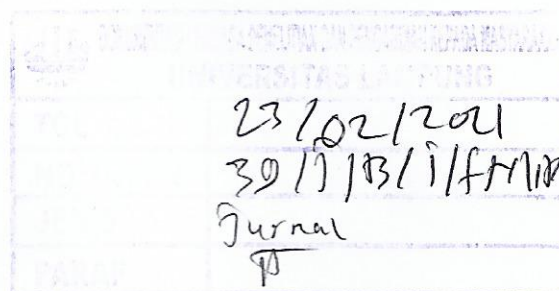
Bandar Lampung, 5 Februari 2021
Penulis

Dr. Eng. Admi Syarif
NIP. 196701031992031003

Mengetahui
Ketua LPPM Universitas Lampung



Dr. Lusmeilia Afriani, D.E.A
NIP. 196505101993032008



2

DETAIL JURNAL

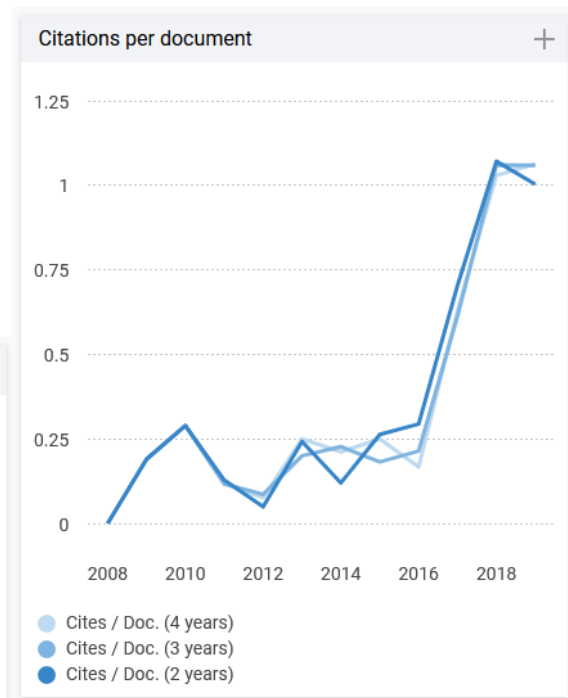
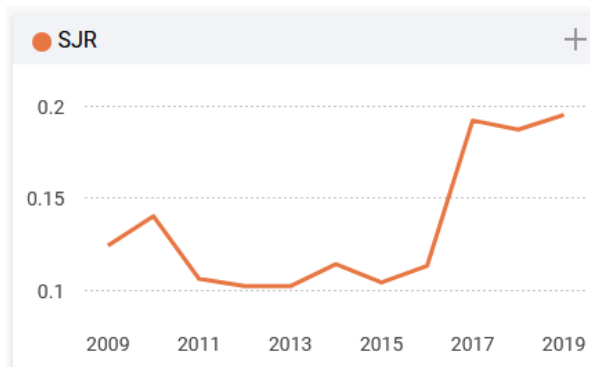
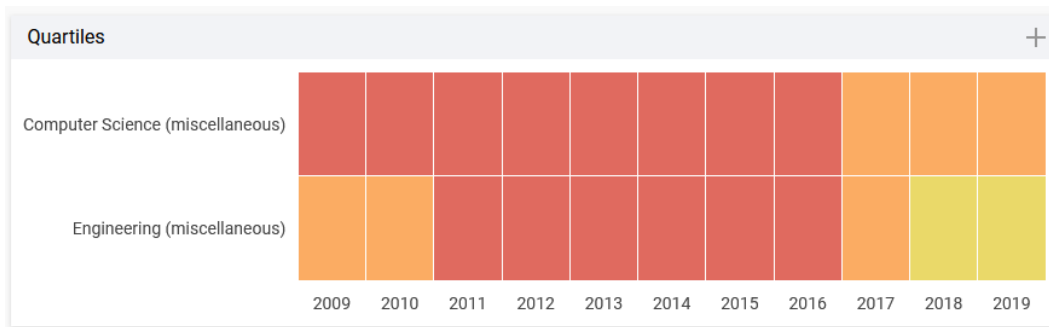
1. Detail Jurnal

Nama Jurnal	: International Journal of Intelligent Engineering and Systems
Publisher	: Intelligent Networks and Systems Society
ISSN	: 21853118 (Online), 2185310X (Prints)
Index Status	: Q2
URL	: http://www.inass.org/publications.html
Publication Frequency	: Quarterly

2. SJR Detail



Country	Japan - SJR Ranking of Japan	11 H Index
Subject Area and Category	Computer Science Computer Science (miscellaneous) Engineering Engineering (miscellaneous)	
Publisher	Intelligent Networks and Systems Society	
Publication type	Journals	
ISSN	21853118, 2185310X	
Coverage	2008-2020	
Scope	International Journal of Intelligent Engineering and Systems is an OPEN ACCESS international journal which gains a foothold in Asia and opens to the world. It aims to promote the integration of intelligent engineering and systems. The focus is to publish papers on state-of-the-art intelligent computing, network engineering, electrical/electronics engineering, and industrial engineering and systems, with emphasis on novel technologies, theoretical work and engineering applications. The audience includes researchers, managers and operators for intelligent engineering and systems as well as designers and developers.	



3. Scopus

The International Journal of Intelligent Engineering and Systems has been officially approved for inclusion in SCOPUS, which is the largest abstract and citation database of research literature and quality web sources covering nearly 18,000 titles from more than 5,000 publishers. Scopus[®] is a registered trademark of Elsevier B.V.

Scopus has become the source for data to establish the annual The Times Higher Education Supplement (THES) World University Rankings because of its transparency, high quality, and broad coverage with all types of university journals.



Source details

International Journal of Intelligent Engineering and Systems

Scopus coverage years: from 2008 to Present

Publisher: Intelligent Networks and Systems Society

ISSN: 2185-210X E-ISSN: 2185-5119

Subject areas: [Engineering \(General Engineering\)](#) [Computer Science \(General Computer Science\)](#)

Source type: Journal

[View all documents >](#)[Set document alert](#)[Source journal link](#) [Source Homepage](#)[CiteScore](#)[CiteScore rank & trend](#)[Scopus content coverage](#)

Improved CiteScore methodology

CiteScore 2019 counts the citations received in 2016-2019 to articles, reviews, conference papers, book chapters and data papers published in 2016-2019, and divides this by the number of publications published in 2016-2019. [Learn more >](#)

CiteScore: 2019



$$1.5 = \frac{325 \text{ Citations 2016 - 2019}}{271 \text{ Publications 2016 - 2019}}$$

Consistent by 2019

CiteScore Tracker 2020



$$1.8 = \frac{1,433 \text{ Citations to date}}{794 \text{ Publications to date}}$$

Last updated on 03 March, 2021 - Updated weekly

CiteScore rank 2019

Category	Rank	Percentile
Engineering		
General Engineering	4111/5299	<div><div></div></div> 54th
Computer Science		
General Computer Science	4121/5221	<div><div></div></div> 40th

[View CiteScore methodology >](#) [CiteScore FAQ >](#) [Add CiteScore to your site >](#)

4. Topics

The topics covered by the journal include, but are not limited to the following:

- Soft computing
- Hard computing
- Vision computing
- Hybrid computing
- Grid computing

- Intelligent information engineering
- Information Education System
- Communication network
- Sensor network
- Computer network
- Network simulation
- Security engineering for networks and computers
- Multi-core technology and parallel processing systems
- Bio-inspired systems
- Circuits design and systems
- Signal processing systems
- Industrial engineering systems
- Power System control
- Power distribution
- Robot engineering and applications
- Intelligent CAD/CAE/CAM systems
- Advanced design and fabrication systems
- Emerging design theory and technologies in manufacturing systems

5. Editorial Staff

5.1. Advisory Board

- Hirofumi Sasaki, Professor Emeritus of Tokai University, Japan
- Hassan Bevrani, University of Kurdistan, Iran

5.2. Editor-in-Chief

- Kei Eguchi, Fukuoka Institute of Technology, Japan

5.3. Associate Editors-in-Chief

- Amphawan Julsereewong, King Mongkut's Institute of Technology Ladkrabang, Thailand
- Sawai Pongswatd, King Mongkut's Institute of Technology Ladkrabang, Thailand

5.4. Editorial Board

- Nam-Soo Kim, Cheongju University, Republic of Korea
- B. T. P. Madhav, K L University, India
- Farzin Asadi, Kocaeli University, Turkey
- Win Pa Pa, University Computer Studies Yangon, Myanmar

- Prasit Julsereewong, King Mongkut's Institute of Technology Ladkrabang, Thailand
- Worapong Tangsirat, King Mongkut's Institute of Technology Ladkrabang, Thailand
- Prapart Ukakimaparn, King Mongkut's Institute of Technology Ladkrabang, Thailand
- Sawatsakorn Chaiyasoonthorn, Ramkhamhaeng University, Thailand
- Krit Smerpitak, King Mongkut's Institute of Technology Ladkrabang, Thailand
- Apinai Rerkratn, King Mongkut's Institute of Technology Ladkrabang, Thailand
- Pittaya Pannil, King Mongkut's Institute of Technology Ladkrabang, Thailand
- Anucha Kaewpoonsuk, Naresuan University, Thailand
- Teerawat Thepmanee, King Mongkut's Institute of Technology Ladkrabang, Thailand
- Chirdpong Deelertpaiboon, King Mongkut's Institute of Technology Ladkrabang, Thailand
- Natee Thong-un, King Mongkut's Institute of Technology Ladkrabang, Thailand
- Sart Kummool, King Mongkut's Institute of Technology Ladkrabang, Thailand
- Ichirou Oota, National Institute of Technology Kumamoto College, Japan
- Takaaki Ishibashi, National Institute of Technology Kumamoto College, Japan
- Kou Yamada, Gunma University, Japan
- Kuniaki Fujimoto, Tokai University, Japan
- Mitsutoshi Yahara, Tokai University Fukuoka Junior College, Japan
- Shuji Kurebayashi, Shizuoka University, Japan
- Wanglok Do, Fukuoka Institute of Technology, Japan
- Riyanarto Sarno, Institut Teknologi Sepuluh Nopember, Indonesia

5.5. Board member responsibilities

- Editorial board members must obey the direction provided by the Editors-in-Chief of the journal.
- Editorial board members must judge every submission equally and transparently.
- Editorial board members should not reveal any information regarding the author, reviewer or the paper information to anyone.
- Editorial board members are responsible for timely publishing the accepted papers.
- Editorial board members may attend the Editorial Board meetings occasionally scheduled by the Editorial Office for taking part in the discussion for the improvement of the Journal.
- Editors should not have any conflict of interest with any assignment.
- Editors should be responsible for a fast and transparent peer review process.
- Editors should keep the policy of fast and effective peer review.
- Editors should be responsible for any query regarding reconsideration of editorial decisions and should provide the decision quickly and clearly with proper reasons.
- Editors should be responsible for selecting potential reviewers considering their expertise in particular subject areas and keeps monitoring the review process.

- Editors will be responsible to convey the expectations of the journal to the reviewers with the review scope, quality and timeliness for an effective, fair and constructive review for the assigned submission.
- Editors should keep in mind the time required for reviewing papers before sending any reminder to the reviewer so that assigned reviewer should get appropriate time he or she requires.
- Editors-in-Chief can take the final decision for any publication oriented issue.

3

**ARTIKEL FINAL
YANG SUDAH
DIPUBLIKASI**



Performance Evaluation of Various Heuristic Algorithms To Solve Job Shop Scheduling Problem (JSSP)

Admi Syarif^{1*}Ade Pamungkas¹Renaldi Kumar¹Mitsuo Gen²¹Department of Computer Science, Faculty of Mathematics and Sciences, Lampung University, Indonesia²Department of Computer Science, Tokyo University of Science, Tokyo, Japan* Corresponding author's Email: admi.syarif@fmipa.unila.ac.id

Abstract: Scheduling is a famous optimization problem that seeks the best strategy of allocating resources over time to perform jobs/tasks satisfying specific criteria. It exists everywhere in everyday life, particularly in manufacturing or industrial applications. An essential class of scheduling problems is a job shop scheduling problem (JSSP), an NP-hard optimization problem. Several researchers have reported the use of heuristic methods to solve JSSP. This paper aims to investigate the performance of various heuristic algorithms to solve JSSP. Firstly, we developed a Genetic Algorithm (GA) and compared the performance of some heuristic algorithms, including Particle Swarm Optimization (PSO), Upper-level algorithm (UPLA), Differential-based Harmony Search (DHS), Grey Wolf Optimization (GWO), Ant Colony Optimization (ACO), Bacterial Foraging Optimization (BFO), Parallel Bat Optimization (PBA), and Tabu Search (TS). The experimental results of the 28 benchmark test problems validated that the algorithms, except ACO, can provide the optimal solution of JSSP. PBA delivers the most impressive performance that solves 26 cases optimally, with the average error equal to 0.05%. Among those 28 test problems, TS, DHS, and PBA can solve 26 instances optimally, followed by GA that solves 21 cases.

Keywords: Combinatorial optimization, Job shop scheduling problem (JSSP), Artificial intelligence, Heuristic algorithms, NP-hard problem.

1. Introduction

Scheduling is one of the most essential and commonly encountered classes of optimization problems. Scheduling problems exist everywhere in everyday life, particularly in industrial or manufacturing applications. What makes scheduling problems important is that many manufacturing companies or industries have limited resources and have to satisfy specific criteria. Determining an excellent strategy to schedule tasks will reduce production costs or improve profitability.

There are many variations of scheduling problems for several real-world uses. There are, in general, two classes of fundamental scheduling problems discussed in many pieces of literature. Those are the Job Shop Scheduling Problem (JSSP) and the Flow Shop Scheduling Problem (FSSP). Among them, JSSP has been the most widespread

and complex problem. The JSSP model has been vital and practical and challenges many researchers in engineering, computing, and operational [1]. It represents a problem of allocating a set of resources (machine) to perform tasks (job) that consists of m different operations, and the separate device has the other processing time. The main objective is to determine the best machine schedule to do all job with the best objective value, i.e., minimizing makespan (C_{\max}), mean flow time, mean tardiness, earliness, maximum lateness, etc. [2]. The JSSP with n job and m machine will have $(n!)^m$ possible solutions. Thus, for the relatively large size problem, it will be computationally expensive to solve scheduling problems optimally [3].

Generally, there are two classes of methods for solving JSSP; exact and heuristic methods [4]. The first-class methods include: integer programming [5] [6], Lagrangian relaxation [7], dynamic

programming [8], and Branch and bound [9]. Another class is called heuristic algorithms, first introduced in early 1960. It was initially concerned with increasing the effectiveness of the problem-solving process. Although those methods do not guarantee the finding of an optimal solution, those have been reported useful in solving many challenging optimization problems within a reasonable computational time.

The term heuristic is usually related to the methods or algorithms for solving problems intelligently. Over the last few decades, with the rapid increase of computer technology, we have seen enormous growth in heuristic approaches to various hard and challenging optimization problems, such as Simulated Annealing (SA), Ant Colony Optimization (ACO), Genetic Algorithm (GA), Branch-and-Bound, Tabu Search (TS), and so on.

Among the heuristics algorithms, GA probably has been the most popular approach [10]. Aided by GAs, researchers evolve solutions to complex combinatorial optimization problems easily and rapidly. Our past researches reported the excellent performance of GA in solving various combinatorial optimization problems [11, 12], and [13]. In contrast to other heuristics methods, it utilizes a set population of solutions in its search. It makes GA more robust to solve many real-world problems [10].

For solving JSSP, several researchers have reported the robustness of heuristics methods. These include the TS algorithm by Mauro [14], Ant Colony Optimization (ACO) by Flórez [15], Bacterial Foraging Optimization (BFO) by Zhao [16], Bat Optimization algorithm by Dao [17], and so on. Despite these interests, however, no researcher said the best method to solve JSSP all-time optimally. This fact shows that researches on the performance evaluation of the heuristic algorithms for JSSP are very crucial.

This paper aims to investigate the performance of heuristic algorithms for solving JSSP. First, we developed a GA approach and conducted intensive numerical experiments on a set of Benchmark test problems (3 test problems of Fisher dan Thompson [18], and 25 instances of Lawrence [19]). Further; we compared the results to those of some heuristic algorithms, including Particle Swarm Optimization (PSO), Upper-level algorithm (UPLA), Differential-based Harmony Search (DHS), Grey Wolf Optimization (GWO), Ant Colony Optimization (ACO), Bacterial Foraging Optimization (BFO), Parallel Bat Optimization (PBA), and Tabu Search (TS). The comparison is made based on the solution's quality, the relative error, and the number of instances solved (NIS) optimally for each test problem

The organization for the remainder of the paper is as follows: the next section describes the formulation

of JSSP. In the third section, we concern ourselves with some essential discussions of several heuristic algorithms, including GA's working mechanisms. Furthermore, in the fourth section, some comparisons of results from the numerical experiments on Benchmark test problems are presented. We evaluate the algorithms' performance based on the solution quality, the relative error, and the number of instances solved (NIS) optimally. In the end section, we provide the conclusion of this study, showing the approaches; remarkable effectiveness.

2. Mathematical model of JSSP

Consider the JSSP with m machines to perform n jobs or tasks. Each job/task consists of m operations. The order of operations for the machines is predetermined. The different device is used for a separate action to complete one job. The problem involves designing an effective strategy (called schedule) of assigning some activities to be done by the devices by meeting constraints.

The main objective of JSSP is to determine the best machine schedule to do all jobs with the best objective function, i.e., minimizing makespan(C_{max}), mean flow time, mean tardiness, earliness, and maximum lateness. The most common constraint of the JSSP is as follows [20]:

1. A machine can process only a job or task at a time.
2. The machine sequence of the machine to process each job must be the same.
3. The process of a job cannot be interrupted.

Let t_{ij} and f_{ij} are the starting and the finishing time of processing job j at machine i . P_{ij} is the processing time of machine i to perform job j . The makespan (C_{max}) here represents the finishing time of the last job. The mathematical model of JSSP is as follows [21]:

$$\min C_{max} \quad (1)$$

s.t.

$$t_{hj} - t_{ij} \geq P_{ij} \quad (2)$$

$$C_{max} - t_{ij} \geq P_{ij} \quad (3)$$

$$t_{ij} - t_{ik} \geq P_{ik} \quad \text{or} \quad t_{ik} - t_{ij} \geq P_{ij} \quad (4)$$

$$t_{ij} \geq 0 \quad (5)$$

In this model, Eq. (1) is the objective function to minimize the makespan. The constraint (2)

guarantees that the next step of machine h for job j is started after finishing the step at machine i for job j . Next, the constraint (3) ensures the makespan is equal to or greater than the finishing time of the last job. Eq. (4) shows that only a machine processed a job at a time. Finally, Eq. (5) is a non-negative constraint.

3. Heuristic approaches for JSSP

In this section, we shall describe clearly the drawbacks of previous heuristic approaches used for solving JSSP, including Particle Swarm Optimization (PSO), Upper-level algorithm (UPLA), Differential-based Harmony Search (DHS), Grey Wolf Optimization (GWO), Ant Colony Optimization (ACO), Bacterial Foraging Optimization (BFO), Parallel Bat Optimization (PBA), and Tabu Search (TS). Next, we introduced the design of the GA approach. We emphasize the difference between the methods to clarify the position of this works.

3.1 ACO (Ant colony optimization)

Ant Colony Optimization (ACO) is a heuristic algorithm that combines concepts from Artificial Intelligence and Biology, inspired by ants' collective behaviour [15]. Dorigo first introduced ACO for solving the Traveling Salesman Problem. Currently, ACO has solved various fields of our daily life applications. The ACO-based method, called Elitist Ant System (EAS) for JSSP, has been carried out among many by Florez in 2013. Each job consists of a sequence of operations, and each process comes with a determined machine and processing time. They adopt the collective intelligence of many simple agents to determine optimal solutions with minimum makespan.

They presented the obtained results for each of the JSSP instances by Lawrence [22]. They compared the results with those of Tabu Search (TS) and the best-known solution (BKS) taken from [23]. The algorithm implemented, Elitist Ant System, has proven to be competitive by finding the more reliable quality solutions for JSSP [15]. However, it also requires more effort to obtain the best-known solution for all LA instances.

3.2 Particle swarm optimization (PSO)

Particle Swarm Optimization (PSO) is a population-based metaheuristic optimization approach, introduced by Eberhart and Kennedy. Animals' behaviour to search for food, such as birds and fishes, inspires the PSO. Each flock of birds or fishes tends to determine its speed based on personal experience and information obtained through

interactions with other members. Pongchairerks and Kachitvichyanukul reported the use of PSO to solve JSSP (JSP-PSO) in 2009 [24]. This paper proposed the GLN-PSO algorithm that allows the swarm to explore the other parts of the search spaces simultaneously. To evaluate the algorithm's performance, they had numerical experiments on 33 well-known benchmark test problems from Fisher and Thompson (FT06, FT10, FT20), and the rest from Lawrence. Their computational results show the algorithm can optimally solve the problem 17 times.

3.3 Tabu search (TS)

Another popular heuristic method for solving combinatorial optimization problems is Tabu Search. Since Glover originally introduced it in 1986, hundreds of researchers reported the success of Tabu Search (TS) applications to various combinatorial optimization problems. It has been reported among practical algorithms and provides optimal/near-optimal solutions for many cases. TS searches for the best solution based on the local search method's optimization. A TS algorithm's main components are memory structures, a trace of the search's evolution, and strategies to use the memorized information in the best possible way. Dell'Amico first introduced the use of TS for solving the JSSP [14]. Their basic idea is to avoid cycles in the search's evolution by inhibiting the algorithm from reoccurring more recently made moves. They evaluated TS's performance on a set of problem instances, including Lawrence (LA01-40) [22]. Their results show that TS is useful in finding the optimal/near-optimal solutions.

3.4 Upper-level algorithm (UPLA)

Nowadays, research on developing the heuristic algorithm for JSSP has become more variegated. In 2019, Pongchairerks proposed a brand new two-level metaheuristic algorithm, consisting of an upper-level algorithm (UPLA) and a lower-level algorithm (LOLA) for the JSSP. The UPLA is a brand new algorithm that begins with a population of the combinations of values from LOLA's input-parameter. At every iteration, UPLA attempts to increase its population by utilizing the feedback returned from LOLA. Thus, LOLA may improve from a local search algorithm to be an iterated local search algorithm.

Furthermore, UPLA and LOLA result in the two-level algorithm, which may adapt to every JSSP instance. Similar to the other population-based algorithms, UPLA examines search space based on the population of the combination from the input

parameter. Real numbers represent all input-parameter values.

Among JSSP algorithms, the most similar algorithm to the proposed algorithm was the two-level Particle Swarm Optimization (PSO) [24]. The correspondence is that they generate parameterized-active schedules with similar methods on both their lower-level algorithms; furthermore, parameters for both algorithms (lower and upper-level algorithms) control the identical. However, the two-level PSO is different from the suggested two-level metaheuristic algorithm that uses GLN-PSO's framework [25]. The authors assessed the algorithms' performance on 53 well-known benchmark instances, including FT06, FT10, FT20, and LA01-LA40 [26]. Considering the similarity and difference, they also compared their results with those of the two-level PSO [24].

3.5 Differential-based harmony search (DHS)

The Differential-based Harmony Search (DHS) to minimize makespan for JSSP was reported by Zhao in 2018 [16]. The DHS improves the variable neighbourhood search (VNS) based on the critical path blocks. The transformed VNS, on the critical path, is embedded into the DHS to seek a more reliable solution based on the blocks. They evaluate DHS's performances on a set of benchmark instances from the OR-library with the objective of minimized makespan [27]. Compared with various HS-based algorithms and other state-of-the-art algorithms, the DHS is superior in solution quality, convergence speed, and stability [16].

3.6 GWO (Grey wolf optimization)

The Grey wolf optimization (GWO) algorithm is a new population-oriented heuristic algorithm inspired by grey wolves' social hierarchy and hunting behaviour. Tianhua Jiang introduced GWO, a brand new swarm-based intelligence algorithm, to deal with optimization problems in 2018. The algorithm is based on the crossover operation and adapt the searching operator to minimize the makespan (maximum completion time). They also introduced an adaptive modification method to the algorithm to keep the variety of population. They compared the results with other published algorithms in the two scheduling cases. According to the experimental results, GWO provided better solutions for some instances [28].

3.7 Bacterial foraging optimization (BFO)

Kim Passino proposed another intelligent heuristic algorithm, called the Bacterial Foraging

Optimization (BFO), in 2002 [29]. The BFO is inspired by the cooperation and competition behaviours of a bacterium named *E. coli* in seeking food. One of the primary processes in BFO is the evolution process. It begins when the bacterium migrates to a better solution according to the advantaged group's activity. Zhou introduced an algorithm called Chemotaxis-enhanced-BFO (CEBFO) to solve the JSSP [30]. To improve the algorithm's performance, they include a local search operation and chemotaxis with the differential evolution (DE). They conducted some numerical experiments on a set of benchmark problems of JSSP. The results demonstrated a good understanding of the algorithm.

3.8 Parallel bat algorithm (PBA)

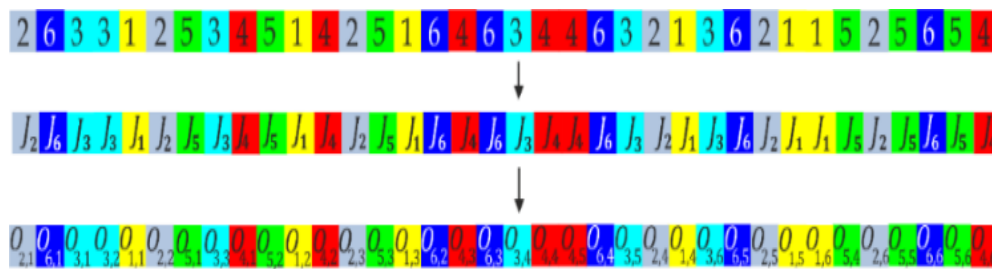
In 2015, Dao proposed a parallel-based heuristic algorithm version, called the parallel bat algorithm (PBA) [17]. The fundamental structure of the PBA is to divide the distribution of the bat populations into several groups. They offered three schemes, namely a random-key encoding, a makespan, and a communication strategy. To examine the method's accuracy, they had some experiments on 43 (forty-three) benchmark instances (Fisher and Thompson with FT06, FT10, FT20 [18], and Lawrence (LA01–LA40) [31]). They compared their experimental results are to those of the PSO algorithm. These show that the intended approach gives competitive returns.

3.9 Genetic algorithm (GA)

Since Holland introduced GA in 1975, it has witnessed many exciting advances in using Genetic Algorithms (GAs) to solve challenging optimization problems in everything from production design to inventory and network design problems. It is a heuristics method, inspired by the process of Darwinian evolution. GA has been a multi-purpose approach for searching the global optimality; adapting GAs to a specific optimization problem is challenging but frustrating. The selection methods, efficient design of the chromosome representation, crossover and mutation process, and GA parameters' value influence GA's success [32]. Therefore, discovering an efficient GA approach system for a particular problem becomes essential in GA research.

3.9.1. The Chromosome representation

When implementing GA for an optimization problem, an important issue is how to generate a chromosome that would bring us to the right solution. For the initial population, we have to create a



pop_size chromosome. Each chromosome consists of $n \times m$ gen generated randomly, and each *job* will appear m in the chromosome. One illustrates an example of the chromosome for the test problem FT06, having six jobs and six machines.

The chromosome in Figure. 1 indicates that the first activity to be scheduled is job two at machine one, followed by job six at machine one; then, job three at machine one, and so on, according to the order arranged in the chromosome list. $O_{i,j}$ represents the operation for the job i at machine j .

3.9.2. Genetic operations

Procedure: Self Crossover:

Step 1: Chose a parent arbitrarily for crossover.

Step 2: Determine two crossover points randomly

Step 3: Move substring between the above two points

The mutation operation is an essential feature of GA to maintain the chromosome's diversity in the generation. This paper adopts the swap mutation that randomly selects two genes in the chromosome and then exchanges them.

3.9.3. Evaluation and selection

In GA implementation, we computed the fitness value to measure how well each chromosome fits the problem's requirements. For JSSP, we can use makespan as the fitness value as follows:

$$\text{Fitness}(x) = \frac{1}{f(x)} \quad (6)$$

where $f(x)$ is the objective function (makespan).

The decoding process to compute the makespan (C_{\max}) of the schedule is as follows:

Step 1: Select the chromosome for the decoding process.

Step 2: Read gen in the chromosome started from the left.

Step 3: Determine the machine number from the machine-order matrix, based on the job operation number's information.

Step 4: Determine the processing time from the processing time matrix based on the job operation number's data.

Step 5: Determine the maximum time of the last job time.

Step 6: Renew the current job finishing time by adding the time to the result of Step 5;

Step 7: Repeat Step 2 to Step 6 until the last-gen in the chromosome.

Another essential process of GA is the way to determine the chromosome for the next population. Of course, the selection process should be done based on the fitness value. There have been several selection strategies introduced in GA applications. Here, we adopt the elitist approach by selecting the best *pop_size* chromosome for the next generation.

4. Numerical experiments and results

4.1 Design of numerical experiments

To evaluate the effectiveness and the efficiency of the algorithm, we first have some numerical experiments for GA on 28 Benchmark test problems: 3 instances (FT06, FT10, dan FT20) of Fisher dan Thompson [18], and 25 instances (LA01-LA25) of Lawrence [19], taken from the OR-library [27]. We implement the algorithm in MATLAB R2015b and run on an Intel Core i5 processor of 2.53 GHz.

The GA parameters are set as: crossover probability ($\mathbf{p_C}$) = 0.4, mutation probability ($\mathbf{p_M}$) = 0.2, population size ($\mathbf{pop_size}$) = 400 and maximum generation ($\mathbf{max_gen}$) = 10-2000, for each test problem, the experiments are conducted 10 (ten) times. Table 1 presents the overall results obtained for all test problems, where the best and the average values represent the best and the average fitness value from the 10 (ten) running times. BKS represents the best-known solution in the literature.

Table 1. The experimental results of the GA approach

Cases	$n \times m$	ACT*	BKS*	GA			Error (%)
				Best	Average	Worst	
FT06	6 × 6	1,00	55	55	55	55	0,00
FT10	10 × 10	2,76	930	951	988.8	1030	2,26
FT20	20 × 5	1,08	1165	1178	1184.3	1197	1,12
LA01	10 × 5	6,00	666	666	666	666	0,00
LA02	10 × 5	3,97	655	655	658.8	666	0,00
LA03	10 × 5	6,44	597	597	611	621	0,00
LA04	10 × 5	3,35	590	590	592	601	0,00
LA05	10 × 5	1,8	593	593	593	593	0,00
LA06	15 × 5	2,00	926	926	926	926	0,00
LA07	15 × 5	9,00	890	890	890	890	0,00
LA08	15 × 5	9,00	863	863	863	863	0,00
LA09	15 × 5	1,7,0	951	951	951	951	0,00
LA10	15 × 5	0,80	958	958	958	958	0,00
LA11	20 × 5	2,80	1222	1222	1222	1222	0,00
LA12	20 × 5	2,80	1039	1039	1039	1039	0,00
LA13	20 × 5	2,60	1150	1150	1150	1150	0,00
LA14	20 × 5	1,00	1292	1292	1292	1292	0,00
LA15	20 × 5	20,00	1207	1207	1207	1207	0,00
LA16	10 × 10	767	945	959	977.2	997	0,00
LA17	10 × 10	774	784	784	788.9	797	0,00
LA18	10 × 10	808	848	848	868.5	909	0,00
LA19	10 × 10	1.395	842	842	850	874	0,00
LA20	10 × 10	1.234	902	907	928.4	992	0,55
LA21	15 × 10	1.743	1046	1061	1097	1114	1,43
LA22	15 × 10	1.443	927	943	987.8	1046	1,08
LA23	15 × 10	752	1032	1032	1035.3	1054	0,00
LA24	15 × 10	1.122	935	948	977	994	1,39
LA25	15 × 10	2.049	977	987	1015,8	1042	1,02
Average							0.3162

*ACT: Average Computational Time (in second)

4.2 Results and discussion

In the above table, the error is computed by using the following formula:

$$Error = \frac{(Best-Optimum) \times 100\%}{Optimum} \quad (7)$$

Here, one can notice the excellent performance of GA to solve JSSP. Despite not reaching the optimal solution all-time, GA presents the optimal solutions (21 instances), with an average error of less than 0.32 percent. The results also show that GA can provide solutions to the problems within reasonable computational time. For some hard/difficult cases, GA can obtain near-optimal solutions with an error from 0.5 to 1.43 percent. More efforts can be made to improve the solutions by possibly hybridizing GA with other local search techniques. The Gantt chart schedule and the convergence of the solution for LA40 are illustrated in Figure. 2 and Figure. 3, respectively.

4.3 Comparison of some heuristic methods

In this research, we evaluate the merit and the limitation of the approaches by comparing the results of some heuristic algorithms, including Ant Colony Optimization (ACO) [15], Particle Swarm Optimization (PSO) [33], Tabu Search (TS) [14], Upper-level algorithm (UPLA) [26], Differential-based Harmony Search (DHS) [16], Grey Wolf Optimization (GWO) [28], Bacterial Foraging Optimization (BFO) [30], Parallel Bat Optimization (PBA) [17], and the proposed Genetic Algorithm (GA). The performances are measured based on the solution quality, the number of instances solved (NIS) optimally, and the relative error. We made a

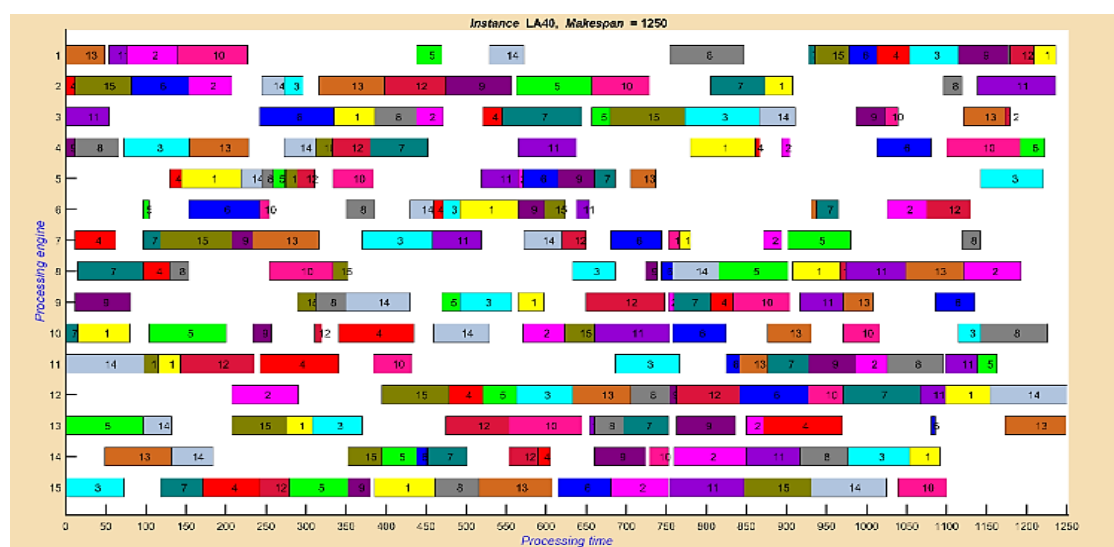


Figure. 2 Gantt chart schedule for LA40

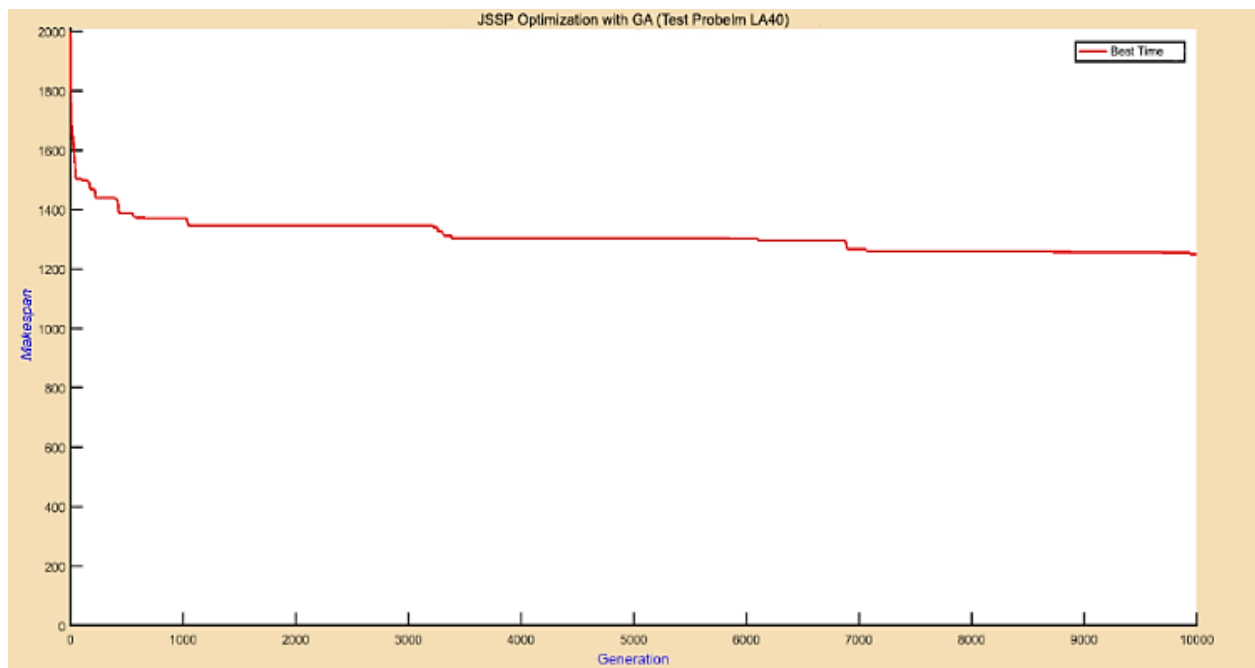


Figure. 3 The convergence of the objective function in the generation for LA40

Table 2. Performance of the heuristic approaches on all instances

Test Problem	Size ($n \times m$)	BKS*	ACO [15]	PSO [33]	TS [14]	UPLA [26]	DHS [16]	GWO [28]	BFO [30]	PBA [17]	GA (Proposed)
FT06	6×6	55	-	55	-	55	55	55	55	55	55
FT10	10×10	930	-	951	-	930	930	940	937	930	951
FT20	20×5	1165	-	1191	-	1165	1165	1178	1171	1165	1178
LA01	10×5	666	666	666	666	666	666	666	666	666	666
LA02	10×5	655	669	663	655	655	655	655	655	655	655
LA03	10×5	597	623	603	597	597	597	597	597	597	597
LA04	10×5	590	611	611	590	590	590	590	590	590	590
LA05	10×5	593	593	593	593	593	593	593	593	593	593
LA06	15×5	926	926	926	926	926	926	926	926	926	926
LA07	15×5	890	890	890	890	890	890	890	890	890	890
LA08	15×5	863	863	863	863	863	863	863	863	863	863
LA09	15×5	951	951	951	951	951	951	951	951	951	951
LA10	15×5	958	958	958	958	958	958	958	958	958	958
LA11	20×5	1222	1222	1222	1222	1222	1222	1222	1222	1222	1222
LA12	20×5	1039	1039	1039	1039	1039	1039	1039	1039	1039	1039
LA13	20×5	1150	1150	1150	1150	1150	1150	1150	1150	1150	1150
LA14	20×5	1292	1292	1292	1292	1292	1292	1292	1292	1292	1292
LA15	20×5	1207	1212	1207	1207	1207	1207	1207	1207	1207	1207
LA16	10×10	945	1005	959	945	945	945	956	945	945	945
LA17	10×10	784	812	784	784	784	784	790	785	784	784
LA18	10×10	848	885	848	848	848	848	859	848	848	848
LA19	10×10	842	875	857	842	842	842	845	844	842	842
LA20	10×10	902	912	910	902	902	902	937	907	902	907
LA21	15×10	1046	1107	1074	1048	1052	1046	1090	-	1046	1061
LA22	15×10	927	1018	944	933	927	927	970	-	933	937
LA23	15×10	1032	1051	1032	1032	1032	1032	1032	-	1032	1032
LA24	15×10	935	1011	971	941	941	979	982	-	941	948
LA25	15×10	977	1062	987	979	982	1016	1008	-	977	987

Table 3. The error comparison of the heuristic approaches in all instances

Test Problem	Dimensi ($n \times m$)	ACO	PSO	TS	UPLA	DHS	GWO	BFO	PBA	GA
FT06	6×6	-	0.00	-	0.00	0.00	0.00	0.00	0.00	0.00
FT10	10×10	-	2.26	-	0.00	0.00	1.08	0.75	0.00	2.26
FT20	20×5	-	2.23	-	0.00	0.00	1.12	0.52	0.00	1.12
LA01	10×5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA02	10×5	2.14	1.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA03	10×5	4.36	1.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA04	10×5	3.56	3.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA05	10×5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA06	15×5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA07	15×5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA08	15×5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA09	15×5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA10	15×5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA11	20×5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA12	20×5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA13	20×5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA14	20×5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA15	20×5	0.41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA16	10×10	6.35	1.48	0.00	0.00	0.00	1.16	0.00	0.00	0.00
LA17	10×10	3.57	0.00	0.00	0.00	0.00	0.77	0.13	0.00	0.00
LA18	10×10	4.36	0.00	0.00	0.00	0.00	1.30	0.00	0.00	0.00
LA19	10×10	3.92	1.78	0.00	0.00	0.00	0.36	0.24	0.00	0.00
LA20	10×10	1.11	0.89	0.00	0.00	0.00	3.88	0.55	0.00	0.55
LA21	15×10	5.83	2.68	0.19	0.57	0.00	4.21	-	0.00	1.43
LA22	15×10	9.82	1.83	0.65	0.00	0.00	4.64	-	0.65	1.08
LA23	15×10	1.84	0.00	0.00	0.00	0.00	0.00	-	0.00	0.00
LA24	15×10	8.13	3.85	0.64	0.64	4.71	5.03	-	0.64	1.39
LA25	15×10	8.70	1.02	0.20	0.51	3.99	3.17	-	0.00	1.02

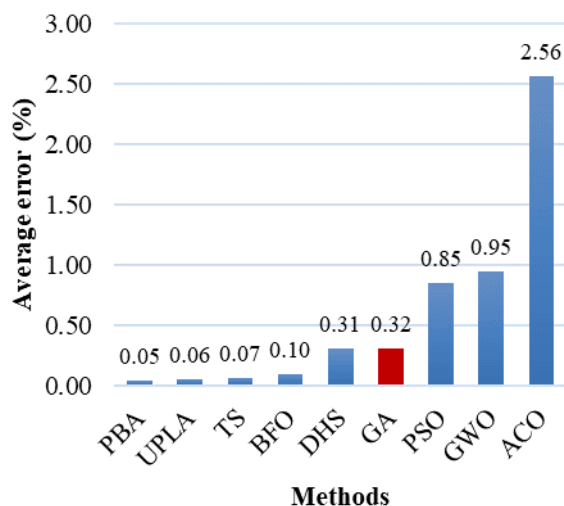


Figure. 4 The comparison of average error

comparison of the results for 28 benchmark test problems (FT06, FT10, F20) and 25 instances (LA01-LA25) of Lawrence [19]. Table 2 summarizes the comparative results. We also computed the percentage relative error concerning BKS, as shown in the following Table 3.

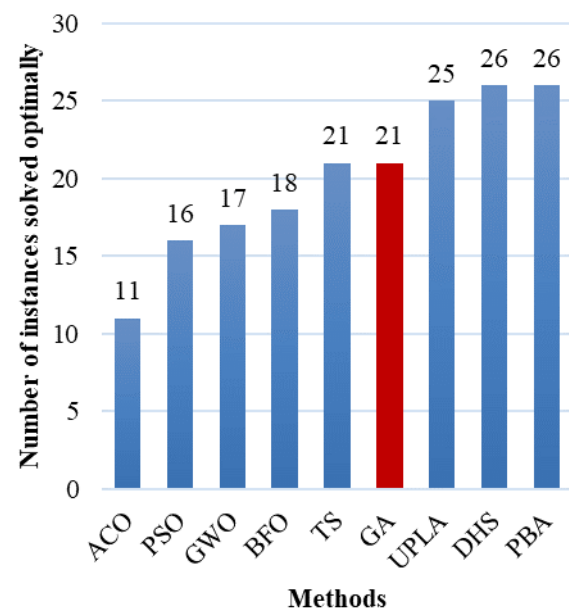


Figure. 5 The comparison of NIS optimally by the heuristic methods

We also analyze the algorithm based on the average errors and the number of instances solved

(NIS) optimally. We illustrated the comparison of the average errors and obtained NIS by the algorithms in Figure. 4 and Figure. 5, respectively. These results indicate that, though no algorithm can give the optimal solution, the algorithms effectively find the optimal/near-optimal solutions to the problems. Almost all algorithms, except ACO, can provide an error of less than 1 percent. PBA presents the most impressive performance that can solve 26 cases optimally, with the average error equal to 0.05%. Among those 28 test problems, TS, DHS, and PBA can solve 26 instances optimally, followed by GA that solves 21 cases.

5. Conclusion

This paper analyzed the performance of some heuristic algorithms for solving JSSP. First, we developed the GA approach and conducted some intensive numerical experiments on a set of Benchmark instances from the literature. We investigated some heuristic methods' performance, based on the solution quality, the relative error, and the number of instances solved (NIS) optimally. The results validate that, though no method presents optimal solutions at all times, the heuristics are robust in searching for the optimal solutions of JSSP. Among them, the PBA is the most effective algorithm that solves 26 instances optimally with an average error of 0.05%, followed by DHS, UPLA, and GA. The computational results show that the proposed GA can obtain competitive results in both NIS (21 cases with an average error of 0.32%) and computational time. These findings add to a growing body of literature on the applications of heuristics.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Admi Syarif supervised the study, analyzed the results, verified the study's findings, and wrote the paper. Mitsuo Gen gave the idea of the algorithm; Renaldi Kumar and Ade Pamungkas designed and performed the experiments;

Acknowledgments

This research was supported by the Scientific "Riset Unggulan" Program, Ministry of Education and Culture, Lampung University, the Grant-in-Aid 2226/UN26.21/PN/2019, Indonesia, 2019. The authors thank the anonymous reviewers for their valuable comments and suggestions on this paper.

References

- [1] J. K. Lenstra and A. H. G. Rinnooy Kan, "Computational Complexity of Discrete Optimization Problems", *Annals of Discrete Mathematics*, Vol. 4, pp. 121–140, 1979.
- [2] J. F. Gonçalves, J. J. De Magalhães Mendes, and M. G. C. Resende, "A hybrid genetic algorithm for the job shop scheduling problem", *European Journal of Operational Research*, Vol. 167, No. 1, pp. 77–95, 2005.
- [3] E. A. C. Uzorh and N. Innocent, "Solving Machine Shops Scheduling Problems using Priority Sequencing Rules Techniques", *The International Journal of Engineering and Science*, Vol. 3, No. 6, pp. 1813–2319, 2014.
- [4] K. Akram, K. Kamal, and A. Zeb, "Fast simulated annealing hybridized with quenching for solving job-shop scheduling problem", *Applied Soft Computing*, Vol. 49, pp. 510–523, 2016.
- [5] C. Özgüven, Y. Yavuz, and L. Özbakir, "Mixed-integer goal programming models for the flexible job-shop scheduling problems with separable and non-separable sequence-dependent setup times", *Applied Mathematical Modelling*, Vol. 36, No. 2, pp. 846–858, 2012.
- [6] D. Catanzaro, L. Gouveia, and M. Labbé, "Improved integer linear programming formulations for the job Sequencing and tool Switching Problem", *European Journal of Operational Research*, Vol. 244, No. 3, pp. 766–777, 2015.
- [7] P. Baptiste, M. Flamini, and F. Sourd, "Lagrangian Bounds for Just-In-Time Job-Shop Scheduling", *Computers & Operations Research*, Vol. 35, No. 3, pp. 906–915, 2008.
- [8] J. A. S. Gromicho, J. J. Van Hoorn, F. Saldanha-da-Gama, and G. T. Timmer, "Solving the job-shop scheduling problem optimally by dynamic programming", *Computers & Operations Research*, Vol. 39, No. 12, pp. 2968–2977, 2012.
- [9] P. Brucker, E. K. Burke, and S. Groenemeyer, "A branch and bound algorithm for the cyclic job-shop problem with transportation", *Computers & Operations Research*, Vol. 39, No. 12, pp. 3200–3214, 2012.
- [10] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons, New York, 2000.
- [11] A. Syarif, Y. S. Yun, and M. Gen, "Study on multi-stage logistic chain network: A spanning tree-based genetic algorithm approach", *Computers and Industrial Engineering*, Vol. 43, No. 1–2, pp. 299–314, 2002.

- [12] M. Gen and A. Syarif, "Double Spanning Tree-Based Genetic Algorithm for Two-Stage Transportation Problem", *International Journal of Knowledge-Based and Intelligent Engineering Systems*, Vol. 7, No. 4, pp. 214–221, 2003.
- [13] A. Syarif, D. Anggraini, K. Muludi, Wamiliana, and M. Gen, "Comparing Various Genetic Algorithm Approaches for Multiple-Choice Multi-Dimensional Knapsack Problem (mm-KP)", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 455–462, 2020.
- [14] M. Dell'Amico and M. Trubian, "Applying tabu search to the job-shop scheduling problem", *Annals of Operations Research*, Vol. 41, No. 3, pp. 231–252, 1993.
- [15] E. Flórez, W. Gómez, L. Bautista, E. Florez, W. Gomez, and Lola Bautista, "An Ant Colony Optimization Algorithm for Job Shop Scheduling Problem", *International Journal of Artificial Intelligence & Applications (IJAIA)*, Vol. 4, No. 4, pp. 53–66, 2013.
- [16] F. Zhao, S. Qin, G. Yang, W. Ma, C. Zhang, and H. Song, "A Differential-Based Harmony Search Algorithm with Variable Neighborhood Search for Job Shop Scheduling Problem and Its Runtime Analysis", *IEEE Access*, Vol. 6, pp. 76313–76330, 2018.
- [17] T. K. Dao, T. S. Pan, T. T. Nguyen, and J. S. Pan, "Parallel bat algorithm for optimizing makespan in job shop scheduling problems", *Journal of Intelligent Manufacturing*, Vol. 29, No. 2, pp. 451–462, 2018.
- [18] H. Fisher and G. L. Thompson, "Probabilistic Learning Combinations of Local Job-Shop Scheduling Rules", *Industrial Scheduling*, Vol. 3, No. 2, pp. 225–251, 1963.
- [19] S. Lawrence, "Supplement to Resource-Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques", *Energy Procedia*, Vol. 4, No. 7, pp. 4411–4417, 1984.
- [20] H. Zaher, N. Ragaa, and H. Sayed, "A novel Improved Bat Algorithm for Job Shop Scheduling Problem", *International Journal of Computer Applications*, Vol. 164, No. 5, pp. 24–30, 2017.
- [21] K. Ploydanai and A. Mungwattana, "Algorithm for Solving Job Shop Scheduling Problem Based on machine availability constraint", *International Journal on Computer Science and Engineering (IJCSE)*, Vol. 02, No. 05, pp. 1919–1925, 2010.
- [22] L. Davis, *Job Shop Scheduling with Genetic Algorithms*, Psychology Press, East Sussex, United Kingdom, 1985.
- [23] D. Applegate and W. Cook, "Computational study of the job-shop scheduling problem", *ORSA Journal on Computing*, Vol. 3, No. 2, pp. 149–156, 1991.
- [24] P. Pongchairerks and V. Kachitvichyanukul, "A two-level Particle Swarm Optimization algorithm on Job-Shop Scheduling Problems", *International Journal of Operational Research*, Vol. 4, No. 4, pp. 390–411, 2009.
- [25] P. Pongchairerks and V. Kachitvichyanukul, "A Non-Homogenous Particle Swarm Optimization with Multiple Social Structures", In: *Proc. of the 2005 International Conf. on Simulation and Modeling*, Nakornpathom, Thailand, pp. 132–136, 2005.
- [26] P. Pongchairerks, "A two-level metaheuristic algorithm for the job-shop scheduling problem", *Complexity*, Vol. 2019, pp. 1–11, 2019.
- [27] <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/jobshop1.txt> (accessed Feb. 07, 2020).
- [28] T. Jiang and C. Zhang, "Application of Grey Wolf Optimization for Solving Combinatorial Problems: Job Shop and Flexible Job Shop Scheduling Cases", *IEEE Access*, Vol. 6, pp. 26231–26240, 2018.
- [29] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control", *IEEE Control Systems Magazine*, Vol. 22, No. 3, pp. 52–67, 2002.
- [30] F. Zhao, X. Jiang, C. Zhang, and J. Wang, "A chemotaxis-enhanced bacterial foraging algorithm and its application in job shop scheduling problem", *International Journal of Computer Integrated Manufacturing*, Vol. 28, No. 10, pp. 1106–1121, 2015.
- [31] J. Dossey, A. Otto, L. Spence, and C. Eynden, *Discrete Mathematics*, 2nd Edition, HarpersCollins College Div, New York City, New York, 1993.
- [32] A. Syarif, W. Wamiliana, P. Lumbanraja, and M. Gen, "Study on genetic algorithm (GA) approaches for solving Flow Shop Scheduling Problem (FSSP)", In: *Proc. of the 5th International Conf. on Science, Technology and Interdisciplinary Research (IC-STAR 2019)*, Bandar Lampung, Indonesia, 2020.
- [33] P. Pongchairerks, "Particle swarm optimization algorithm applied to scheduling problems", *Science Asia*, Vol. 35, No. 1, pp. 89–94, 2009.

4

**COVER LETTER
JURNAL**

EInternational Journal of Intelligent Engineering and Systems (IJIES)

Cover Letter

<http://www.inass.org>

Please complete this form and send it to us with your manuscript.

Title of a corresponding author (Prof./Assoc.Prof./Assis.Prof./Dr./Mr./Ms.) * Choose either	Dr. Eng.
Full Name and Surname of Corresponding Author	Admi Syarif
Paper Title	Performance Evaluation of Various Heuristic Algorithms To Solve Job Shop Scheduling Problem (JSSP)
Co-author(s)	Admi Syarif, Ade Pamungkas, Renaldi Kumar M and Mitsuo Gen
Organisation	Department of Computer Science, Faculty of Mathematics and Sciences, Lampung University, Jl. S. Brodjonegoro No.1, Bandar Lampung, 35145
Country	Indonesia
E-mail	admi.syarif@fmipa.unila.ac.id

Publication Option:

Please **delete** either:

✓ **Special Publication + Format Correction: (USD 250)**

In this option, your manuscript is formatted by IJIES staffs. Accepted papers will be published **about 2 months later**.

(* **Paypal** is the only payment method. The bank transfer is not supported.)

The review process of IJIES consists of two processes: 1st review and 2nd review. Each process requires about 2 weeks to review your manuscript.

Note: Please submit your 1st version with a twin-column format to clarify the page length of your manuscript.

----- Please read the following conditions carefully -----

Before Submission

Please confirm the following terms.

- ✓ **The paper length is more than 8 pages. (Max. 10 pages.)**
(If the page length exceeds 10 pages, the **extra page charge USD50 per extra page** will be requested.)

- ✓ **Your paper is not Survey Research (or Review paper).**
- ✓ **All figures/tabels/equations are original works.**
- ✓ **All figures/tabels are clear.**
(The letter of figures is 10-point Times New Roman.)
- ✓ **The full name of authors is given in your manuscript.**
- ✓ **Manuscript submitted to the IJIES journal signifies that it has not been published previously, has not been copyrighted, has not been submitted elsewhere. The authors declare no conflict of interest.**

Currently, the IJIES is indexed by **SCOPUS**.

(see <https://www.scopus.com/sourceid/21100199790?origin=sbrowse>)

However, please bear in mind that SCOPUS suddenly drop journals from the list. **No refund** will be given even if the IJIES is dropped from SCOPUS list.

From April 1, 2020

Information of IJIES Awards

Excellent Citation Award

The Excellent Citation Award of IJIES is granted to an individual or group who has made significant and important contributions to the International Journal of Intelligent Engineering and Systems (IJIES).

Nomination criteria:

More than 15 citation counts are necessary **within a year** after paper publication.
(e.g. If your paper was published in 2019, more than 15 citation counts are necessary until 31/12/2021.)

How to apply "Excellent Citation Award":

To proof your citation counts, please submit "Author search result" of "Scopus Preview" to IJIES office. (Check <https://www.scopus.com/>)

* The citation is counted by using "Scopus" database.

For Excellnt Citation Award recipients:

Honorary certificate & Free Publication Charge will be provided to an individual or group.

Outstanding Citation Award

The Outstanding Citation Award of IJIES is granted to an individual or group who has made significant and important contributions to the International Journal of Intelligent Engineering and Systems (IJIES).

Nomination criteria:

More than 20 citation counts are necessary within **three years** after paper publication.
(e.g. If your paper was published in 2017, more than 30 citation counts are necessary until 31/12/2020.)

From 01/04/2020

How to apply "Outstanding Citation Award":

To proof your citation counts, please submit "Author search result" of "Scopus Preview" to IJIES office. (Check <https://www.scopus.com/>)

* The citation is counted by using "Scopus" database.

For Outstanding Citation Award recipients:

Honorary certificate will be provided to an individual or group.

INVAS

5

**KOMENTAR
EDITOR TERKAIT
PENERIMAAN
ARTIKEL UNTUK
DIPUBLIKASIKAN**

Review Form

International Journal of Intelligent Engineering and Systems (IJIES)

Paper ID	Ijies
Paper Title	Performance Evaluation of Various Heuristic Algorithms to Solve Job Shop Scheduling Problem (JSSP)

Recommendation for Publication	
<input checked="" type="checkbox"/> (Evaluation A:) Accept	<input checked="" type="checkbox"/> (Evaluation B:) Accept after Minor Revision
<input type="checkbox"/> (Evaluation C:) Accept after Major Revision	<input checked="" type="checkbox"/> (Evaluation D:) Reject

Reply to the editor comments

First of all, we would express our great thanks for the opportunity to resubmit the revised version of our manuscript, "Performance Evaluation of Various Heuristic Algorithms To Solve Job Shop Scheduling Problem (JSSP)" for publication in the *International Journal of Intelligent Engineering and Systems*. We found that your comments and advice have improved and enriched the quality of the paper immensely. In this revised version, we have modified the manuscript following reviewer comments and suggestions. Those changes are highlighted in the document. We have carefully revised some grammatical errors on the earliest version of our paper and have proof-reading. The paper has also checked for similarity (Plagiarism) of the manuscript using Turnitin (percent).

From Editor:

Please improve the reference format. This is very important for indexing service. If you did not follow the following format, your paper will be rejected automatically.

*Do not use "et al." in author names.

e.g.

[1] R. Ruskone, S. Airault, and O. Jamet, "Vehicle Detection on Aerial Images", *International Journal of Intelligent Engineering and Systems*, Vol.1, No.1, pp.123-456, 2009.

(In the case of Journal Papers)

[2] R. Ruskone, L. Guigues, S. Airault, and O. Jamet, "Vehicle Detection on Aerial Images", In: *Proc. of International Conf. On Pattern Recognition*, Vienna, Austria, pp.900-904, 1996.

(In the case of Conference Proceedings)

*Note: e.g. In the case of the author name:"John Doe", express as "J. Doe". ("John" is the first name and "Doe" is the family name.)

- We have added and modified reference on the manuscript following the IJIES format**

Please add "Conflicts of Interest" and "Author Contributions". (see the IJIES format.docx)

Review Form

We have included "Conflicts of Interest" and "Author Contributions" following the IJIES format

- Conflicts of Interest**

The authors declare no conflict of interest.

- Author Contributions**

Admi Syarif supervised the study, analyzed the results, verified the study's findings, and wrote the paper. Mitsuo Gen gives the idea of the algorithm; Renaldi Kumar and Ade Pamungkas designed and performed the experiments;

Finally, we would like to appreciate your great effort in managing this journal.

Bandar Lampung, 2 January 2021

Dr. Eng. Admi Syarif

Evaluation of Paper

Contents	Innovation	<input type="checkbox"/> Highly Innovate <input type="checkbox"/> Sufficiently Innovate <input type="checkbox"/> Slightly Innovate <input type="checkbox"/> Not Novel
	Integrity	<input type="checkbox"/> Poor <input type="checkbox"/> Fair <input type="checkbox"/> Good <input type="checkbox"/> Outstanding
	Presentation	<input type="checkbox"/> Totally Accessible <input type="checkbox"/> Mostly Accessible <input type="checkbox"/> Partially Accessible <input type="checkbox"/> Inaccessible
	Technical depth	<input type="checkbox"/> Superficial <input type="checkbox"/> Suitable for the non-specialist <input type="checkbox"/> Appropriate for the generally knowledgeable individual working in the field <input type="checkbox"/> Suitable only for an expert
Presentation & English	<input type="checkbox"/> Satisfactory <input type="checkbox"/> Needs improvement <input type="checkbox"/> Poor	
Overall organization	<input type="checkbox"/> Satisfactory <input type="checkbox"/> Could be improved <input type="checkbox"/> Poor	

6

**REVIEW PERTAMA
DARI REVIEWER
JURNAL**

Review Form

International Journal of Intelligent Engineering and Systems (IJIES)

Paper ID	Ijies3740
Paper Title	Performance Evaluation of Various Heuristic Algorithms To Solve Job Shop Scheduling Problem (JSSP)

Recommendation for Publication

☐ (Evaluation A:) Accept☐ (Evaluation B:) Accept after Minor Revision☐ (Evaluation C:) Accept after Major Revision☐ (Evaluation D:) Reject

Comments from reviewers 1 & 2:

This is well written and organized paper. It is scientifically sound and contains sufficient interest. However, the novelty of this paper is weak, because this is a comparative study between well-known existing techniques.

Some figures were quoted in this manuscript. However, all figures must be original to protect copyrights. Do the authors have a permission to use for these figures?

Overall, the authors have made a good attempt. However, the authors' proposed method does not adequately describe their data. The results are not supported by any theoretical/mathematical reasons. Readers will fail to understand the scientific contribution of this research. The authors should justify the effectiveness of the proposed technique theoretically. Otherwise, the obtained result seems a simulation report.

Other comments:

1. In the abstract part, the novelty of the proposed method and the concrete results with data should be described.
2. In abstract, the result of this work must be described briefly with data. The result of this work is not clear. The authors only described that "the results validate the usefulness of the heuristic approaches in searching the optimal/near-optimal solutions of JSSP."
3. In the Introduction part, strong points of this proposed method should be further stated. The new features of the proposed method and the main advantages of the results over others should be clearly described. In Sect. 1, then authors described that "This paper aims to investigate the performance of heuristic algorithms for solving JSSP. First, we developed a GA approach and conducted intensive numerical experiments on a set of Benchmark test problems (3 test problem of Fisher dan Thompson [11], and 25 instances of Lawrence [12], taken from OR-Library. Further, the performances of approaches for JSSP are evaluated based on the quality of the solution, the relative error, and the number of instances solved (NIS) optimally for each test problem. It is shown that GA has the merit of high effectiveness and can obtain competitive results with the other heuristics". The reviewer cannot understand why the authors focused on only GA.
4. Unify the font style in sentences.
5. The problem definition of this work is not clear. In Sect.3, the drawbacks of each conventional technique should be described clearly. The authors should emphasize the difference with other methods to clarify the position of this work further.
6. In tables, letters are too small. Unify the font size of letters (more than 10pt). Enlarge or Redraw tables.
7. Please distinguish mathematical expression from programming code. Don't use "*" for multiplication in equations.

Review Form

8. Use clear images for Figs. 3 and 4.
9. The results of this research are not clear in Conclusions. Show the scientific contribution of this work with concrete data.

From Editor:

Please improve the reference format. This is very important for indexing service. If you did not follow the following format, your paper will be rejected automatically.

*Do not use “et al.” in author names.

e.g.

[1] R. Ruskone, S. Airault, and O. Jamet, “Vehicle Detection on Aerial Images”, *International Journal of Intelligent Engineering and Systems*, Vol.1, No.1, pp.123-456, 2009.

(In the case of Journal Papers)

[2] R. Ruskone, L. Guigues, S. Airault, and O. Jamet, “Vehicle Detection on Aerial Images”, In: *Proc. of International Conf. On Pattern Recognition*, Vienna, Austria, pp.900-904, 1996.

(In the case of Conference Proceedings)

*Note: e.g. In the case of the author name: "John Doe", express as "J. Doe". ("John" is the first name and "Doe" is the family name.)

* * Please send your revised manuscript with the response letter for the 2nd review. (Please highlight modifications and additions inside the paper by red font.)

Please add “Conflicts of Interest” and “Author Contributions”. (see the IJIES format.docx)

Conflicts of Interest (Mandatory)

Declare conflicts of interest or state “**The authors declare no conflict of interest.**” Authors must identify and declare any personal circumstances or interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results.

Author Contributions (Mandatory)

For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used as follows: “conceptualization, XXX and YYY; methodology, XXX; software, XXX; validation, XXX, YYY, and ZZZ; formal analysis, XXX; investigation, XXX; resources, XXX; data curation, XXX; writing—original draft preparation, XXX; writing—review and editing, XXX; visualization, XXX; supervision, XXX; project administration, XXX; funding acquisition, YYY”, etc. **Authorship must be limited to those who have contributed substantially to the work reported.**

Evaluation of Paper

Contents	Innovation	<input type="checkbox"/> Highly Innovate <input type="checkbox"/> Sufficiently Innovate <input type="checkbox"/> Slightly Innovate <input type="checkbox"/> Not Novel
	Integrity	<input type="checkbox"/> Poor <input type="checkbox"/> Fair <input type="checkbox"/> Good <input type="checkbox"/> Outstanding

Review Form

	Presentation	<input type="checkbox"/> Totally Accessible <input type="checkbox"/> Mostly Accessible <input type="checkbox"/> Partially Accessible <input type="checkbox"/> Inaccessible
	Technical depth	<input type="checkbox"/> Superficial <input type="checkbox"/> Suitable for the non-specialist <input type="checkbox"/> Appropriate for the generally knowledgeable individual working in the field <input type="checkbox"/> Suitable only for an expert
Presentation & English	<input type="checkbox"/> Satisfactory <input type="checkbox"/> Needs improvement <input type="checkbox"/> Poor	
Overall organization	<input type="checkbox"/> Satisfactory <input type="checkbox"/> Could be improved <input type="checkbox"/> Poor	

7

**LAMPIRAN
JAWABAN PENULIS
UNTUK REVIEW
PERTAMA DARI
REVIEWER JURNAL**



Answer to the reviewer comments

Title: Performance Evaluation of Various Heuristic Algorithms to Solve Job Shop Scheduling Problem (JSSP)

Authors: Admi Syarif, Ade Pamungkas, Renaldi Kumar M, Mitsuo Gen

Dear reviewer 1 and 2

First of all, we would like to thank you for allowing us to submit our revised manuscript, "Performance Evaluation of Various Heuristic Algorithms to Solve Job Shop Scheduling Problem (JSSP)" for publication in the *International Journal of Intelligent Engineering and Systems*. We much appreciate the time and effort that the reviewers dedicated to providing feedback on our manuscript. We found that those comments and advice have improved and enriched the quality of the paper immensely. We have incorporated most of the suggestions made by the reviewers. Those changes are highlighted in the manuscript. Please see below, in blue, for a point-by-point response to the reviewers' comments and concerns. We have also made the proof-reading and similarity check (Plagiarism) of the manuscript using Turnitin (..... percent).

General comments from reviewers 1 & 2:

This is well written and organized paper. It is scientifically sound and contains sufficient interest. However, this paper's novelty is weak because this is a comparative study between well-known existing techniques. Some figures were quoted in this manuscript. However, all figures must be original to protect copyrights. Do the authors have permission to use these figures?

Overall, the authors have made a good attempt. However, the authors' proposed method does not adequately describe their data. The results are not supported by any theoretical/mathematical reasons. Readers will fail to understand the scientific contribution of this research. The authors should justify the effectiveness of the proposed technique theoretically. Otherwise, the obtained result seems a simulation report.

- **We agree with the reviewer's assessment. Accordingly, throughout the manuscript, we have revised the paper following the comment and suggestion.**

Other comments

1. In the abstract part, the novelty of the proposed method and the concrete results with data should be described.

2. In the abstract, the result of this work must be described briefly with data. The result of this work is not clear. The authors only described that "the results validate the usefulness of the heuristic approaches in searching the optimal/near-optimal solutions of JSSP."

- **Thank you for pointing this out. We agree with your opinion. We have made a significant modification in the abstract part of the paper. We have made clear the contribution of this research. We also have a more detailed discussion on the experimental design and the results. We also have added and modified reference on the manuscript following the IJIES format**

3. In the Introduction part, strong points of this proposed method should be further stated. The new features of the proposed method and the main advantages over others should be clearly described. In Sect. 1, the authors described that "This paper aims to investigate the performance of heuristic algorithms for solving JSSP. First, we developed a GA approach and conducted intensive numerical experiments on a set of Benchmark test problems (3 test problem of Fisher dan Thompson [11], and 25 instances of Lawrence [12], taken from OR-Library. Further, the performances of approaches for JSSP are evaluated based on the quality of the solution, the relative error, and the number of instances solved (NIS) optimally for each test problem. It is shown that GA has the merit of high effectiveness and can obtain competitive results with the other heuristics". The reviewer cannot understand why the authors focused only on GA.

- **We agree with the reviewer's assessment. Accordingly, throughout the introduction, we have revised the manuscript. We have clearly described further explanation on strong points and the main advantages of the proposed approach.**

4. Unify the font style in sentences.

- **We are so sorry for this carelessness. The font style has been modified following this suggestion.**

5. The problem definition of this work is not clear. In Sect.3, the drawbacks of each conventional technique should be described clearly. The authors should emphasize the difference between other methods to clarify the position of this work further.

- **Thank you for pointing this out. Accordingly, throughout the manuscript, we have made revisions. We have described clearly the drawbacks of previous heuristic approaches used for solving JSSP, including Particle Swarm Optimization (PSO), Upper-level algorithm (UPLA), Differential-based Harmony Search (DHS), Grey Wolf Optimization (GWO), Ant Colony Optimization (ACO), Bacterial Foraging Optimization (BFO), Parallel Bat Optimization (PBA), and Tabu Search (TS). Next, we introduced strongness and the design of the GA approach. We have more discussion and emphasize the difference between the methods**

6. In tables, letters are too small. Unify the font size of the letters (more than 10pt). Enlarge or Redraw tables.

- **Following your recommendation, we have revised the manuscript.**

7. Please distinguish mathematical expression from programming code. Don't use "*" for multiplication in equations.

■ Thank you for pointing this out. Eq. (7) has been updated

8. Use clear images for Figs. 3 and 4.

Following this suggestion, we have made clear Figs 3 and 4 on the paper.

9. The results of this research are not clear in the Conclusions. Show the scientific contribution of this work with concrete data.

■ Thank you very much. we think this study makes a valuable contribution to the field because :

- In this paper, we also developed a new simple GA to solve JSSP
- This paper determines the performance of various heuristic algorithms to solve JSSP. We investigated several Heuristics' performance based on the solution quality, average error, and the number of instances solved optimally.
- The conclusion has been modified following your suggestion. We also have included the following data have been included.

The PBA is the most effective algorithm that solves 26 instances optimally with an average error of 0.05%, followed by DHS, UPLA, and GA. The computational results show that GA has a good performance in both NIS (22 cases optimally with an average error of 0.05%) and computational time. These findings add to a growing body of literature on the applications of GA.

Evaluation of Paper		
Contents	Innovation	<input type="checkbox"/> Highly Innovate <input type="checkbox"/> Sufficiently Innovate <input type="checkbox"/> Slightly Innovate <input type="checkbox"/> Not Novel
	Integrity	<input type="checkbox"/> Poor <input type="checkbox"/> Fair <input type="checkbox"/> Good <input type="checkbox"/> Outstanding

8

**LAMPIRAN
TRACING RESULT
PERBAIKAN
ARTIKEL**



Performance Evaluation of Various Heuristic Algorithms To Solve Job Shop Scheduling Problem (JSSP)

Admi Syarif^{1*} Ade Pamungkas² Renaldi Kumar³ Mitsuo Gen⁴

^{1 2 3}Department of Computer Science, Faculty of Mathematics and Sciences, Lampung, Indonesia

²Department of Computer Science, Tokyo University of Science, Tokyo, Japan

*Corresponding author's Email: admi.syarif@fmipa.unila.ac.id

Abstract: Scheduling is a famous optimization problem that seeks the best strategy of allocating resources over time to perform jobs/tasks satisfying specific criteria. It exists everywhere in everyday life, particularly in manufacturing or industrial applications. An essential class of scheduling problems is a job shop scheduling problem (JSSP), an NP-hard optimization problem. Several researchers have reported the use of heuristic methods to solve JSSP. However, no one presented an algorithm that can solve the problem optimally all time to our knowledge. In this paper, we investigated aims to investigate the performance of several various heuristic algorithms. We evaluated the merit to solve JSSP. Firstly, we developed a Genetic Algorithm (GA) and the limitation of the approaches by comparing the results compared the performance of some heuristic algorithms, including Particle Swarm Optimization (PSO), Upper-level algorithm (UPLA), Differential-based Harmony Search (DHS), Grey Wolf Optimization (GWO), Ant Colony Optimization (ACO), Bacterial Foraging Optimization (BFO), Parallel Bat Optimization (PBA), and Tabu Search (TS), and the proposed Genetic Algorithm (GA). The performances are measured based on the solution quality, the number of instances solved (NIS) optimally, and the relative error on experimental results of the 28 benchmark test problems, which are of different sizes and difficulty. The results validate the usefulness of the heuristic approaches in searching validated that the algorithms, except ACO, can provide the optimal/near-optimal solutions solution of JSSP. PBA delivers the most impressive performance that solves 26 cases optimally, with the average error equal to 0.05%. Among those 28 test problems, TS, DHS, and PBA can solve 26 instances optimally, followed by GA that solves 21 cases.

Keywords: Combinatorial optimization, Job Shop Scheduling Problem (JSSP), Artificial Intelligence, Heuristic Algorithms, NP-hard Problem

Formatted: Font color: Text 1

1. Introduction

Scheduling is one of the most essential and commonly encountered classes of optimization problems. Scheduling problems exist everywhere in everyday life, particularly in industrial or manufacturing applications. What makes scheduling problems important is that many manufacturing companies or industries have limited resources and have to satisfy specific criteria. Determining an excellent strategy to schedule tasks will reduce production costs or improve profitability.

There are many variations of scheduling problems for several real-world uses. There are, in general, two classes of fundamental scheduling

problems discussed in many pieces of literature. Those are the Job Shop Scheduling Problem (JSSP) and the Flow Shop Scheduling Problem (FSSP). Among them, JSSP has been the most widespread and complex problem. The JSSP model has been vital and practical and challenges many researchers in engineering, computing, and operational [1]. It represents a problem of allocating a set of resources (machine) to perform tasks (job) that consists of m different operations, and the separate device has a distinct the other processing time. The main objective is to determine the best machine schedule to do all job with the best objective value, i.e., minimizing makespan (C_{max}), mean flow time, mean tardiness, earliness, maximum lateness, etc. [2]. The JSSP with

n job and m machine will have $(n!)^m$ possible solutions. Thus, for the relatively large size problem, it will be computationally expensive to solve scheduling problems optimally [3].

Generally, there are two classes of methods for solving JSSP; exact and heuristic methods [4]. The first-class methods include: integer programming [5] [6], Lagrangian relaxation [7], dynamic programming [8], and Branch and bound [9]. Another class is called heuristic algorithms. The term heuristic search was first introduced in early 1960. It was initially concerned with increasing the effectiveness of the problem-solving process. Although those methods do not guarantee the finding of an optimal solution, those have been reported useful in solving many challenging optimization problems within a reasonable computational time.

The term heuristic is usually related to the methods or algorithms for solving problems intelligently. Over the last few decades, with the rapid increase of computer technology, we have seen enormous growth in heuristic approaches to various hard and challenging optimization problems, such as Simulated Annealing (SA), Ant Colony Optimization (ACO), Genetic Algorithm (GA), Branch-and-Bound, Tabu Search (TS), and so on.

Among the heuristics algorithms, GA probably has been the most popular heuristic approach [10]. Aided by GAs, researchers evolve solutions to complex combinatorial optimization problems easily and rapidly. Several researchers reported that the robustness of heuristics for a variety of challenging optimization problems. Despite these interests, no researcher said the best method to solve the problem optimally to the best of our knowledge. This fact does researches on the performance evaluation of the heuristic algorithms for JSSP is still. Our past researches reported the excellent performance of GA in solving various combinatorial optimization problems [11], [12], and [13]. In contrast to other heuristics methods, it utilizes a set population of solutions in its search. It makes GA more robust to solve many real-world problems [10].

For solving JSSP, several researchers have reported that the robustness of heuristics methods. These include the TS algorithm by Mauro [14], Ant Colony Optimization (ACO) by Flórez [15], Bacterial Foraging Optimization (BFO) by Zhao [16], Bat Optimization algorithm by Dao [17], and so on. Despite these interests, however, no researcher said the best method to solve JSSP all-time optimally. This fact shows that researches on the performance evaluation of the heuristic algorithms for JSSP are very crucial.

This paper aims to investigate the performance of heuristic algorithms for solving JSSP. First, we

developed a GA approach and conducted intensive numerical experiments on a set of Benchmark test problems (3 test problem problems of Fisher dan Thompson [11][18], and 25 instances of Lawrence [12], taken from OR Library [19]). Further, we compared the performances results to those of approaches for JSSP are evaluated some heuristic algorithms, including Particle Swarm Optimization (PSO), Upper-level algorithm (UPLA), Differential-based Harmony Search (DHS), Grey Wolf Optimization (GWO), Ant Colony Optimization (ACO), Bacterial Foraging Optimization (BFO), Parallel Bat Optimization (PBA), and Tabu Search (TS). The comparison is made based on the solution's quality of the solution, the relative error, and the number of instances solved (NIS) optimally for each test problem

It is shown that GA has the merit of high effectiveness and can obtain competitive results with the other heuristics.

The organization for the remainder of the paper is as follows: the next section describes the formulation of JSSP. In the third section, we concern ourselves with some essential discussions of several heuristic algorithms, including the GA's working mechanisms of our GA approach. Furthermore, in the fourth section, some comparisons of results from the numerical experiments on Benchmark test problems are presented. We evaluate the algorithms' performance based on the solution quality, the relative error, and the number of instances solved (NIS) optimally. In the end section, we provide the conclusion of this study, showing the approaches; remarkable effectiveness.

2. Mathematical Model of JSSP

Consider the JSSP with m machines to perform n jobs or tasks. Each job/task consists of m operations. The order of operations for the machines is predetermined. The different device is used for a separate action to complete one job. The problem involves designing an effective strategy (called schedule) of assigning some activities to be done by the devices by meeting constraints.

The main objective of JSSP is to determine the best machine schedule to do all jobs with the best objective function, i.e., minimizing makespan(C_{max}), mean flow time, mean tardiness, earliness, and maximum lateness. The most common constraint of the JSSP is as follows [13][20]:

1. A machine can process only a job or task at a time.
2. The machine sequence of the machine to process each job must be the same.
3. The process of a job cannot be interrupted.

Formatted: Font: Not Italic, Font color: Auto

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Field Code Changed

Formatted: Font color: Dark Red

Let t_{ij} and f_{ij} are the starting and the finishing time of processing job j at machine i . P_{ij} is the processing time of machine i to perform job j . The makespan (C_{max}) here represents the finishing time of the last job. The mathematical model of JSSP is as follows [14,21]:

$$\min C_{max} \quad (1)$$

s.t.

$$t_{hj} - t_{ij} \geq P_{ij} \quad (2)$$

$$C_{max} - t_{ij} \geq P_{ij} \quad (3)$$

$$t_{ij} - t_{ik} \geq P_{ik} \quad \text{or} \quad t_{ik} - t_{ij} \geq P_{ij} \quad (4)$$

$$t_{ij} \geq 0 \quad (5)$$

In this model, equation (1) is the objective function to minimize the makespan. The constraint (2) guarantees that the next step of machine h for job j is started after finishing the step at machine i for job j . Next, the constraint (3) ensures the makespan is equal to or greater than the finishing time of the last job. Equation (4) shows that only a machine processed a job at a time. Finally, equation (5) is a non-negative constraint.

3. Heuristic ~~Approach~~Approaches for JSSP

In section, we shall describe clearly the drawbacks of previous heuristic approaches used for solving JSSP, including Particle Swarm Optimization (PSO), Upper-level algorithm (UPLA), Differential-based Harmony Search (DHS), Grey Wolf Optimization (GWO), Ant Colony Optimization (ACO), Bacterial Foraging Optimization (BFO), Parallel Bat Optimization (PBA), and Tabu Search (TS). Next, we introduced the design of the GA approach. We emphasize the difference between the methods to clarify the position of this works.

3.1 ACO (Ant Colony Optimization)

Ant Colony Optimization (ACO) is a heuristic algorithm that combines concepts from Artificial Intelligence and Biology, inspired by ants' collective ~~behavior~~behaviour [15]. Dorigo first introduced ACO for solving the Traveling Salesman Problem. Currently, ~~ACO~~ACO has ~~been used for solved~~been used for solving various fields of ~~engineering our daily life applications~~engineering our daily life applications. The ~~An~~ACO-based method, called Elitist Ant System (EAS) for JSSP, has been carried out among many by Florez in 2013. Each job consists of a sequence of operations, and each process comes with a determined machine and processing time. They adopt the collective intelligence of many simple agents to determine optimal solutions with minimum makespan.

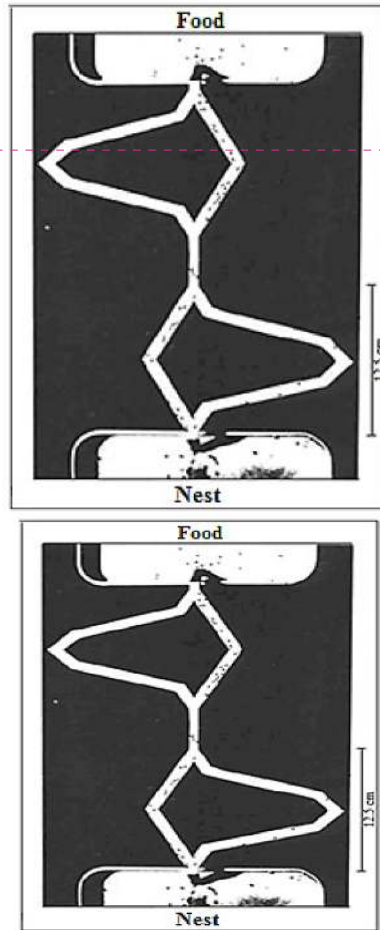


Figure. 1 Illustration of a colony of ants ~~that to~~ find the shortest path ~~to for~~ for the food [15].

They presented the obtained results for each of the JSSP instances by Lawrence [16]. ~~The results are also compared with [22]. They compared the results with those of Tabu Search (TS) and the best-known solution (BKS) taken from [17] and Tabu Search (TS) [23].~~ The algorithm implemented, Elitist Ant System, has proven to be competitive by finding the more reliable quality solutions for JSSP [15]. However, it also requires more effort to obtain the best-known solution for all LA instances.

Field Code Changed

Formatted: English (India)

3.2 Particle Swarm Optimization (PSO)

Another Particle Swarm Optimization (PSO) is a population-based metaheuristic optimization technique is called Partiele Swarm Optimization (PSO) approach, introduced by Eberhart and Kennedy first introduced PSO, inspired by the social behavior of a group of animals, such as fish and birds. To. Animals' behaviour to search for food, each such as birds and fishes, inspires the PSO. Each flock of birds or fishes tends to determine its speed based on personal experience and information obtained through interactions with other members. Pisut Pongchairerks reported the use of Partiele Swarm Optimization for JSSP (JSP-PSO) in 2009. They proposed the GLN-PSO algorithm that enables the swarm to explore different Pongchairerks and Kachitvichyanukul reported the use of PSO to solve JSSP (JSP-PSO) in 2009 [24]. This paper proposed the GLN-PSO algorithm that allows the swarm to explore the other parts of the search spaces simultaneously. To evaluate the algorithm's performance, they had numerical experiments on 33 well-known benchmark 33 instances test problems from Fisher and Thompson: f06, f10, f20, (FT06, FT10, FT20), and the rest from Lawrence. Their computational results show the algorithm can optimally solve the problem 17 times.

3.3 Tabu Search (TS)

Another popular heuristic method for solving combinatorial optimization problems is Tabu Search. Since Glover originally introduced it in 1986, hundreds of researchers reported the success of Tabu Search (TS) applications to various combinatorial optimization problems. It has been reported among the most effective algorithm practical algorithms and provides optimal/near-optimal solutions for many cases. TS is TS searches for the best solution based on the local search method's optimization: the search goes from one to another solution, determining the best solution. A TS algorithm's main components are memory structures, a trace of the search's evolution, and strategies to use the memorized information in the best possible way. The Dell'Amico first introduced the use of TS for solving the JSSP was given by Mauro Dell'Amico [18], [14]. Their basic idea is to avoid cycles in the search's evolution by inhibiting the algorithm from reoccurring more recently made moves. They evaluated TS's performance was evaluated on a set of problem instances, including from Lawrence (LA01-40) [16]. It was shown [22]. Their results show that TS is useful in finding the optimal/near-optimal solutions solutions.

3.4 Upper-Level Algorithm (UPLA)

Nowadays, research on developing the Job-Shop Scheduling Problem (heuristic algorithm for JSSP) has become more variegated. In 2019, Pongchairerks proposed a brand new two-level metaheuristic algorithm, consisting of an upper-level algorithm (UPLA) and a lower-level algorithm (LOLA) for the JSSP. The UPLA is a brand new algorithm that begins with a population of the combinations of values from LOLA's input-parameter. All input-parameter values in every input parameter value combination are represented in real numbers. At every iteration, UPLA attempts to increase its population by utilizing the feedback returned from LOLA. With the support of UPLA Thus, LOLA may improve from a local search algorithm to be an iterated local search algorithm.

Furthermore, UPLA and LOLA result in the two-level algorithm, which may adapt to every JSP instance. The performance of UPLA was assessed on 53 well-known JSSP benchmark instances, including FT06, FT10, FT20, and LA01-LA40. The authors also compared their algorithm results with those of the two-level PSO of [19].

Alike the other population based algorithms, UPLA examines a real number search space based on the population of the combination from the input parameter. Nevertheless, increasing the UPLA's community is different because UPLA is developed for being a parameter controller for LOLA. Among JSSP algorithms, the most similar algorithm to the proposed algorithm was the two-level Particle Swarm Optimization (PSO). The correspondence is that they generate parameterized active schedules with similar methods on both their lower level algorithms; furthermore, parameters for both algorithms (lower and upper level algorithms) control the identical. However, the two-level PSO [19] is different from the suggested two-level metaheuristic algorithm in that it uses the GLN-PSO's framework [20] within both its levels. Due to their similarity and difference just considered, this paper then decides the two-level PSO to compare with the proposed two-level metaheuristic algorithm in their performances [21].

3.5 Differential-Based Harmony Search (DHS)

Fuqing Zhao reported the Differential-based Harmony Search (DHS) to minimize makespan for job shop scheduling problems in 2018. DHS method is based on the critical-path blocks to combine the differential-based improve the mechanism of variable neighborhood search (VNS). The transformed VNS, on the critical path, is embedded into DHS to seek a

Formatted: English (India)

more reliable solution, which is based on the blocks. It is purposed for solving JSSP with the objective of minimized makespan. The DHS's performances are also evaluated on a set of benchmark instances from the OR-library [22]. Compared with various HS-based algorithms and other state of the art algorithms, the DHS is superior in solution quality, convergence speed, and stability [23].

Furthermore, UPLA and LOLA result in the two-level algorithm, which may adapt to every JSSP instance. Similar to the other population-based algorithms, UPLA examines search space based on the population of the combination from the input parameter. Real numbers represent all input-parameter values.

Among JSSP algorithms, the most similar algorithm to the proposed algorithm was the two-level Particle Swarm Optimization (PSO) [24]. The correspondence is that they generate parameterized-active schedules with similar methods on both their lower-level algorithms; furthermore, parameters for both algorithms (lower and upper-level algorithms) control the identical. However, the two-level PSO is different from the suggested two-level metaheuristic algorithm that uses GLN-PSO's framework [25]. The authors assessed the algorithms' performance on 53 well-known benchmark instances, including FT06, FT10, FT20, and LA01-LA40 [26]. Considering the similarity and difference, they also compared their results with those of the two-level PSO [24].

3.5 Differential-Based Harmony Search (DHS)

The Differential-based Harmony Search (DHS) to minimize makespan for JSSP was reported by Zhao in 2018 [16]. The DHS improves the variable neighbourhood search (VNS) based on the critical path blocks. The transformed VNS, on the critical path, is embedded into the DHS to seek a more reliable solution based on the blocks. They evaluate DHS's performances on a set of benchmark instances from the OR-library with the objective of minimized makespan [27]. Compared with various HS-based algorithms and other state-of-the-art algorithms, the DHS is superior in solution quality, convergence speed, and stability [16].

3.6 GWO (Grey Wolf Optimization)

The Grey wolf optimization (GWO) algorithm is a new population-oriented heuristic algorithm inspired by grey wolves' social hierarchy and hunting behavior. To dealing with optimization problems, in 2018, behaviour. Tianhua Jiang introduced GWO, a brand new swarm-based intelligence algorithm. It is intended initially, to solve constant deal with

optimization problems in 2018. The GWO algorithm is based on the crossover operation and adapted adapt the searching operator to minimize the makespan (maximum completion time (makespan). To keep the population's variety, they also). They also introduced an adaptive modification method to the algorithm compared to keep the variety of population. They compared the results with other published algorithms in the two scheduling cases. According to the experimental results, GWO provided better solutions for some instances [24],[28].

3.7 Bacterial Foraging Optimization (BFO)

Kim Passino proposed another intelligent heuristic algorithm, called the Bacterial Foraging Optimization (BFO), in 2002 [29]. The BFO is inspired by the cooperation and competition behaviours of a bacterium named *E. coli* in seeking food. One of the primary processes in BFO is the evolution process. It begins when the bacterium migrates to a better solution according to the advantaged group's activity. Zhou introduced an algorithm called Chemotaxis-enhanced-BFO (CEBFO) to solve the JSSP [30]. To improve the algorithm's performance, they include a local search operation and chemotaxis with the differential evolution (DE). They conducted some numerical experiments on a set of benchmark problems of JSSP. The results demonstrated a good understanding of the algorithm.

3.8 Parallel Bat Algorithm (PBA)

In 2015, Dao proposed a parallel-based heuristic algorithm version, called the parallel bat algorithm (PBA) [17]. The fundamental structure of the PBA is to divide the distribution of the bat populations into several groups. They offered three schemes, namely a random-key encoding, a makespan, and a communication strategy. To examine the method's accuracy, they had some experiments on 43 (forty-three) benchmark instances (Fisher and Thompson with FT06, FT10, FT20 [18], and Lawrence (LA01–LA40) [31]). They compared their experimental results are to those of the PSO algorithm. These show that the intended approach gives competitive returns.

3.9 Genetic Algorithm (GA)

Since Holland introduced GA in 1975, it has witnessed many exciting advances in using Genetic Algorithms (GAs) to solve challenging optimization problems in everything from production design to inventory and network design problems. It is a

heuristics method, inspired by the process of Darwinian evolution. GA has been a multi-purpose approach for searching the global optimality; adapting GAs to a specific optimization problem is challenging but frustrating. The selection methods, efficient design of the chromosome representation, crossover and mutation process, and GA parameters' value influence GA's success [32]. Therefore, discovering an efficient GA approach system for a particular problem becomes essential in GA research.

Kim Passino proposed another intelligent heuristic algorithm in 2002 is The Bacterial Foraging Optimization (BFO). The BFO is based on the cooperation and competition behaviors of a bacterium named *E. coli* in seeking food. One bacterium's location corresponds to a solution to the problem. One of the primary processes in BFO is called the evolution process. This process begins when the bacterium migrates to a better solution according to the advantaged group's activity. An algorithm called Chemotaxis enhanced BFO (CEBFO) to solve the JSSP was proposed by Fuqing Zhao [23]. This algorithm is based on chemotaxis with the differential evolution (DE) operator added. A local search operation was included to improve the performance of the algorithm significantly. The experiments were conducted on a set of benchmark problems of JSSP. The results demonstrated a good understanding of the algorithm.

3.8.1.1 Parallel Bat Algorithm (PBA)

To share the computation load, Dao proposed the parallel bat algorithm (PBA) by dividing the artificial bats into independent subpopulations. The fundamental structure of the Bat Algorithm (BA) is bat populations distribution into several groups. It plays an essential role in practical computations of optimization. In 2015, Dao proposed a parallel-based heuristic algorithm version of the BA. They offered a random-key encoding scheme, a makespan scheme, and a communication strategy scheme. In their experiments, forty-three instances of the benchmark data set are used to examine the method's accuracy. Forty-three test problems from two classes of standard JSSP test problems included Fisher and Thompson with FT06, FT10, FT20, [11], and Lawrence (LA01-LA40) are used for experiments [25]. Dao compared outcomes using the BA and PSO methods. The comparison shows that the intended process gives competitive returns. [26].

3.9 Genetic Algorithm (GA)

Since Holland introduced GA in 1975, it has witnessed many exciting advances in the use of

Genetic Algorithms (GAs) to solve challenging optimization problems in everything from production design to inventory and network design problems. It is a heuristics method, inspired by the process of Darwinian evolution. In contrast to other heuristics methods, the GA utilizes a set population of solutions in its search. Aided by GAs, researchers evolve solutions to complex combinatorial optimization problems easily and rapidly. Our past research reported that GA's succeed in solving various combinatorial optimization problems [27],[28], and [29]. GA has been a multi-purpose approach for searching the global optimality; adapting GAs to a specific optimization problem is challenging but frustrating. The selection methods, efficient design of the chromosome representation, crossover and mutation process, and GA parameters' value influence GA's success [30]. Therefore, discovering an efficient GA approach system for a particular problem becomes essential in GA research. Countless efforts should be made to get an efficient implementation of GAs to the problem.

3.9.1. The Chromosome Representation

When implementing GA for an optimization problem, an important issue is how to generate a chromosome that would bring us to the right solution. For the initial population, we have to create a *pop_size* chromosome. Each chromosome consists of $n \times m$ gen generated randomly, and each *job* will appear *m* in the chromosome. One illustrates an example of the chromosome for the test problem FT06, having six jobs and six machines.

The chromosome in Figure. 2 indicates that the first activity to be scheduled is job two at machine one, followed by job six at machine one; then, job three at machine one, and so on, according to the order arranged in the chromosome list. $O_{i,j}$ represents the operation for the job *i* at machine *j*.

3.9.2. Genetic Operations

Procedure: Self Crossover:

Step 1: Chose a parent arbitrarily for crossover.

Step 2: Determine two crossover points randomly

Step 3: Move substring between the above two points

The mutation operation is an essential feature of GA to maintain the chromosome's diversity in the generation. This paper adopts the swap mutation that randomly selects two ~~gens~~genes in the chromosome and then exchanges them.

Formatted: Font: Not Bold

3.9.3. Evaluation and Selection

In GA implementation, ~~each chromosome is evaluated~~ we computed the fitness value to measure how well each chromosome fits the problem's requirements. ~~We For JSSP, we can use the makespan as the fitness value. Since the JSSP is a minimization problem, the computation of fitness value is done using the following formula as follows:~~

$$\text{Fitness}(x) = \frac{1}{f(x)} \quad (6)$$

where $f(x)$ is the objective function (makespan).

The decoding process to compute the makespan (C_{\max}) of the schedule is ~~done~~ as follows:

Step 1: Select the chromosome for the decoding process.

Step 2: Read gen in the chromosome started from the left.

Step 3: Determine the machine number from the machine-order matrix, based on the job operation number's information.

Step 4: Determine the processing time from the processing time matrix based on the job operation number's data.

Step 5: Determine the maximum time of the last job time.

Step 6: Renew the current job finishing time by adding the time to the result of Step 5;

Step 7: Repeat Step 2 to Step 6 until the last-gen in the chromosome.

Another essential process of GA is the way to determine the chromosome for the next population. Of course, the selection process should be done based on the fitness value. ~~Several~~ There have been several selection strategies ~~have been~~ introduced in GA applications. Here, we adopt the elitist approach by selecting the best *pop_size* chromosome for the next generation.

Formatted: Font: TimesNewRoman, Font color: Black

Formatted: Font: Times New Roman, Font color: Auto

Formatted: Font: Times New Roman, Not Italic

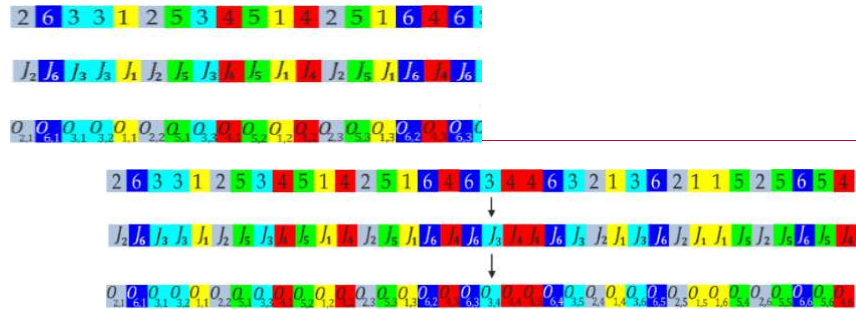


Figure. 2 An example of the chromosome for problem FT06

Formatted: Font: +Body (Calibri), English (United States)

Formatted: English (United States)

4. Numerical Experiments and Results

4.1 Design of numerical experiments

To evaluate the effectiveness and the efficiency of the algorithm, we first have some numerical experiments for GA on 28 Benchmark test problems: 3 instances (FT06, FT10, dan FT20) of Fisher dan Thompson [14][18], and 25 instances (LA01-LA25) of Lawrence [12][19], taken from the OR-library [22]. We implement the algorithm in MATLAB R2015b and run on an Intel Core i5 processor of 2.53 GHz and 3-GB of RAM [27]. We implement the algorithm in MATLAB R2015b and run on an Intel Core i5 processor of 2.53 GHz.

The GA parameters are set as: crossover probability (p_C) = 0.4, mutation probability (p_M) = 0.2, population size (*pop_size*) = 400 and maximum generation (*max_gen*) = 10-2000, for each test problem, the experiments are conducted 10 (ten) times. Table 1 presents the overall results obtained for all test problems, where the best and the average values represent the best and the average fitness value from the 10 (ten) running times. BKS represents the best-known solution ~~to in~~ the literature.

4.2 Results and Discussion

Table 1. The experimental results of the GA on 43 Benchmark test problems approach

Test problem	$n \times m$	ACT*	BKS*	GA	Error (%)
--------------	--------------	------	------	----	-----------

Split Cells

Formatted: Font: 10 pt

Formatted: Indent: Hanging: 0.15 cm, Tab stops: 1.05 cm, Left + Not at 0.75 cm

msCas				Best	Average	Worst	▲
FT06	6 × 6	1,00	55	55	55	55	0,00
FT10	10 × 10	2,76	930	951	988.8	1030	2,26
FT20	20 × 5	1,08	1165	1178	1184.3	1197	1,12
LA01	10 × 5	6,00	666	666	666	666	0,00
LA02	10 × 5	3,97	655	655	658.8	666	0,00
LA03	10 × 5	6,44	597	597	611	621	0,00
LA04	10 × 5	3,35	590	590	592	601	0,00
LA05	10 × 5	1,8	593	593	593	593	0,00
LA06	15 × 5	2,00	926	926	926	926	0,00
LA07	15 × 5	9,00	890	890	890	890	0,00
LA08	15 × 5	9,00	863	863	863	863	0,00
LA09	15 × 5	1,7,0	951	951	951	951	0,00
LA10	15 × 5	0,80	958	958	958	958	0,00
LA11	20 × 5	2,80	1222	1222	1222	1222	0,00
LA12	20 × 5	2,80	1039	1039	1039	1039	0,00
LA13	20 × 5	2,60	1150	1150	1150	1150	0,00
LA14	20 × 5	1,00	1292	1292	1292	1292	0,00
LA15	20 × 5	20,00	1207	1207	1207	1207	0,00
LA16	10 × 10	767	945	959	977.2	997	0,00
LA17	10 × 10	774	784	784	788.9	797	0,00
LA18	10 × 10	808	848	848	868.5	909	0,00
LA19	10 × 10	1,395	842	842	850	874	0,00
LA20	10 × 10	1,234	902	907	928.4	992	0,55
LA21	15 × 10	1,743	1046	1061	1097	1114	1,43
LA22	15 × 10	1,443	927	943	987.8	1046	1,08
LA23	15 × 10	752	1032	1032	1035.3	1054	0,00
LA24	15 × 10	1,122	935	948	977	994	1,39

LA25	15 × 10	2,049	977	987	1015,8	1042	1,02
Average							0,3162

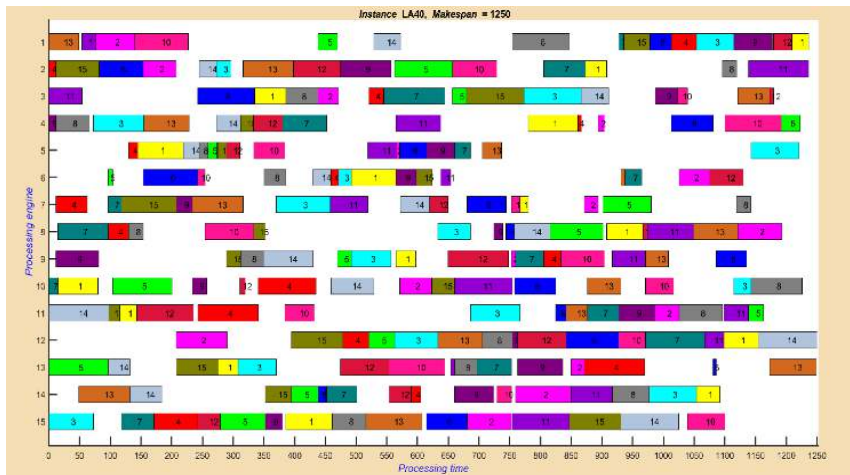
*ACT: Average Computational Time (in second)

In the above table, the error is computed by using the following formula:

$$\text{Error} = \frac{(\text{Best} - \text{Optimum}) * 100\%}{\text{Optimum}} = \frac{(\text{Best} - \text{Opti}) * 100\%}{\text{Opti}}$$

From the above results, one can notice the excellent performance of GA to solve JSSP. Despite it not reaching the optimal solution all-time, it still gives GA presents the optimal solutions (21 instances), with an average error of less than 0.32 percent. The results also show that GA can provide solutions to the problems within reasonable computational time. However, for some hard/difficult cases, GA can obtain near-optimal solutions with an error from 0.5 to 1.43 percent. More efforts can be made to improve the solutions by possibly hybridizing GA with other local search techniques. In the first place, for solving large-scale scheduling problems, the DHS's performance needs to enhance further.

The Gantt chart schedule and the convergence of the solution for LA40 are illustrated in Figure. 3 and Figure. 4, respectively.



Split Cells

Formatted: Font: 10 pt

Formatted: Tab stops: Not at 0.75 cm

Formatted: Font: 10 pt, Not Italic

Formatted: Font: 10 pt

Formatted: Indent: Left: -0.2 cm, Right: -0.2 cm

Formatted Table

Formatted: Font: 10 pt, Bold

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

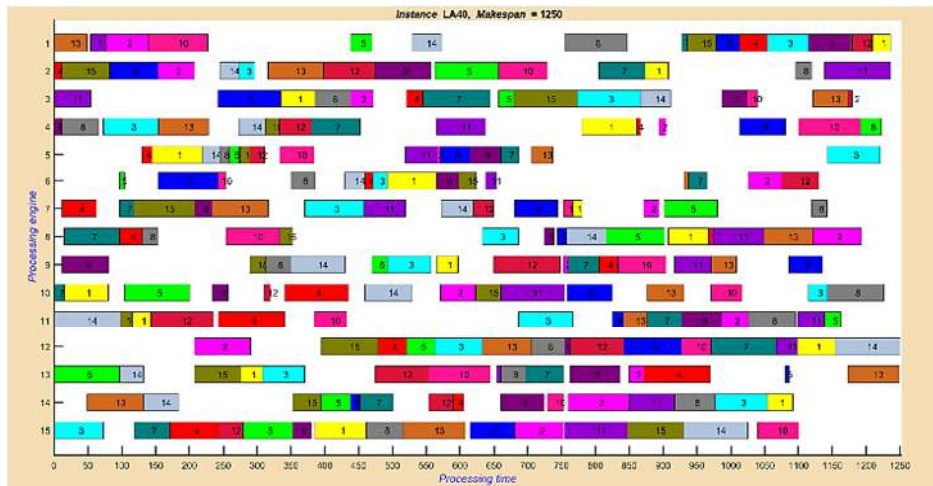
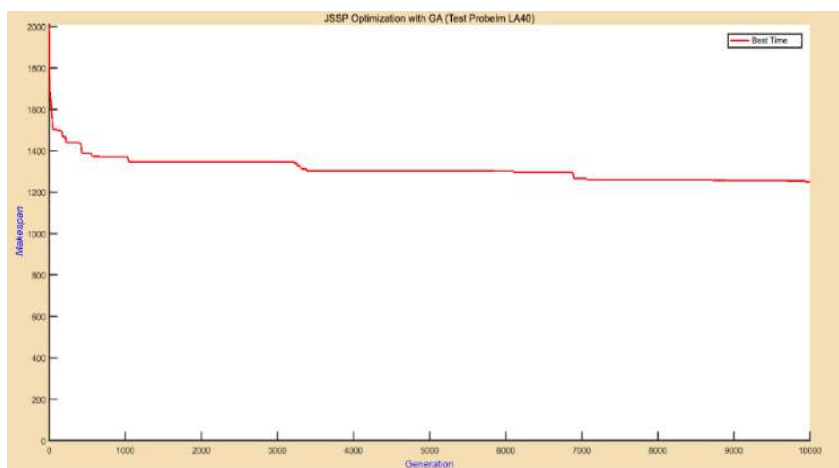


Figure. 3 Gantt chart schedule for LA40



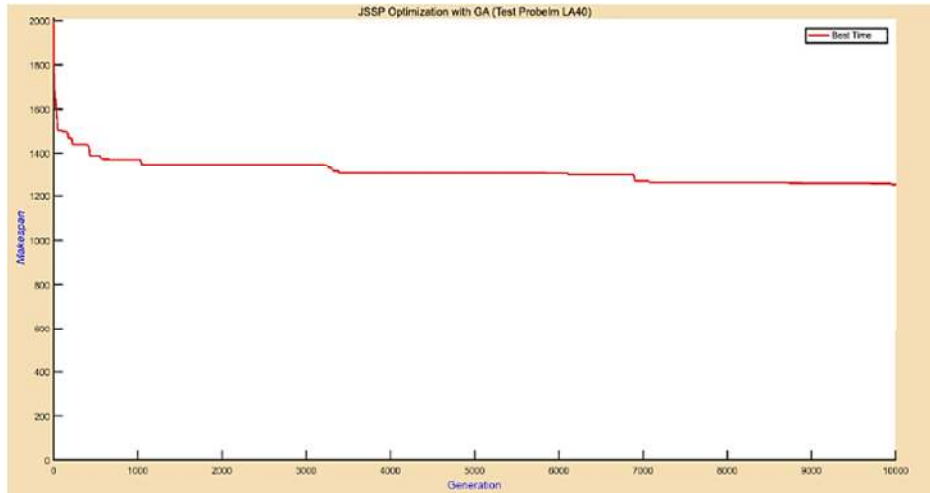


Figure. 4 The convergence of the objective function in the generation for LA40

4.3 Comparison of some heuristic methods

In this research, we evaluate the merit and the limitation of the approaches by comparing the results of some heuristic algorithms, including Artificial Immune System (AIS), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Tabu Search (TS), Upper-level algorithm (UPLA), Differential-based Harmony Search (DHS), Grey Wolf Optimization (GWO), Bacterial Foraging Optimization (BFO), Parallel Bat Optimization (PBA), and the proposed Genetic Algorithm (GA).

The performances are measured based on the solution quality, the number of instances solved (NIS) optimally, and the relative error. The comparisons are also made on a comparison of the results for 28 benchmark test problems—(FT06, FT10, and F20 (Fisher & Thompson (1963))) and 25 instances (LA01-LA40LA25) of Lawrence [12]–[19]. Table 2 summarizes the comparative results. In this research, we also computed the percentage relative error concerning BKS. The results are, as shown in the following Table 3.

u

Table 2. Performance of the heuristic approaches on all instances

Test Problem	Size ($n \times m$)	BKS*	ACO [15]	PSO [31][33]	TS [18][14]	UPLA [21][26]	DHS [23][16]	GWO [24][28]	BFO [32][30]	PBA [26][17]	GA (Proposed)
FT06	6 × 6	55	-	55	-	55	55	55	55	55	55
FT10	10 × 10	930	-	951	-	930	930	940	937	930	951
FT20	20 × 5	1165	-	1191	-	1165	1165	1178	1171	1165	1178
LA01	10 × 5	666	666	666	666	666	666	666	666	666	666
LA02	10 × 5	655	669	663	655	655	655	655	655	655	655
LA03	10 × 5	597	623	603	597	597	597	597	597	597	597
LA04	10 × 5	590	611	611	590	590	590	590	590	590	590
LA05	10 × 5	593	593	593	593	593	593	593	593	593	593
LA06	15 × 5	926	926	926	926	926	926	926	926	926	926
LA07	15 × 5	890	890	890	890	890	890	890	890	890	890
LA08	15 × 5	863	863	863	863	863	863	863	863	863	863
LA09	15 × 5	951	951	951	951	951	951	951	951	951	951
LA10	15 × 5	958	958	958	958	958	958	958	958	958	958
LA11	20 × 5	1222	1222	1222	1222	1222	1222	1222	1222	1222	1222
LA12	20 × 5	1039	1039	1039	1039	1039	1039	1039	1039	1039	1039
LA13	20 × 5	1150	1150	1150	1150	1150	1150	1150	1150	1150	1150
LA14	20 × 5	1292	1292	1292	1292	1292	1292	1292	1292	1292	1292

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

LA15	20 × 5	1207	1212	1207	1207	1207	1207	1207	1207	1207	1207
LA16	10 × 10	945	1005	959	945	945	945	956	945	945	945
LA17	10 × 10	784	812	784	784	784	784	790	785	784	784
LA18	10 × 10	848	885	848	848	848	848	859	848	848	848
LA19	10 × 10	842	875	857	842	842	842	845	844	842	842
LA20	10 × 10	902	912	910	902	902	902	937	907	902	907
LA21	15 × 10	1046	1107	1074	1048	1052	1046	1090	-	1046	1061
LA22	15 × 10	927	1018	944	933	927	927	970	-	933	937
LA23	15 × 10	1032	1051	1032	1032	1032	1032	1032	-	1032	1032
LA24	15 × 10	935	1011	971	941	941	979	982	-	941	948
LA25	15 × 10	977	1062	987	979	982	1016	1008	-	977	987

Table 3. The Error Comparison-Error of the heuristic approaches in all instances

Test Problem	Dimensi (n × m)	ACO	PSO	TS	UPLA	DHS	GWO	BFO	PBA	GA
FT06	6 × 6	-	0.00	-	0.00	0.00	0.00	0.00	0.00	0.00
FT10	10 × 10	-	2.26	-	0.00	0.00	1.08	0.75	0.00	2.26
FT20	20 × 5	-	2.23	-	0.00	0.00	1.12	0.52	0.00	1.12
LA01	10 × 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA02	10 × 5	2.14	1.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA03	10 × 5	4.36	1.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA04	10 × 5	3.56	3.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA05	10 × 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA06	15 × 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA07	15 × 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA08	15 × 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA09	15 × 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA10	15 × 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA11	20 × 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA12	20 × 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA13	20 × 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA14	20 × 5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA15	20 × 5	0.41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
LA16	10 × 10	6.35	1.48	0.00	0.00	0.00	1.16	0.00	0.00	0.00
LA17	10 × 10	3.57	0.00	0.00	0.00	0.00	0.77	0.13	0.00	0.00
LA18	10 × 10	4.36	0.00	0.00	0.00	0.00	1.30	0.00	0.00	0.00
LA19	10 × 10	3.92	1.78	0.00	0.00	0.00	0.36	0.24	0.00	0.00
LA20	10 × 10	1.11	0.89	0.00	0.00	0.00	3.88	0.55	0.00	0.55
LA21	15 × 10	5.83	2.68	0.19	0.57	0.00	4.21	-	0.00	1.43
LA22	15 × 10	9.82	1.83	0.65	0.00	0.00	4.64	-	0.65	1.08
LA23	15 × 10	1.84	0.00	0.00	0.00	0.00	0.00	-	0.00	0.00
LA24	15 × 10	8.13	3.85	0.64	0.64	4.71	5.03	-	0.64	1.39
LA25	15 × 10	8.70	1.02	0.20	0.51	3.99	3.17	-	0.00	1.02

We also analyze the algorithm based on the average errors and the number of instances solved (NIS) optimally. We illustrated the comparison of the average errors and obtained NIS by the algorithms in Figure. 5 and Figure. 6, respectively. These results indicate that, though no algorithm can give the optimal solution, the algorithms effectively find the optimal/near-optimal solutions to the problems. Almost all algorithms, except ACO, can provide an error of less than 1 percent. PBA presents the most impressive performance that can solve 26 cases optimally, with the average error equal to 0.05%.

Among those 28 test problems, TS, DHS, and PBA can solve 26 instances optimally, followed by GA that solves 22 cases.

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 11 pt

Formatted: Font: 11 pt

Formatted: Font: 11 pt

Formatted: Font: 11 pt

Formatted: Font: 11 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

Formatted: Font: 10 pt

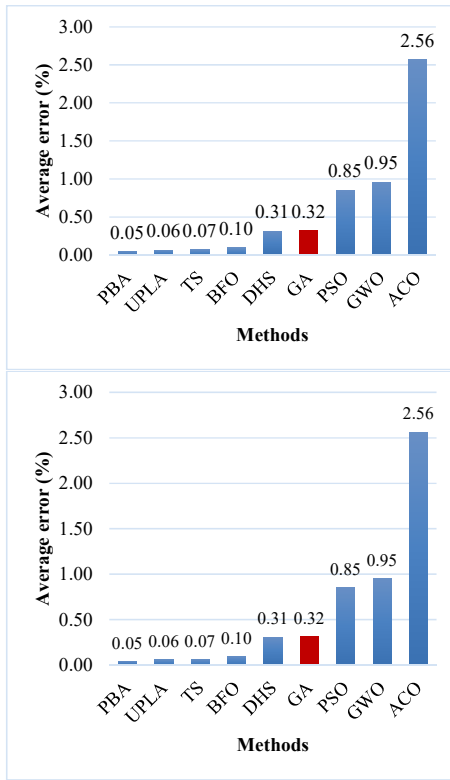


Figure. 5-Average The comparison of average error

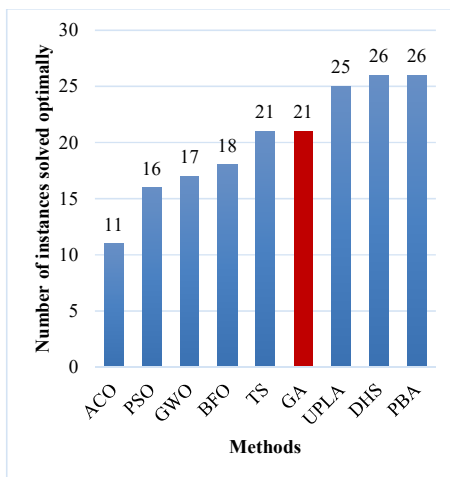


Figure. 6 The comparison of Number of Instances Solved (NIS) Optimally by the heuristic methods

5. Conclusion

This paper analyzed the performance of some heuristic algorithms for solving JSSP. ~~We~~First, we developed the GA approach and conducted some intensive numerical experiments ~~using on~~ a set of JSSP-Benchmark instances from the literature. We investigated ~~the some~~ heuristic methods' performance, based on the solution quality, the relative error, and the number of instances solved (NIS) optimally. The results validate that, ~~though no method presents optimal solutions at all times, the heuristic algorithms are useful and~~heuristics are robust in searching for the optimal solutions of JSSP. ~~Those demonstrate that the heuristic algorithms can search the optimal solutions of JSSP. However, no algorithm can give optimal solutions at all times. Among them, the Tabu search algorithm PBA is the most effective algorithm to solve JSSP, that solves 26 instances optimally with an average error of 0.05%, followed by proposed DHS, UPLA, and GA and PSO. It is shown. The computational results show that the proposed GA approach can solve can obtain competitive results in both NIS (21 cases among 28 instances optimally, with an average error of 0.32%) and computational time.~~ These findings add to a growing body of literature on the applications of GA-heuristics.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Admi Syarif supervised the study, analyzed the results, verified the study's findings, and wrote the paper. Mitsuo Gen ~~gives~~gave the Algorithm; Admi Syarif ~~designed idea of the experiments~~algorithm; Renaldi Kumar ~~M.~~ and Ade Pamungkas ~~designed and~~ performed the experiments;

Acknowledgment

Acknowledgement

This research was supported by the Scientific "Riset Unggulan" Program, Ministry of Education and Culture, Lampung University, the Grant-in-Aid 2226/UN26.21/PN/2019, Indonesia, 2019. The authors thank the anonymous reviewers for their valuable comments and suggestions on this paper.

References

- [1] J. K. Lenstra and A. H. G. Rinnooy Kan, "Computational Complexity of Discrete Optimization Problems", *Annals of Discrete Mathematics*, Vol. 4, pp. 121–140, 1979.
- [2] J. F. Gonçalves, J. J. De Magalhães Mendes, and M. G. C. Resende, "A hybrid genetic algorithm for the job shop scheduling problem", *European Journal of Operational Research*, Vol. 167, No. 1, pp. 77–95, 2005.
- [3] E. A. C. Uzorh and N. Innocent, "Solving Machine Shops Scheduling Problems using Priority Sequencing Rules Techniques", *The International Journal of Engineering And Science*, Vol. 3, No. 6, pp. 1813–2319, 2014.
- [4] K. Akram, K. Kamal, and A. Zeb, "Fast simulated annealing hybridized with quenching for solving job-shop scheduling problem", *Applied Soft Computing*, Vol. 49, pp. 510–523, 2016.
- [5] C. Özgüven, Y. Yavuz, and L. Özbakir, "Mixed-integer goal programming models for the flexible job-shop scheduling problems with separable and non-separable sequence-dependent setup times", *Applied Mathematical Modelling*, Vol. 36, No. 2, pp. 846–858, 2012.
- [6] D. Catanzaro, L. Gouveia, and M. Labbé, "Improved integer linear programming formulations for the job Sequencing and tool Switching Problem", *European Journal of Operational Research*, Vol. 244, No. 3, pp. 766–777, 2015.
- [7] P. Baptiste, M. Flamini, and F. Sourd, "Lagrangian Bounds for Just-In-Time Job-Shop Scheduling", *Computers & Operations Research*, Vol. 35, No. 3, pp. 906–915, 2008.
- [8] J. A. S. Gromicho, J. J. Van Hoorn, F. Saldanha-Dada-Gama, and G. T. Timmer, "Solving the job-shop scheduling problem optimally by dynamic programming", *Computers & Operations Research*, Vol. 39, No. 12, pp. 2968–2977, 2012.
- [9] P. Brucker, E. K. Burke, and S. Groenemeyer, "A branch and bound algorithm for the cyclic job-shop problem with transportation", *Computers & Operations Research*, Vol. 39, No. 12, pp. 3200–3214, 2012.
- [10] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons, New York, 2000.
- [11] A. Syarif, Y. S. Yun, and M. Gen, "Study on multi-stage logistic chain network: A spanning tree-based genetic algorithm approach", *Computers and Industrial Engineering*, Vol. 43, No. 1–2, pp. 299–314, 2002.
- [12] M. Gen and A. Syarif, "Double Spanning Tree-Based Genetic Algorithm for Two-Stage Transportation Problem", *International Journal of Knowledge-Based and Intelligent Engineering Systems*, Vol. 7, No. 4, pp. 214–221, 2003.
- [13] A. Syarif, D. Anggraini, K. Muludi, Wamiliana, and M. Gen, "Comparing Various Genetic Algorithm Approaches For Multiple-Choice Multi-Dimensional Knapsack Problem (mm-KP)", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 455–462, 2020.
- [14] M. Dell'Amico and M. Trubian, "Applying tabu search to the job-shop scheduling problem", *Annals of Operations Research*, Vol. 41, No. 3, pp. 231–252, 1993.
- [15] E. Flórez, W. Gómez, L. Bautista, E. Florez, W. Gomez, and Lola Bautista, "An Ant Colony Optimization Algorithm For Job Shop Scheduling Problem", *International Journal of Artificial Intelligence & Applications (IJAI)*, Vol. 4, No. 4, pp. 53–66, 2013.
- [16] F. Zhao, S. Qin, G. Yang, W. Ma, C. Zhang, and H. Song, "A Differential-Based Harmony Search Algorithm with Variable Neighborhood Search for Job Shop Scheduling Problem and Its Runtime Analysis", *IEEE Access*, Vol. 6, pp. 76313–76330, 2018.
- [17] T. K. Dao, T. S. Pan, T. T. Nguyen, and J. S. Pan, "Parallel bat algorithm for optimizing makespan in job shop scheduling problems", *Journal of Intelligent Manufacturing*, Vol. 29, No. 2, pp. 451–462, 2018.
- [18] H. Fisher and G. L. Thompson, "Probabilistic Learning Combinations of Local Job-Shop Scheduling Rules", *Industrial Scheduling*, Vol. 3, No. 2, pp. 225–251, 1963.
- [19] S. Lawrence, "Supplement to Resource-Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques", *Energy Procedia*, Vol. 4, No. 7, pp. 4411–4417, 1984.
- [20] H. Zaher, N. Ragaa, and H. Sayed, "A novel Improved Bat Algorithm for Job Shop Scheduling Problem", *International Journal of Computer Applications*, Vol. 164, No. 5, pp. 24–30, 2017.
- [21] K. Ploydanai and A. Mungwattana, "Algorithm for Solving Job Shop Scheduling Problem Based on machine availability constraint", *International Journal on Computer*

- Science and Engineering (IJCE), Vol. 02, No. 05, pp. 1919–1925, 2010.
- [15] E. Flórez, W. Gómez, L. Bautista, E. Flórez, W. Gómez, and Ms. Lola Bautista, "An Ant Colony Optimization Algorithm For Job Shop Scheduling Problem", *International Journal of Artificial Intelligence & Applications (IJAA)*, Vol. 4, No. 4, pp. 52–66, 2012.
- [16] L. Davis, *Job Shop Scheduling with Genetic Algorithms*, Psychology Press, East Sussex, United Kingdom, 1985.
- [17] D. Applegate and W. Cook, "Computational study of the job-shop scheduling problem", *ORSA Journal on Computing*, Vol. 3, No. 2, pp. 149–156, 1991.
- [18] M. Dell'Amico and M. Trubian, "Applying tabu search to the job shop scheduling problem", *Annals of Operations Research*, Vol. 41, No. 3, pp. 231–252, 1993.
- [19] P. Pongchairerks and V. Kachitvichyanukul, "A two-level Particle Swarm Optimization algorithm on Job-Shop Scheduling Problems", *International Journal of Operational Research*, Vol. 4, No. 4, pp. 390–411, 2009.
- [20] P. Pongchairerks and V. Kachitvichyanukul, "A Non-Homogenous Particle Swarm Optimization With Multiple Social Structures", In: *Proc. of the 2005 International Conf. on Simulation and Modeling*, Nakompathom, Thailand, pp. 132–136, 2005.
- [21] P. Pongchairerks, "A two-level metaheuristic algorithm for the job-shop scheduling problem", *Complexity*, Vol. 2019, pp.1–11, 2019.
- [22] <http://people.brunel.ac.uk/~mastijb/jeb/orlib/files/jobshop1.txt> (accessed Feb. 07, 2020).
- [23] F. Zhao, S. Qin, G. Yang, W. Ma, C. Zhang, and H. Song, "A Differential Based Harmony Search Algorithm with Variable Neighborhood Search for Job Shop Scheduling Problem and Its Runtime Analysis", *IEEE Access*, Vol. 6, pp. 76313–76330, 2018.
- [24] T. Jiang and C. Zhang, "Application of Grey Wolf Optimization for Solving Combinatorial Problems: Job Shop and Flexible Job Shop Scheduling Cases", *IEEE Access*, Vol. 6, pp. 26231–26240, 2018.
- [25] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control", *IEEE Control Systems Magazine*, Vol. 22, No. 3, pp. 52–67, 2002.
- [30] F. Zhao, X. Jiang, C. Zhang, and J. Wang, "A chemotaxis-enhanced bacterial foraging algorithm and its application in job shop scheduling problem", *International Journal of Computer Integrated Manufacturing*, Vol. 28, No. 10, pp. 1106–1121, 2015.
- [31] J. Dossey, A. Otto, L. Spence, and C. Eynden, *Discrete Mathematics*, 2nd Edition, Harpercollins College Div, New York City, New York, 1993.
- [32] T. K. Dao, T. S. Pan, T. T. Nguyen, and J. S. Pan, "Parallel bat algorithm for optimizing makespan in job shop scheduling problems", *Journal of Intelligent Manufacturing*, Vol. 29, No. 2, pp. 451–462, 2018.
- [27] A. Syarif, Y. S. Yun, and M. Gen, "Study on multi stage logistic chain network: A spanning tree-based genetic algorithm approach", *Computers and Industrial Engineering*, Vol. 42, No. 1–2, pp. 299–314, 2002.
- [28] M. Gen and A. Syarif, "Double Spanning Tree-Based Genetic Algorithm for Two-Stage Transportation Problem", *International Journal of Knowledge-Based and Intelligent Engineering Systems*, Vol. 7, No. 4, pp. 214–221, 2003.
- [29] A. Syarif, D. Anggraini, K. Muludi, Wamiliana, and M. Gen, "Comparing Various Genetic Algorithm Approaches For Multiple Choice Multi-Dimensional Knapsack Problem (mm-KP)", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 455–462, 2020.

- [30] A. Syarif, W. Wamiliana, P. Lumbanraja, and M. Gen, "Study on genetic algorithm (GA) approaches for solving Flow Shop Scheduling Problem (FSSP)", In: *The 5th International Conference on Science, Technology and Interdisciplinary Research (IC-STAR 2019)*, Bandar Lampung, Indonesia, 2020.
- [31] P. Pongchairerks, "Particle swarm optimization algorithm applied to scheduling problems", *ScienceAsia*, Vol. 35, No. 1, pp. 89–94, 2009.
- [32] F. Zhao, X. Jiang, C. Zhang, and J. Wang, "A chemotaxis enhanced bacterial foraging algorithm and its application in job shop scheduling problem", *International Journal of Computer Integrated Manufacturing*, Vol. 28, No. 10, pp. 1106–1121, 2015.

9

LAMPIRAN HASIL PENGECEKAN PLAGIASI ARTIKEL

IJIES_JSSP_3740_04012021 - Plagialism Check

by Admi Syarif

Submission date: 05-Jan-2021 12:53PM (UTC+1030)

Submission ID: 1483137999

File name: IJIES_JSSP_3740_04012021_revised_Renaldi.pdf (1.07M)

Word count: 6448

Character count: 32389

ORIGINALITY REPORT

10%

SIMILARITY INDEX

7%

INTERNET SOURCES

9%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

- | | | |
|---|--|-----|
| 1 | hbduan.buaa.edu.cn
Internet Source | 3% |
| 2 | A. Udomsakdigool, V. Kachitvichyanukul.
"Multiple colony ant algorithm for job-shop scheduling problem", International Journal of Production Research, 2008
Publication | 2% |
| 3 | Lin Cheng, Qingzhen Zhang, Fei Tao, Kun Ni, Yang Cheng. "A novel search algorithm based on waterweeds reproduction principle for job shop scheduling problem", The International Journal of Advanced Manufacturing Technology, 2015
Publication | 2% |
| 4 | www.hindawi.com
Internet Source | 1% |
| 5 | ir.nctu.edu.tw
Internet Source | <1% |
| 6 | hl-128-171-57-22.library.manoa.hawaii.edu
Internet Source | <1% |

7	Submitted to Universitas Pendidikan Ganesha Student Paper	<1 %
8	www.mdpi.com Internet Source	<1 %
9	Submitted to Amrita Vishwa Vidyapeetham Student Paper	<1 %
10	www.arpnjournals.org Internet Source	<1 %
11	www.cs.uoi.gr Internet Source	<1 %
12	Lecture Notes in Computer Science, 2003. Publication	<1 %
13	Leila Asadzadeh. "A parallel artificial bee colony algorithm for the job shop scheduling problem with a dynamic migration strategy", Computers & Industrial Engineering, 2016 Publication	<1 %
14	Admi Syarif, Mitsuo Gen. "Hybrid Genetic Algorithm for Production/Distribution System in Supply Chain", International Journal of Smart Engineering System Design, 2003 Publication	<1 %
15	scholar.uwindsor.ca Internet Source	<1 %

16

A. V. Sudhakara Reddy, M. Damodar Reddy, M. Satish Kumar Reddy. "Network Reconfiguration of Primary Distribution System Using GWO Algorithm", International Journal of Electrical and Computer Engineering (IJECE), 2017

Publication

<1 %

17

academicjournals.org

Internet Source

<1 %

18

es.scribd.com

Internet Source

<1 %

19

worldwidescience.org

Internet Source

<1 %

20

A Syarif, W Wamiliana, P Lumbanraja, M Gen. "Study on genetic algorithm (GA) approaches for solving Flow Shop Scheduling Problem (FSSP)", IOP Conference Series: Materials Science and Engineering, 2020

Publication

<1 %

21

Goncalves, J.F.. "A hybrid genetic algorithm for the job shop scheduling problem", European Journal of Operational Research, 20051116

Publication

<1 %

22

Lecture Notes in Computer Science, 2010.

Publication

<1 %

10

**REVIEW KEDUA
DARI REVIEWER
JURNAL**

Review Form

International Journal of Intelligent Engineering and Systems (IJIES)

Paper ID	Ijies
Paper Title	Performance Evaluation of Various Heuristic Algorithms to Solve Job Shop Scheduling Problem (JSSP)

Recommendation for Publication			
<input type="checkbox"/> (Evaluation A:) Accept <input type="checkbox"/> (Evaluation C:) Accept after Major Revision	<input type="checkbox"/> (Evaluation B:) Accept after Minor Revision <input type="checkbox"/> (Evaluation D:) Reject		
<p>Reply to the editor</p> <p>Thank you for your interest and support to IJIES. After the following small revisions, we'd like to accept your paper.</p> <p style="text-align: center;">Thank you very much for your kind email. We really appreciate your comments. We have made revisions on the paper as follows:</p> <p>1. Do you have a permission of use for Fig. 1? If "No", we cannot publish your paper. Otherwise, if "Yes", please send a confirmation.</p> <p style="text-align: center;">We have excluded the Fig. 1 from the paper.</p> <p>2. In Sect. 4, please provide the reference number for the article compared to the proposed method.</p> <p style="text-align: center;">The manuscript has been modified following this comment.</p> <p>Again thank you very much.</p> <p>Best regards,</p> <p>Dr. Eng. Admi Syarif</p>			
Evaluation of Paper			
Contents	Innovation	<input type="checkbox"/> Highly Innovate <input type="checkbox"/> Sufficiently Innovate <input type="checkbox"/> Slightly Innovate <input type="checkbox"/> Not Novel	
	Integrity	<input type="checkbox"/> Poor <input type="checkbox"/> Fair <input type="checkbox"/> Good <input type="checkbox"/> Outstanding	
	Presentation	<input type="checkbox"/> Totally Accessible <input type="checkbox"/> Mostly Accessible <input type="checkbox"/> Partially Accessible <input type="checkbox"/> Inaccessible	

Review Form

	Technical depth	<input type="checkbox"/> Superficial <input type="checkbox"/> Suitable for the non-specialist <input type="checkbox"/> Appropriate for the generally knowledgeable individual working in the field <input type="checkbox"/> Suitable only for an expert
Presentation & English	<input type="checkbox"/> Satisfactory <input type="checkbox"/> Needs improvement <input type="checkbox"/> Poor	
Overall organization	<input type="checkbox"/> Satisfactory <input type="checkbox"/> Could be improved <input type="checkbox"/> Poor	

INASS

11

**EMAIL DARI
EDITOR TERKAIT
ACCEPTANCE
LETTER ARTIKEL**



ADMI SYARIF <admi.syarif@fmipa.unila.ac.id>

ijies3740: Acceptance letter

1 message

EGUCHI Kei <eguti@fit.ac.jp>

Wed, Jan 13, 2021 at 8:49 AM

To: ADMI SYARIF <admi.syarif@fmipa.unila.ac.id>

Dear Author(s),

Paper ID: ijies3740

It is our great pleasure to inform you that the contribution referenced above, for which you are listed as the corresponding author, has been accepted for the IJES journal.

Congratulations!

The camera-ready version of your paper will be sent to you within a few weeks.

Best regards,
IJES Editors.

From: ADMI SYARIF <admi.syarif@fmipa.unila.ac.id>

Sent: Wednesday, January 13, 2021 10:06 AM

To: 江口 啓 <eguti@fit.ac.jp>; ADMI SYARIF <admi.syarif@fmipa.unila.ac.id>

Subject: Invoice and Copyright #3740

Dear Prof. Eguchi (Editor in chief IJES)

Thank you very much for your great work managing IJES.


We have filled out the copyright document and paid the invoice. as attached in this email.


Again thank you very much and look forward to hearing from you.

Best regards

Dr. Eng. Admi Syarif
Lampung University

2 attachments

 **(ijies3740) Acceptance Letter.pdf**
56K

 **(ijies3740) Receipt.pdf**
56K

12

**LAMPIRAN
ACCEPTANCE
LETTER**



Intelligent Networks and Systems Society

Acceptance Letter

International Journal of Intelligent Engineering and Systems (IJIES)

January 13, 2021

Dear Admi Syarif,

Manuscript Title: Performance Evaluation of Various Heuristic Algorithms To Solve Job Shop Scheduling Problem (JSSP)

Author(s): Admi Syarif, Ade Pamungkas, Renaldi Kumar, Mitsuo Gen

Thank you for submitting your paper to the International Journal of Intelligent Engineering and Systems (IJIES). Based on double blind review process, we are pleased to inform you that our Review Committee has accepted your paper.

The paper will be included in the IJIES, which will be published with ISSN (ISSN: 2185-3118) in online on the website (<http://www.inass.org/publications.html>). We are looking forward to your further contribution to our journal.

Kind regards

Prof. Dr. Kei EGUCHI
Editor-in-Chief, International Journal of Intelligent Engineering and Systems

Department of Information Electronics
Fukuoka Institute of Technology

E-mail: ijies@inass.org

13

**LAMPIRAN
TAGIHAN
PEMBAYARAN
PUBLIKASI
ARTIKEL**



Intelligent Networks and Systems Society

Official Receipt

International Journal of Intelligent Engineering and Systems (IJIES)

January 13 2021

Received from

Admi Syarif

Department of Computer Science, Faculty of Mathematics and Sciences, Lampung

Description	Unit	Amount
IJIES formatting fee	1	USD 300
Total		USD 300

Prof. Dr. Kei EGUCHI

President, International Journal of Intelligent Engineering and Systems

Department of Information Electronics
Fukuoka Institute of Technology

Kei Eguchi

E-mail: ijies@inass.org

14

**COPYRIGHT
STATEMENT
PENULIS**

Intelligent Networks and Systems Society (INASS)
COPYRIGHT TRANSFER FORM
<http://www.inass.org>

Please complete and sign this form and send it back to us with the final version of your manuscript (PDF & DOCx). It is required to obtain a written confirmation from authors in order to acquire copyrights for papers published in IJIES (International Journal of Intelligent Engineering and Systems).

Title of a corresponding author (Prof./Assoc.Prof./Assis.Prof./Dr./Mr./Ms.)	Dr. Eng.
Full Name and Surname	ADMI SYARIF
Paper ID	#3740
Paper Title	Performance Evaluation of Various Heuristic Algorithms To Solve Job Shop Scheduling Problem (JSSP)
Authors	Admi Syarif, Ade Pamungkas, Renaldi Kumar, and Mitsuo Gen
Organisation	Department of Computer Science, Faculty of Mathematics and Sciences, Lampung University
Address	Jl. S. Brodjonegoro No.1
Postal code	35145
City	Bandar Lampung
Country	Indonesia
Telephone	+62-811-722-666
E-mail	admi.syarif@fmipa.unila.ac.id

Copyright Transfer Statement

The copyright to this article is transferred to the Intelligent Networks and Systems Society (INASS) if and when the article is accepted for publication. The undersigned hereby transfers any and all rights in and to the paper including without limitation all copyrights to the Intelligent Networks and Systems Society (INASS). The undersigned hereby represents and warrants that the paper is original and that he/she is the author of the paper, except for material that is clearly identified as to its original source, with permission notices from the copyright owners where required. The undersigned represents that he/she has the power and authority to make and execute this assignment.

We declare that:

1. This paper has not been published in the same form elsewhere.
2. It will not be submitted anywhere else for publication prior to acceptance/rejection by this Conference/Journal.
3. A copyright permission is obtained for materials published elsewhere and which require this permission for reproduction.

The copyright transfer covers the exclusive right to reproduce and distribute the article, including reprints, translations, photographic reproductions, microform, electronic form (offline, online) or any other reproductions of similar nature. The corresponding author signs for and accepts responsibility for releasing this material on behalf of any and all co-authors. This agreement is to be signed by at least one of the authors who has obtained the assent of the co-author(s) where applicable. After submission of this agreement signed by the corresponding author, changes of authorship or in the order of the authors listed will not be accepted.

Yours Sincerely,

Corresponding Authors's Full name & Signature: **ADMI SYARIF**

Date: **13/01/2021**

**** Please Compile This Form, Sign and Send By E-mail : ijies@inass.org**

15

**INVOICE
PEMBAYARAN
PUBLIKASI
ARTIKEL**

FAKTUR

Intelligent Networks and Systems Society

啓 江口

eguti@fit.ac.jp

Lunas

Nomor Faktur: 0862
Tanggal Faktur: 12 Jan 2021
Tanggal jatuh tempo: 11 Feb 2021

Jumlah jatuh tempo:
\$0,00



Pindai. Bayar.
Mulai

Tagih ke:

admi.syarif@fmipa.unila.ac.id

Uraian	Kuantitas	Harga	Jumlah
Publication fee: Intelligent Networks and Systems Society (INASS)	1	\$300,00	\$300,00
Publication fee for the International Journal of Intelligent Engineering and Systems (IJIES)			
Subtotal			\$300,00
Diskon (0%)			\$0,00
Total			\$300,00
Jumlah yang dibayar			-\$300,00
Jumlah jatuh tempo			\$0,00 USD

Catatan

Dear authors,

Thank you for your interest and support to IJIES.
Your final manuscript is ready for the publication in IJIES.
We kindly request you to pay your publication fee.

After you finished your payment, please send the payment proof to ijies@inass.org.
We'll send the official receipt and your acceptance letter later.

P.S.
Currently, the IJIES is indexed by SCOPUS.
(see <https://www.scopus.com/sourceid/21100199790?origin=sbrowse>)
However, please bear in mind that SCOPUS suddenly drop journals from the list. No refund will be given even if the IJIES is dropped from SCOPUS list.

Best regards,
IJIES Editor.

16

**EMAIL DARI
EDITOR TERKAIT
ARTIKEL YANG
SIAP
DIPUBLIKASIKAN**



ADMI SYARIF <admi.syarif@fmipa.unila.ac.id>

ijies3740: camera-ready

3 messages

EGUCHI Kei <eguti@fit.ac.jp>

Wed, Jan 20, 2021 at 6:37 AM

To: ADMI SYARIF <admi.syarif@fmipa.unila.ac.id>

Dear author(s),

Thank you for your interest and support to IJIES.

I'd like to send the camera-ready version of your final manuscript.

(Note: To prevent illegal revision, the MS-Word file of the camera-ready version is not provided.)

Please check and confirm the attached file, and send the check result within ONE week.

*** Please indicate revised points clearly by coloring fonts.

Note:

The camera-ready process is only one time.

To avoid illegal revision, only small revision is permitted.

If you have any question, please contact us with your paper ID.

Best regards,
IJIES Editors

-----Original Message-----

From: 江口 啓

Sent: Wednesday, January 13, 2021 10:50 AM

To: ADMI SYARIF <admi.syarif@fmipa.unila.ac.id>

Subject: ijies3740: Acceptance letter

Dear Author(s),

Paper ID: ijies3740

It is our great pleasure to inform you that the contribution referenced above, for which you are listed as the corresponding author, has been accepted for the IJIES journal.

Congratulations!

The camera-ready version of your paper will be sent to you within a few weeks.

Best regards,
IJIES Editors.

From: ADMI SYARIF <admi.syarif@fmipa.unila.ac.id>

Sent: Wednesday, January 13, 2021 10:06 AM

To: 江口 啓 <eguti@fit.ac.jp>; ADMI SYARIF <admi.syarif@fmipa.unila.ac.id>


Subject: Invoice and Copyright #3740

Dear Prof. Eguchi (Editor in chief IJIES) Thank you very much for your great work managing IJIES. We have filled out the copyright document and paid the invoice. as attached in this email.

Again thank you very much and look forward to hearing from you.

Best regards

Dr. Eng. Admi Syarif
Lampung University

 **2021043030.pdf**
567K

ADMI SYARIF <admi.syarif@fmipa.unila.ac.id>
To: EGUCHI Kei <eguti@fit.ac.jp>

Wed, Jan 20, 2021 at 11:49 AM


Dear Prof. eguchi

Thank you very much for your kind email. We have checked the camera-ready version of our paper (#3740) carefully. Unfortunately, we have few revisions of our paper as marked in the attached file. We greatly appreciate your great work managing this high quality journal.
Again thank you very much and look forward to hearing any information from you.

Sincerely yours

Dr. Eng. Admi Syarif

[Quoted text hidden]

 **2021043030_Revision_Camera_Ready_3740.pdf**
676K

EGUCHI Kei <eguti@fit.ac.jp>
To: ADMI SYARIF <admi.syarif@fmipa.unila.ac.id>

Wed, Jan 20, 2021 at 12:22 PM

Dear Author(s),

Thank you for your confirmation.
Please see the attached file.

You completed all processes.
On 28/2/2021 (tentative), your paper will appear on the IJIES homepage.
Appreciate your patiently wait.

Congratulations!

Best regards,
IJIES Editors.

From: ADMI SYARIF <admi.syarif@fmipa.unila.ac.id>
Sent: Wednesday, January 20, 2021 1:49 PM
To: 江口 啓 <eguti@fit.ac.jp>
Subject: Re: ijies3740: camera-ready

Dear Prof. eguchi

Thank you very much for your kind email. We have checked the camera-ready version of our paper (#3740) carefully. Unfortunately, we have few revisions of our paper as marked in the attached file. We greatly appreciate your great work managing this high quality journal.
Again thank you very much and look forward to hearing any information from you.

Sincerely yours

Dr. Eng. Admi Syarif

[Quoted text hidden]

[Quoted text hidden]



2021043030.pdf

567K