Research Article/Araştırma Makalesi

GAZİ
JOURNAL OF ENGINEERING SCIENCES

Advanced Tree-Seed Algorithm for Large Sized Job Shop Scheduling Problems

Mehmet Akif Şahman[*a]
Submitted: 28.02.2022 Revised: 28.03.2022 Accepted: 12.04.2022 doi:10.30855/gmbd.0705004

Keywords: Tree-Seed Algorithm
Job Shop Scheduling Problems
Meta-Heuristic Algorithms
Mutation Operators

[a,*] Selcuk University,
Technology Faculty,
Dept. of Electrical and Electronics Eng.
42075 - Konya, Türkiye
Orcid: 0000-0002-1718-3777
e mail: asahman@selcuk.edu.tr

[*]Corresponding author:
asahman@selcuk.edu.tr

ABSTRACT

Globalizing economies force manufacturing companies to develop themselves and take new measures. Correct planning of the production process is vital in order to increase the competitiveness of companies. For this reason, the Job shop Scheduling Problem (JSP) has a great role in the planning of production in companies. In JSPs, the determined jobs in the manufacturing companies must be run in the correct order on existing and suitable machines. Planning to complete the work in the company in the shortest possible time on the smachines and in the order, is a combinatorial difficult optimization problem. Both exact methods and meta-heuristic approaches are used in solving JSPs, which is an NP-hard optimization problem. Depending on the number of jobs and machines in the companies, the size of this optimization problem changes. In JSPs where the number of jobs and machines is high, precise methods may be insufficient to produce solutions in operational time. Therefore, meta-heuristic algorithms are frequently used in solving JSPs. In this work, the exploration and exploitation capabilities in the Tree Seed Algorithm (TSA) are enhanced with the swap, symmetry, and shift mutation operators. The proposed new TSA (Advanced TSA-ATSA) algorithm is compared on large-sized JSPs with meta-heuristic algorithms which are well known in the literature. The proposed ATSA has shown promising performance in experimental studies.

Büyük Boyutlu Atölye Tipi Çizelgeleme Problemleri için Gelişmiş Ağaç-Tohum Algoritması

ÖZ

Küreselleşen ekonomiler, imalatçı firmaların kendilerini geliştirmeye ve yeni önlemler almaya zorlamaktadır. Üretim sürecinin doğru bir şekilde planlanması firmaların rekabet edebilme gücünü arttırmak için hayatidir. Bu nedenle firmalardaki üretimin planlanmasında Atölye tipi Çizelgeleme Probleminin (AÇP) büyük rolü vardır. AÇP'de, üretici firmalardaki belirlenen işlerin mevcut ve uygun makinelerde doğru sırada çalıştırılması gerekmektedir. Firmadaki işlerin belirlenen makinelerde ve sırada mümkün olan en kısa sürede tamamlanması için hazırlanan planlama ise kombinatoryal zorlu bir optimizasyon problemidir. NP-Zor bir optimizasyon problemi olan AÇP'lerinin çözümünde hem kesin metotlar hem de meta-sezgisel yaklaşımlar kullanılmaktadır. Firmalardaki iş ve makine sayılarına bağlı olarak, bu optimizasyon probleminin boyutu değişmektedir. Yüksek sayısaki iş ve makine sayısına sahip AÇP'lerde, kesin metotlar operasyonel zamanda çözüm üretmekte yetersiz kalmaktadır. Bu nedenle, AÇP'lerinin çözümünde meta-sezgisel algoritmalar sıklıkla kullanılmaktadır. Bu çalışmada, Ağaç Tohum Algoritması'ndaki (ATA) keşif ve sömürü yetenekleri, takas, simetri ve kaydırma mutasyon operatörleri ile geliştirilmiştir. Önerilen yeni ATA (Gelişmiş ATA-GATA) algoritması, büyük boyutlu AÇP'ler literatürde iyi bilinen meta-sezgisel algoritmalarla karşılaştırılmıştır. Önerilen GATA, deneysel çalışmalarda umut verici bir performans göstermiştir.

Anahtar Kelimeler: Ağaç-Tohum Algoritması
Atölye Tipi Çizelgeleme Problemi
Meta-Sezgisel Algoritmalar
Mutasyon Operatörleri

To cite this article: M. A. Şahman, "Advanced Tree-Seed Algorithm for Large Sized Job Shop Scheduling Problems," *Gazi Journal of Engineering Sciences,* vol. 8, no. 2, pp. 201-214, 2022. doi:10.30855/gmbd.0705004

1. Introduction

In recent years, there has been a compelling development in production manufacturing due to reasons such as technological innovations in the industrial field, increase in demand, and commercial globalization. Manufacturing enterprises must be able to contend with problems such as on-time delivery, rapid response to product changes, and multiple product options to survive in the marketplace [1]. Therefore, enterprises require transparency over production scheduling. This is a process that can only be achieved with realistic and effective scheduling. Due to the realities of the current situation, the Job shop Scheduling Problem (JSP) which is the core of job planning in manufacturing is of great practical importance [2].

Due to the discrete and combinational nature of JSP, is an NP-HARD optimization problem [3]. Exact deterministic methods that produce precise results are suggested for solving NP-HARD problems such as JSP. However, even if these methods benefit from today's technological developments, they cannot produce results in a reasonable time in solving large-sized discrete problems [4]. Even a small increase in the size of an NP-Hard optimization problem such as JSP provides an exponential increase in computation time. For this reason, meta-heuristic algorithms that can reach the near-optimum in a reasonable computation time are used, even if they do not guarantee the optimum.

Meta-heuristic algorithms are artificial intelligence-based approaches that are generally inspired by intelligent behaviours in nature and model these behaviours. One of the best known of these meta-heuristic algorithms is the particle swarm optimization algorithm proposed by Kennedy and Eberhart [5]. The population-based PSO algorithm is inspired by the foraging strategies of fish and bird flocks in nature. It has been preferred in many optimization studies because it is easily adaptable and has few design parameters [6-9]. Inspired by the nectar and pollen search approach of honey bees in nature, the Artificial Bee Colony (ABC) algorithm proposed by Karaboga is one of the well-known meta-heuristic algorithms [10]. The ABC algorithm is preferred by researchers because of its success in local and global search and its ability to resist problems such as local optima [11-15]. The grey wolf optimizer (GWO) proposed by Mirjalili et al. [16], especially due to its hierarchy in social status and modeling of its hunting strategies, has been used by researchers in solving optimization problems in recent years [17-19]. Inspired by the random distribution and growth of trees and their seeds in nature, the Tree-Seed Algorithm (TSA) that is population based was proposed by Kiran for the solution of continuous optimization problems [20]. TSA is a preferred meta-heuristic algorithm in optimization problems due to its exploration and exploitation capacity, its ability to converge to the optimum, and its avoidance of computational complexity.

Different meta-heuristic algorithms have been studied to improve the solution quality of the JSP problem. Song et al. [21] used the PSO and Simulated Annealing algorithm(SA) as a hybrid (HPSOSA) to solve JSP problems. Pongchairerks and Kachitvichyanukul [22] proposed a new approach for JSPs by hybridizing the variable neighborhood search algorithm with the PSO algorithm. In their other studies, Pongchairrks and Kachitvichyanukul obtained more effective results by suggesting the combination of parameter tunning PSO (PT-PSO) and JSP-PSO algorithms suggested for JSPs [23]. Sha and Hsu [24] adapted the update processes of the continuous PSO for discrete optimization problems and produced effective results for large-sized JSPs by making use of the tabu search algorithm. Du and Liu [25] proposed the Improved ABC (IABC) algorithm by adding together the mutation operator to the ABC algorithm. Jiang and Zhang [26] have studied JSPs and flexible JSPs to demonstrate the success of the GWO algorithm in solving combinatorial problems. Jiang [27] proposed a new discretization approach to solve JSPs using the GWO algorithm working in the continuous search space, increased the diversity of the initial population, and hybridized the variable neighbourhood search algorithm with GWO and used it in solving JSPs. There are other meta-heuristic algorithms used in the literature for JSPs [28-30].

According to the literature search, the performance of the TSA algorithm in both small-sized and large-sized JSPs has not been carried out before. Apart from that, due to the structure of the TSA algorithm, since the objective function is run more than once during the seed production process, an experimental study based on iteration would not be fair, so in this study, the results are discussed according to the number of function evaluations (FEs). In the seed generation process in the TSA approach, a new approach called Advanced TSA (ATSA) has been proposed by including swap, symmetry, and shift mutation operators. By testing JSPs on well-known and frequently used meta-heuristic (PSO, ABC, and GWO) algorithms in the literature, comparisons were made with the ATSA proposed in this study, on solution quality and solution time.

The remainder of the study is organized as follows. Problem definition and encoding scheme are given in Section 2. TSA algorithm and new proposed ATSA approach are given Section 3 and 4, respectively. Section 5 gives the experimental studies. Conclusion and future works are presented in section 6.

2. Problem Definition and Encoding Scheme

JSP is known as NP-Hard, which has n jobs and m machines to process these jobs and is a difficult optimization problem to solve. In JSP, the number of probable solutions can be calculated as $n!^m$ depending on the number of jobs (n) and machines (m). Thus, a very small change in the number of jobs or machines will cause an exponential increase in the number of possible solutions. There are some presuppositions stated below in JSP. In JSP problems, there are n sets of jobs $J = \{J_1, J_2, J_3, \dots, J_n\}$ and m sets of machines $M = \{M_1, M_2, M_3, \dots, M_m\}$. Each job has a fixed set of operations $O_{ij} = \{O_{i1}, O_{i2}, O_{i3}, \dots, O_{im}\}$ $(i = 1,2, \dots, n\,;\ j = 1,2, \dots, m)$ on which certain machine it will work in which order and for which time. It is assumed that the processes of a job can run on only one machine at a time in the specified order, the specified machine order is not able to change, and there will be no interruptions or waiting during the process. The total number of operations in JSP is $n\ x\ m$. Each operation has makespan described as $C_{ij} = \{C_{i1}, C_{i2}, C_{i3}, \dots, C_{im}\}$ $(i = 1,2, \dots, n\,;\quad j = 1,2, \dots, m)$. As shown in Eq.1 below, the makespan (C_{max}) which is taken as the objective function is the total unit time between the completion of the first operation and the completion of the last operation [26].

$$minC_{max} = \min \max C_{im}\ \ (i = 1,2, \dots, n\,;\ j = 1,2, \dots, m) \tag{1}$$

Meta-heuristic algorithms are often proposed to solve continuous optimization problems. However, JSPs is a discrete optimization problem. Therefore, there is a need for an encoding scheme that transforms the meta-heuristic algorithms used in this study from continuous search space to discrete search space. Bean [31] suggested the random-key encoding scheme (RK) that is often used in JSPs. The following Eq.2 is used to convert the continuous values to the order of operations.

$$\vec{S} = (\ \tau_l\ mod\ n) + 1, l = 1,2, \dots, n\ x\ m \tag{2}$$

Where, $\vec{S}$ is the vector showing the implementation order of the operations of the jobs, τ_l is integer sequence of the operation of the lth job, n is the number of jobs and m is the number of machines. Let us suppose, we have two jobs and three machines as shown in Table 1.

Table 1. JSP instance

Job	**Machine**			**Processing Order**
	1	**2**	**3**	
1	8	6	3	{1,3,2}
2	4	10	8	{2,1,3}

For the 2-job and 3-machine JSP given in Table 1, the continuous data of {-2.1, 4.5, 4.9, 1.2, -4.3, 3.8} in the range of -5 to 5 are randomly generated respectively. First of all, these continuous data are ordered from smallest to largest, and the sequence numbers are placed in the position where they are located, like {2,5,6,3,1,4}. Then this ordering is converted into the order of operations of things as {1,2,1,2,2,1} using Eq.2. This sequencing shows the order of operations of the jobs in the JSP problem as {$O_{1,1}$, $O_{2,1}$, $O_{2,1}$, $O_{2,2}$, $O_{2,3}$, $O_{1,3}$} (Table 2). For example, $O_{2,3}$ indicates that the 2nd job here will be processed on the 3rd machine.

Table 2. Example of the using RK encoding scheme

Real Values	-2.1	4.5	4.9	1.2	-4.3	3.8
Integer Series (φ)	2	5	6	3	1	4
Job Indexes	1	2	1	2	2	1
Operation Sequence	$O_{1,1}$	$O_{2,1}$	$O_{1,2}$	$O_{2,2}$	$O_{2,3}$	$O_{1,3}$

When the operations specified in Table 2 are run according to the given order, the Gantt chart becomes as in Figure 1 below. As can be seen in the Gantt chart, the last completed operation is the 3rd operation of the 2nd job ($O_{2,3}$) and makespan (C_{max}) is 24 units of time.

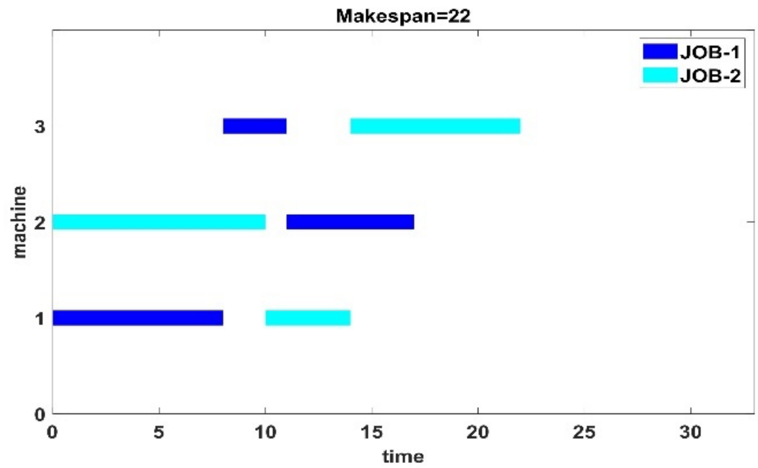

Figure 1. Gantt scheme of the example

3. Tree-Seed Algorithm

The TSA proposed by Kıran [20] has been used to solve continuous optimization problems by modeling the natural process of trees and seeds [32-35]. In population-based TSA, trees disperse their own seeds in random locations [36-38]. Those that grew up from these seeds that spread in nature and are of better quality than the tree from which they spread continue to live.

TSA uses location information of trees and seeds. The area in which they spread in nature corresponds to the search space in optimization problems. The exploration and exploitation process is carried out both by making use of randomly selected neighbor solutions and by using the search tendency (st) parameter in the TSA approach. How many seeds will be produced in the seed production process depends on the NS (number of seed) parameter.

Algorithm 1. *Pseudo code of the TSA for JSP*

```
Define parameters (N, ST,MaxFEs)
Initialize population for N trees randomly, %N is pop. size
Use encoding scheme for each tree (RK)
Evaluate objective function, FEs=N
while FEs < maxFEs
  for i=1 to N    //seed mechanism
    Determine NS, Select r (random neighbor tree)
    for j=1 to NS
      for k=1 to D
        if rand<ST use Eq.3
          Check the min and max boundaries of seed
        else use Eq.4
          Check the min and max boundaries of seed
        end if
      end for
    end for
    Use RK for each seeds
    Evaluate seed, FEs=FEs+NS
    Determine best seed
    if best seed < tree , change tree with best seed
  end for
Store the best solution so far
end while
Output : Best Solution (min obj. func. value)
```

Figure 2. Pseudo code of the TSA algorithm

The NS parameter is randomly determined to be between 10% and 25% of the total tree population. In the TSA algorithm, two equations (Eq.3 and Eq.4) are used in the seed generation process.

$$\overrightarrow{Seed} = \overrightarrow{Tree} + \vec{r}.\left(\overrightarrow{Best} - \overrightarrow{Tree_r}\right) \tag{3}$$

$$\overrightarrow{Seed} = \overrightarrow{Tree} + \vec{r}.\left(\overrightarrow{Tree} - \overrightarrow{Tree_r}\right) \tag{4}$$

where, $\overrightarrow{Seed}$ is the position vector of the seeds to be produced, $\overrightarrow{Tree}$ is the position vector of the tree from which seed will be produced. $\overrightarrow{Best}$ is the position vector of the tree for the best solution, $\overrightarrow{Tree_r}$ is the position vector of the relevant parameters of the trees that will be randomly selected up to the number of parameters. $\vec{r}$ is a vector of randomly selected values in the range of [-1,1]. The st parameter is decisive in the selection of the equation to be preferred in seed production. If the st parameter is less than the randomly selected value in the range of [0,1], Eq.3 is used, otherwise Eq.4 is used. Pseudo code of the TSA is given as follows in Figure 2.

4. Advanced Tree-Seed Algorithm

In the exploration and exploitation process of the TSA approach in the solution space, both the position information of the randomly selected trees and the position information of the best solution is used. In this study, the seed production process of TSA is tried to be improved. In the seed generation process, three different mutation operators are used, except for the position information of the random neighbor and best tree. These mutation operators (swap, symmetry, and shift mutation operators) are presented as follows.

- **Swap operator:** One of the most used mutation operators is the swap operator. The swap operator is used to exchange values at two randomly (r1=3, r2=6) determined positions (Figure 3).

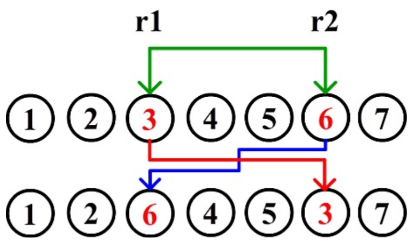

Figure 3. Swap Operation

- **Symmetry operator:** The symmetry operator works with two parameters. The first parameter (r1=4) is the position to get symmetry in the given vector, and the second parameter (r2=2) is how many values will be symmetrical (Figure 4).

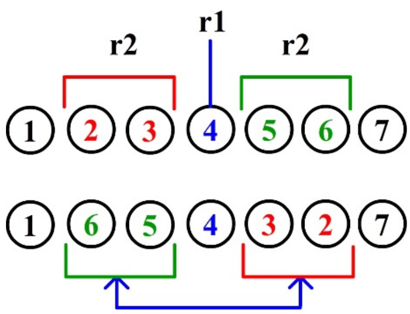

Figure 4. Symmetry Operation

- **Shift operator:** In this mutation operator, the value at the parameter (r1=2) position with determined is placed at the position of the second parameter (r2=5) value. Intermediate values are shifted one left.

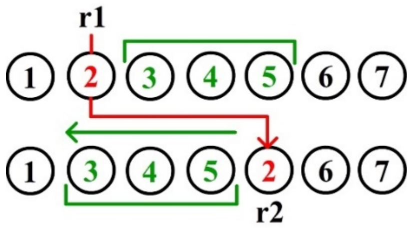

Figure 5. Shift Operation

After the operators (Swap, Symmetry, and Shift) given above are included in the seed generation process of the TSA algorithm, the pseudo-code of the new seed generation process of the ATSA approach proposed in this study is given in the figure below. Except for the seed mechanism of the ATSA approach, the parts are the same as for the TSA. Therefore, these sections are not shown in the pseudocode.

Algorithm 2. *Pseudo code of seed mechanism*

```
while FEs < maxFEs
  for i=1 to N    //seed mechanism
    Determine NS, Select r (random neighbor tree)
    for j=1 to NS
      if rand<0.5
        if rand<ST
          Use swap operator for best tree, use RK
          Evaluate, FEs=FEs+1
        else
          Use Symmetry operator for rth neighbor(Sy)
          Use Shift operator for best tree (Sh)
          Use RK, and Evaluate for Sy and Sh, FEs=FEs+2
          Select the best seed between Sy and Sh
        end if
      else
        if rand<0.75 use Eq.3
          Check the min and max boundaries of seeds
          Use RK and Evaluate, FEs=FEs+1
        else use Eq.4
          Check the min and max boundaries of seed
          Use RK and Evaluate, FEs=FEs+1
        end if
      end if
    end for
    Determine best seed
    if best seed < tree , change tree with best seed
  end for
  Store the best solution so far
end while
Output : Best Solution (min obj. func. value)
```

Figure 6. Pseudo code of seed mechanism

5. Experimental Studies

In this study, 40 benchmark problems proposed by Taillard [39] for different shop scheduling problems have been used. While choosing the benchmark problems, attention has been paid to the fact that they were in groups of five and different sizes.

The size of the smallest benchmark problem is 225 (n=15, m=15) while the size of the largest dimension problem is 2000 (n=100, m=20). Thus, the performance of meta-heuristic algorithms can be evaluated on benchmark problems of different sizes (Table 3)

The parameter values of the metaheuristic algorithms (PSO, TSA, ABC, GWO, and ATSA proposed in this study) used in the study are given in Table 4.

Table 3. Benchmark problem set

Name	Jobs x Machines	Dimension	Name	Jobs xMachines	Dimension
ta01	15×15	225	ta41	30×20	600
ta02	15×15	225	ta42	30×20	600
ta03	15×15	225	ta43	30×20	600
ta04	15×15	225	ta44	30×20	600
ta05	15×15	225	ta45	30×20	600
ta11	20×15	300	ta51	50×15	750
ta12	20×15	300	ta52	50×15	750
ta13	20×15	300	ta53	50×15	750
ta14	20×15	300	ta54	50×15	750
ta15	20×15	300	ta55	50×15	750
ta21	20×20	400	ta61	50×20	1000
ta22	20×20	400	ta62	50×20	1000
ta23	20×20	400	ta63	50×20	1000
ta24	20×20	400	ta64	50×20	1000
ta25	20×20	400	ta65	50×20	1000
ta31	30×15	450	ta71	100×20	2000
ta32	30×15	450	ta72	100×20	2000
ta33	30×15	450	ta73	100×20	2000
ta34	30×15	450	ta74	100×20	2000
ta35	30×15	450	ta75	100×20	2000

Table 4. Peculiar parameters of algorithms

	Algorithms				
Parameters	**PSO**	**TSA**	**ABC**	**GWO**	**ATSA**
Population Size (N)	40	40	40	40	40
Dimension Size (D)	n x m	n x m	n x m	n x m	n x m
dmin ,dmax	-5, 5	-5, 5	-5, 5	-5, 5	-5, 5
Max_FEs	D*1000	D*1000	D*1000	D*1000	D*1000
C1	2	NA	NA	NA	NA
C2	2	NA	NA	NA	NA
maxLimit	NA	NA	40	NA	NA
L	NA	N*0.1	NA	NA	N*0.1
U	NA	N*0.25	NA	NA	N*0.25
ST	NA	0.2	NA	NA	0.2
$\vec{a}$	NA	NA	NA	2 to 0	NA

PSO, TSA, ABC, and GWO algorithms have been re-coded in order to carry out experimental studies. These algorithms are meta-heuristic algorithms suggested to work in continuous search space and to solve continuous problems. In order to use these algorithms in discrete optimization problems (JSP), RK encoding scheme is used. Algorithms evaluated in experimental studies were run 20 times under the same conditions and the results were saved. For each benchmark problem, the mean (Mean), standard deviation (Std), median value (Med), best-worst solution(min-max) (Table 5), and elapsed time (Table 6) of the results of 20 runs were presented.

The environment in which the experimental studies are carried out on notebook PC with an Intel Core i7 machine, CPU 2.6 GHz, 16 GB of RAM, Windows 10 (64) system, and MATLAB® software. In Table 5, the average of the results obtained from all benchmark problems of the relevant algorithms are given in the AVG line, and the number of winners in the comparison of the statistical values of the algorithms are given as Winner/Total (W/T).

When the results of 20 studies of algorithms are examined in Table 5, it is seen that the ATSA approach proposed in this study outperforms the average, median, minimum, and maximum values for all benchmarking problems. According to the average results of the comparison problems, the closest result to the ATSA approach was obtained from the PSO algorithm, and the result obtained from the PSO algorithm was improved by 12.04% with ATSA. The worst result was obtained from the TSA algorithm, and the result was improved by 30.33% compared between the ATSA and TSA algorithms. According to these results, the ATSA proposed in the study performed better than the original TSA algorithm.

Table 5. Obtained results of the algorithms

	PSO				TSA				ABC				GWO				ATSA			
Problem	**Mean**	**Med**	**Min**	**Max**	**Mean**	**Med**	**Min**	**Max**	**Mean**	**Med**	**Min**	**Max**	**Mean**	**Med**	**Min**	**Max**	**Mean**	**Med**	**Min**	**Max**
ta01	1519.20	1518.5	1468	1569	1724.55	1725.5	1651	1776	1680.25	1682	1636	1749	1732.00	1742.5	1641	1769	1444.8	1445	1347	1517
ta02	1540.25	1547.5	1465	1601	1725.60	1736.5	1645	1772	1679.70	1679	1623	1763	1745.10	1744	1712	1786	1435	1421	1374	1543
ta03	1553.05	1544.5	1494	1626	1736.70	1737.5	1709	1766	1670.40	1671	1591	1731	1714.10	1728	1623	1764	1430.6	1433.5	1353	1504
ta04	1630.10	1631	1567	1705	1716.35	1724.5	1651	1747	1659.50	1667.5	1574	1711	1713.30	1722	1634	1754	1438.1	1432.5	1366	1600
ta05	1589.55	1593.5	1531	1644	1736.60	1738.5	1672	1781	1694.25	1708.5	1620	1759	1734.55	1739.5	1624	1798	1448.5	1448	1344	1535
ta11	1880.10	1875.5	1778	1950	2091.70	2098	2017	2136	2035.00	2036	1971	2115	2083.50	2091	1996	2134	1663.7	1661.5	1581	1731
ta12	1870.35	1864	1831	1932	2104.95	2105.5	2009	2169	2065.20	2082	1974	2125	2113.95	2106.5	2055	2177	1685.6	1678.5	1566	1851
ta13	1871.50	1864.5	1798	1947	2060.50	2069	1986	2101	2005.10	2010.5	1951	2068	2063.15	2069.5	1991	2108	1638.8	1647.5	1517	1756
ta14	1768.55	1765.5	1718	1827	2000.05	1999	1918	2044	1948.20	1944.5	1895	2012	1997.80	1998.5	1928	2040	1589.9	1579.5	1518	1719
ta15	1843.10	1833.5	1769	1989	2079.90	2080	1969	2127	2020.65	2026	1955	2070	2072.85	2083	1990	2118	1685	1676	1607	1792
ta21	2156.30	2173.5	2091	2212	2553.15	2550	2507	2601	2482.15	2490	2403	2548	2534.15	2530	2492	2572	1990.6	1985.5	1919	2109
ta22	2162.05	2155.5	2072	2266	2526.95	2527.5	2473	2565	2434.35	2447.5	2313	2497	2521.75	2521	2470	2570	1944.3	1924	1843	2050
ta23	2083.30	2076.5	2027	2157	2476.85	2481	2422	2511	2389.20	2391	2280	2472	2466.15	2471	2391	2496	1902.1	1906	1791	2012
ta24	2142.80	2153	2074	2205	2520.80	2524.5	2457	2566	2461.25	2465.5	2392	2508	2515.55	2509.5	2464	2577	1994.2	2005.5	1872	2124
ta25	2139.65	2144.5	2081	2178	2477.75	2484	2406	2523	2430.45	2437	2342	2488	2485.70	2492.5	2435	2521	1934.3	1935	1825	2090
ta31	2365.10	2359.5	2273	2491	2794.75	2801.5	2696	2854	2705.80	2712	2584	2788	2792.70	2806	2683	2846	2152.9	2151.5	2050	2255
ta32	2567.65	2592.5	2411	2665	2908.25	2926.5	2785	2961	2833.95	2849	2725	2903	2918.00	2926	2854	2974	2242.8	2245.5	2153	2337
ta33	2490.90	2491	2412	2590	2916.10	2920	2873	2951	2843.25	2848.5	2783	2899	2909.45	2910	2853	2957	2250.9	2238.5	2154	2352
ta34	2473.95	2483.5	2344	2581	2868.40	2868	2803	2916	2776.10	2783	2698	2842	2830.30	2843.5	2727	2897	2234.5	2219.5	2171	2373
ta35	2532.15	2526.5	2449	2633	2867.95	2878.5	2785	2910	2800.40	2813	2740	2858	2872.90	2882	2786	2930	2351.7	2347.5	2214	2562
ta41	2841.20	2822.5	2741	2966	3402.60	3412	3304	3477	3327.20	3335	3195	3390	3405.40	3407.5	3338	3443	2571.9	2587	2395	2657
ta42	2868.70	2863	2758	2985	3317.20	3323	3239	3373	3233.65	3234.5	3144	3352	3343.05	3341	3281	3387	2520.4	2504	2361	2689
ta43	2801.60	2783	2707	2919	3260.90	3258.5	3216	3302	3191.45	3193.5	3041	3263	3262.90	3263	3150	3332	2424.7	2404.5	2312	2717
ta44	2891.15	2889.5	2778	3006	3392.90	3392.5	3345	3445	3304.70	3310.5	3207	3376	3394.20	3398.5	3330	3441	2544.5	2532.5	2382	2678
ta45	2857.00	2846.5	2772	2966	3356.35	3358	3303	3404	3275.20	3281.5	3127	3354	3345.20	3356	3183	3417	2484.9	2482.5	2366	2576
ta51	3793.75	3761.5	3703	3996	4288.05	4298.5	4222	4340	4186.00	4201	3992	4251	4294.65	4302	4186	4374	3337.2	3331	3243	3524
ta52	3741.85	3716.5	3613	3896	4221.90	4234	4100	4279	4140.35	4157	4036	4192	4224.55	4215.5	4159	4292	3371.4	3360	3235	3563
ta53	3410.65	3371	3285	3632	4051.95	4055	4000	4111	3937.85	3954	3815	3999	4048.25	4059	3944	4109	3136.2	3122.5	3023	3290
ta54	3475.65	3440.5	3381	3689	4088.15	4084.5	4022	4144	4001.45	4004	3905	4090	4095.55	4096.5	4038	4162	3206.9	3206	3056	3308
ta55	3719.10	3698.5	3573	3868	4187.90	4196.5	4036	4251	4106.05	4114	3997	4172	4185.60	4205.5	4049	4263	3261.3	3264	3145	3418
ta61	4064.60	4061	3910	4220	4801.55	4807.5	4735	4861	4683.70	4695	4530	4796	4799.65	4802.5	4711	4856	3521.7	3527	3385	3684
ta62	4150.40	4126	4039	4385	4899.75	4899	4836	4941	4814.30	4811.5	4716	4925	4904.10	4913.5	4734	4974	3548	3563.5	3365	3625
ta63	3818.20	3802	3653	4103	4604.80	4607.5	4508	4667	4502.65	4510.5	4394	4585	4567.15	4578.5	4369	4640	3322.3	3321.5	3212	3530
ta64	3718.65	3722.5	3600	3817	4519.65	4528.5	4418	4615	4420.40	4407.5	4303	4539	4542.15	4551	4464	4627	3333.8	3342.5	3239	3523
ta65	3852.25	3855	3732	3966	4666.30	4665	4609	4706	4577.05	4585	4475	4649	4662.15	4659.5	4567	4757	3371.9	3367	3277	3548
ta71	7187.45	7189.5	6915	7345	8371.05	8389.5	8209	8424	8260.55	8280	8034	8363	8347.00	8365	8188	8456	6441	6448.5	6255	6614
ta72	6768.15	6781	6396	6966	7984.40	7990	7878	8053	7829.20	7825.5	7698	7919	7986.30	8000.5	7853	8062	5910.9	5918	5680	6052
ta73	7112.05	7118	6905	7323	8326.20	8335	8227	8405	8196.85	8200	7942	8391	8303.25	8317.5	8188	8405	6527	6521	6324	6792
ta74	6689.95	6730	6290	6950	8076.15	8093.5	7949	8172	7982.35	8005.5	7822	8071	8108.80	8107.5	7997	8223	5987.3	6013	5797	6108
ta75	7308.95	7306.5	7019	7533	8281.00	8281.5	8165	8371	8117.60	8137	7978	8212	8288.60	8300	8181	8343	6293.1	6282	6056	6592
AVG	3181.27	3176.95	3060.58	3307.48	3699.67	3704.63	3618.88	3752.95	3617.59	3624.65	3510.03	3694.88	3698.14	3703.90	3606.48	3758.78	**2839.35**	**2836.99**	**2716.83**	**2982.50**
W/T	0/48	0/48	0/48	0/48	0/48	0/48	0/48	0/48	0/48	0/48	0/48	0/48	0/48	0/48	0/48	0/48	**48/48**	**48/48**	**48/48**	**48/48**

When the elapsed times (sec) of the algorithms are analyzed separately, it is seen in Table 6 that ATSA achieved effective results for benchmark problems in a shorter time than other meta-heuristic algorithms.

Table 6. Elapsed time(s) of the algorithms in JSPs

Problem	PSO	TSA	ABC	GWO	ATSA	Problem	PSO	TSA	ABC	GWO	ATSA
ta01	10.62	9.84	7.25	14.28	5.59	ta41	67.06	63.02	40.79	93.19	32.37
ta02	10.64	9.95	7.14	14.87	5.51	ta42	67.08	62.40	40.74	92.40	34.45
ta03	10.59	9.87	7.12	14.76	5.54	ta43	67.16	62.55	40.71	91.00	34.54
ta04	10.65	9.91	7.06	14.64	5.71	ta44	67.06	62.42	40.72	91.90	32.63
ta05	10.46	9.90	7.05	14.62	5.59	ta45	67.84	62.48	40.58	91.14	33.24
ta11	17.90	16.86	11.69	25.15	8.77	ta51	98.47	94.81	61.21	150.96	51.02
ta12	17.75	16.77	11.65	25.07	8.98	ta52	98.19	93.78	61.20	165.25	51.52
ta13	17.84	16.75	11.59	24.98	8.81	ta53	98.43	93.89	61.20	173.27	51.64
ta14	17.76	16.76	11.59	24.92	8.87	ta54	101.13	93.77	61.07	174.05	53.39
ta15	17.77	16.68	11.62	24.93	8.70	ta55	106.27	97.66	60.97	177.39	51.02
ta21	30.49	28.92	19.40	43.32	14.89	ta61	200.46	185.20	105.87	331.11	90.07
ta22	30.81	28.99	19.33	43.38	14.85	ta62	205.70	197.31	105.95	333.13	90.83
ta23	30.68	28.90	19.34	43.10	14.91	ta63	203.71	197.84	105.96	300.66	88.11
ta24	30.55	28.84	19.33	43.05	14.89	ta64	206.30	197.79	108.91	302.13	88.06
ta25	30.74	28.81	19.35	42.90	15.22	ta65	214.24	201.86	111.61	298.76	88.02
ta31	38.32	35.79	23.76	53.82	19.00	ta71	788.35	713.20	395.36	1086.65	323.21
ta32	38.61	35.80	23.77	53.39	18.93	ta72	707.66	623.32	397.00	1021.44	322.50
ta33	38.17	35.66	23.72	53.22	18.91	ta73	710.71	594.60	414.71	854.60	336.77
ta34	38.24	35.74	23.71	53.25	18.90	ta74	709.07	594.44	392.64	843.49	326.04
ta35	38.25	35.71	23.63	53.19	18.77	ta75	613.23	593.48	367.95	848.01	317.04
AVG	24,34	22,82	15,45	34,04	12,07	AVG	269,91	244,29	150,76	376,03	124,82

Since 40 benchmark problems are considered in this study and convergence graphs for all of them cannot be given, the first 4 benchmark problems were chosen to show the convergence of meta-heuristic algorithms. The convergence graphs of the algorithms are given in Figure 7.

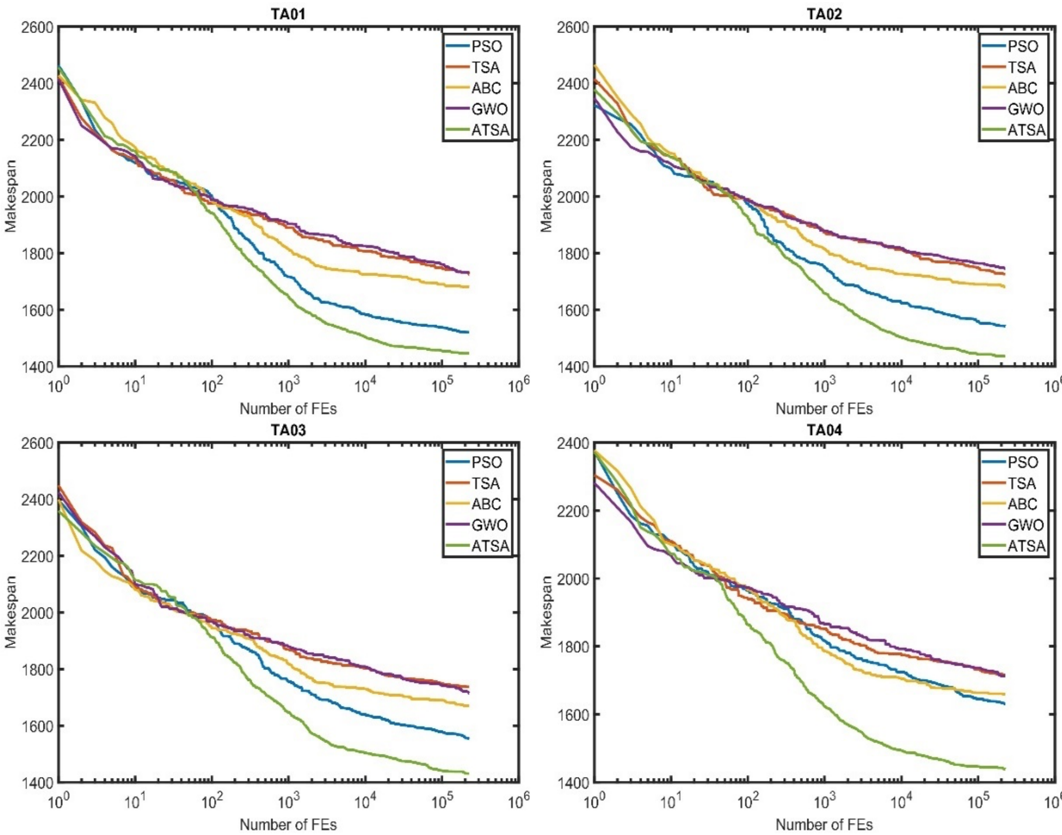

Figure 7. Convergence graphs of the algorithms in ta01-ta04

According to the convergence graphs, it is seen that the TSA and GWO algorithms converge very slowly and are insufficient to increase the solution quality. The closest convergence to the ATSA approach was obtained from the PSO algorithm. In all of the convergence graphs in Figure 7, it appears that the ATSA approach converges slowly at the first stage but continues to converge until the given number of FEs is completed.

Also, the Gantt chart of all algorithms belonging to the ta01 benchmark problem is given in Figure 8. Looking at the Gantt charts, it is seen that other algorithms cannot efficiently place the operations of the jobs and therefore the time lost increases. Such a situation causes other algorithms to increase the completion time.

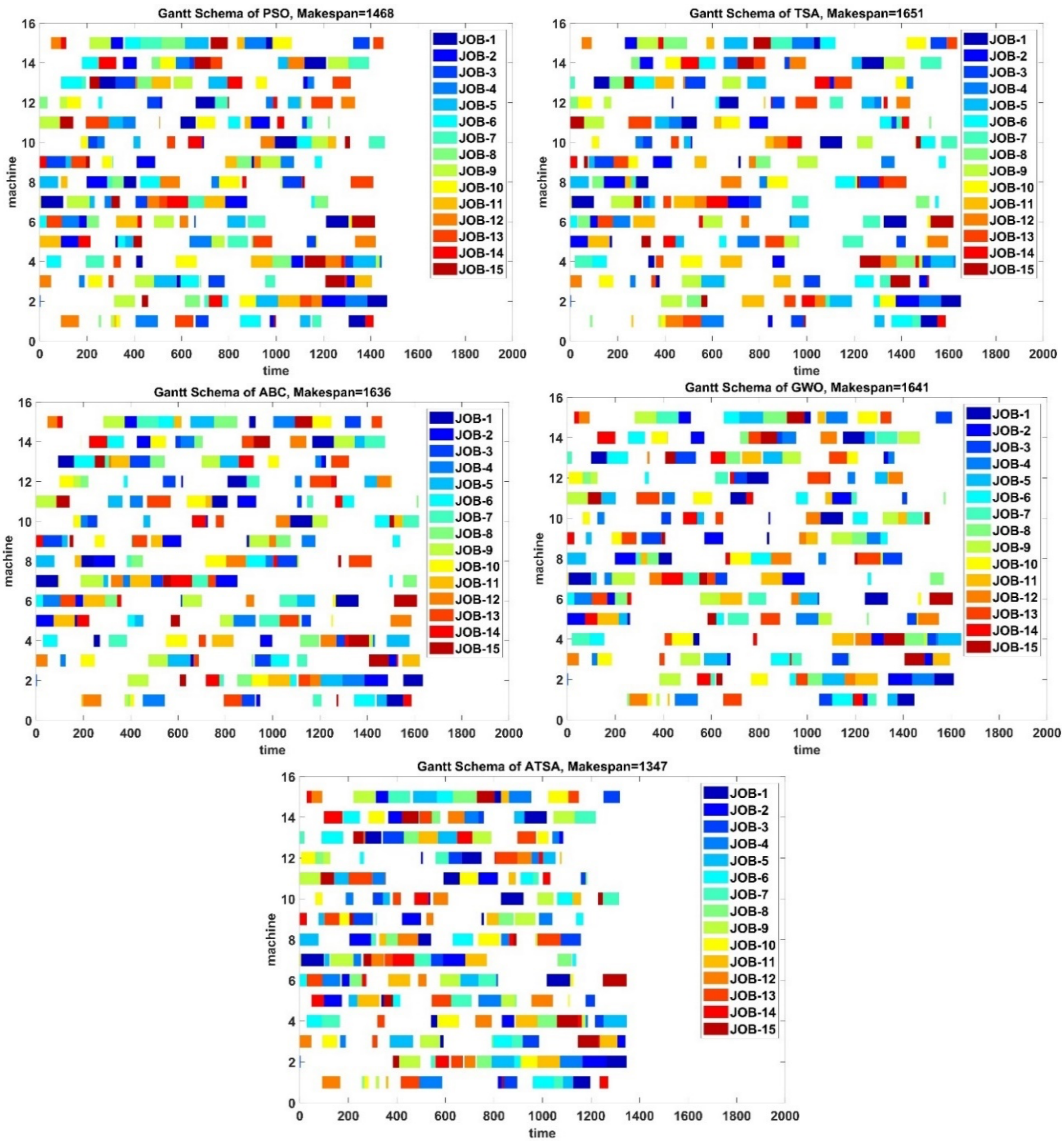

Figure 8. Gantt schemas of the algorithms for ta01

Two nonparametric statistical tests (Wilcoxon signed rank [40] and Friedman's test [41]) are used to evaluate the performances of the algorithms in this study.

Wilcoxon signed-rank test with a significance level of %5 is performed between ATSA and other algorithms and it is seen that ATSA produced remarkable results for all benchmark problems with other algorithms. With Friedman's test, the results obtained by the algorithms in the benchmark problems are statistically ranked (Table 7).

Table 7. Friedman's rank test of all algorithms

Problem	PSO	TSA	ABC	GWO	ATSA	Problem	PSO	TSA	ABC	GWO	ATSA
ta01	1.90	4.30	3.30	4.40	1.10	ta41	2.00	4.40	3.10	4.50	1.00
ta02	1.95	4.20	3.20	4.60	1.05	ta42	2.00	4.25	3.15	4.60	1.00
ta03	2.00	4.60	3.23	4.18	1.00	ta43	2.00	4.40	3.15	4.45	1.00
ta04	2.25	4.40	2.95	4.40	1.00	ta44	2.00	4.55	3.00	4.45	1.00
ta05	2.00	4.20	3.58	4.23	1.00	ta45	2.00	4.40	3.25	4.35	1.00
ta11	2.00	4.45	3.35	4.20	1.00	ta51	2.00	4.48	3.18	4.35	1.00
ta12	2.00	4.15	3.50	4.35	1.00	ta52	2.00	4.45	3.10	4.45	1.00
ta13	2.00	4.38	3.23	4.40	1.00	ta53	2.00	4.50	3.05	4.45	1.00
ta14	2.00	4.50	3.30	4.20	1.00	ta54	2.00	4.45	3.10	4.45	1.00
ta15	2.05	4.50	3.15	4.30	1.00	ta55	2.00	4.45	3.25	4.30	1.00
ta21	2.00	4.53	3.15	4.33	1.00	ta61	2.00	4.45	3.20	4.35	1.00
ta22	2.00	4.50	3.00	4.50	1.00	ta62	2.00	4.30	3.20	4.50	1.00
ta23	2.00	4.55	3.05	4.40	1.00	ta63	2.00	4.70	3.10	4.20	1.00
ta24	2.00	4.50	3.10	4.40	1.00	ta64	2.00	4.38	3.05	4.58	1.00
ta25	2.00	4.25	3.35	4.40	1.00	ta65	2.00	4.50	3.20	4.30	1.00
ta31	2.00	4.48	3.15	4.38	1.00	ta71	2.00	4.45	3.35	4.20	1.00
ta32	2.00	4.45	3.30	4.25	1.00	ta72	2.00	4.35	3.10	4.55	1.00
ta33	2.00	4.60	3.05	4.35	1.00	ta73	2.00	4.45	3.30	4.25	1.00
ta34	2.00	4.70	3.25	4.05	1.00	ta74	2.00	4.10	3.25	4.65	1.00
ta35	2.00	4.50	3.15	4.35	1.00	ta75	2.00	4.45	3.05	4.50	1.00

According to Friedman's test, it is seen that the best ranking value is obtained with the ATSA approach (Figure 9).

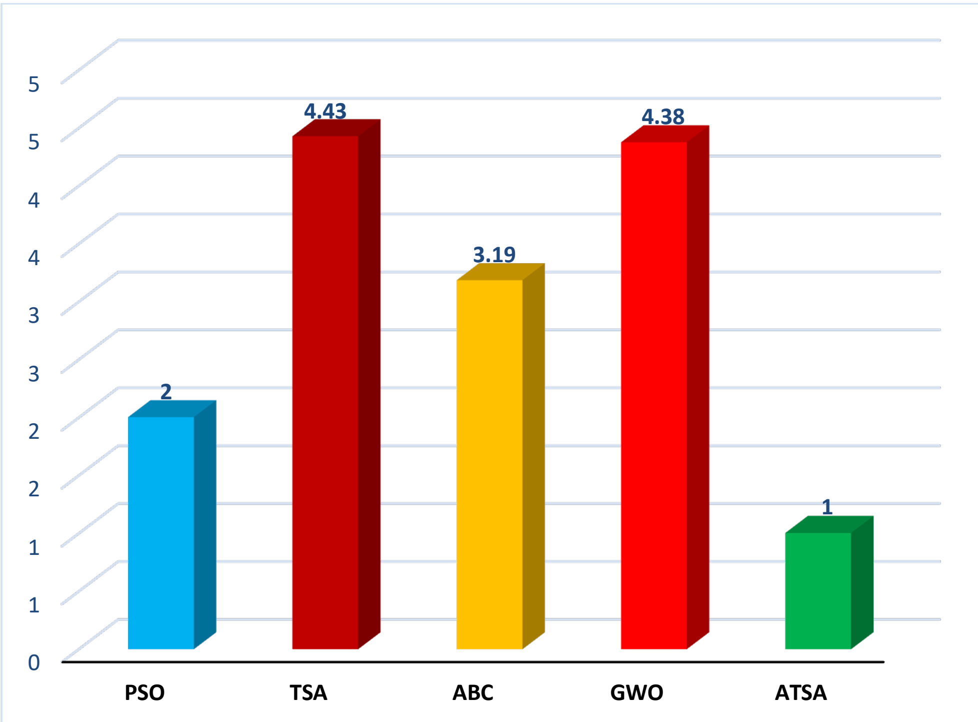

Figure 9. Average of Friedman's rank test results

By the Wilcoxon signed-rank test, it is desired to check whether the ATSA approach proposed in this study produces considerable results compared to other algorithms (Table 8). In Table 8, the (+) value shows that the null hypothesis is rejected, and (-) the null hypothesis is not rejected at the determined significance level.

Table 8. Wilcoxon sign test results

Problem	PSO		TSA		ABC		GWO		ATSA	
ta01	0.0001	(+)	0.8832	(+)	0.8858	(+)	0.8858	(+)	1	(-)
ta02	0.0001	(+)	0.8845	(+)	0.8858	(+)	0.8845	(+)	1	(-)
ta03	0.8858	(+)	0.8845	(+)	0.8845	(+)	0.8858	(+)	1	(-)
ta04	0.8858	(+)	0.8845	(+)	0.8858	(+)	0.8845	(+)	1	(-)
ta05	0.8858	(+)	0.8858	(+)	0.8845	(+)	0.8845	(+)	1	(-)
ta11	0.8845	(+)	0.8858	(+)	0.8845	(+)	0.8858	(+)	1	(-)
ta12	0.8858	(+)	0.8845	(+)	0.8858	(+)	0.8858	(+)	1	(-)
ta13	0.8845	(+)	0.8845	(+)	0.8845	(+)	0.8858	(+)	1	(-)
ta14	0.8858	(+)	0.8845	(+)	0.8858	(+)	0.8845	(+)	1	(-)
ta15	0.8858	(+)	0.8858	(+)	0.8832	(+)	0.8845	(+)	1	(-)
ta21	0.8858	(+)	0.8845	(+)	0.8858	(+)	0.8820	(+)	1	(-)
ta22	0.8858	(+)	0.8858	(+)	0.8832	(+)	0.8858	(+)	1	(-)
ta23	0.8858	(+)	0.8845	(+)	0.8845	(+)	0.8832	(+)	1	(-)
ta24	0.8858	(+)	0.8858	(+)	0.8845	(+)	0.8858	(+)	1	(-)
ta25	0.8845	(+)	0.8845	(+)	0.8845	(+)	0.8858	(+)	1	(-)
ta31	0.8858	(+)	0.8858	(+)	0.8832	(+)	0.8858	(+)	1	(-)
ta32	0.8858	(+)	0.8858	(+)	0.8858	(+)	0.8845	(+)	1	(-)
ta33	0.8832	(+)	0.8845	(+)	0.8845	(+)	0.8845	(+)	1	(-)
ta34	0.8858	(+)	0.8832	(+)	0.8858	(+)	0.8807	(+)	1	(-)
ta35	0.8832	(+)	0.8845	(+)	0.8845	(+)	0.8858	(+)	1	(-)
ta41	0.8845	(+)	0.8858	(+)	0.8845	(+)	0.8845	(+)	1	(-)
ta42	0.8832	(+)	0.8858	(+)	0.8858	(+)	0.8845	(+)	1	(-)
ta43	0.8858	(+)	0.8858	(+)	0.8858	(+)	0.8832	(+)	1	(-)
ta44	0.8858	(+)	0.8858	(+)	0.8858	(+)	0.8845	(+)	1	(-)
ta45	0.8858	(+)	0.8858	(+)	0.8858	(+)	0.8845	(+)	1	(-)
ta51	0.8845	(+)	0.8845	(+)	0.8858	(+)	0.8845	(+)	1	(-)
ta52	0.8832	(+)	0.8845	(+)	0.8845	(+)	0.8858	(+)	1	(-)
ta53	0.8858	(+)	0.8845	(+)	0.8858	(+)	0.8858	(+)	1	(-)
ta54	0.8832	(+)	0.8858	(+)	0.8845	(+)	0.8858	(+)	1	(-)
ta55	0.8858	(+)	0.8858	(+)	0.8858	(+)	0.8858	(+)	1	(-)
ta61	0.8858	(+)	0.8845	(+)	0.8858	(+)	0.8858	(+)	1	(-)
ta62	0.8858	(+)	0.8845	(+)	0.8858	(+)	0.8858	(+)	1	(-)
ta63	0.8832	(+)	0.8832	(+)	0.8845	(+)	0.8845	(+)	1	(-)
ta64	0.8858	(+)	0.8845	(+)	0.8858	(+)	0.8858	(+)	1	(-)
ta65	0.8858	(+)	0.8858	(+)	0.8858	(+)	0.8820	(+)	1	(-)
ta71	0.8858	(+)	0.8845	(+)	0.8858	(+)	0.8858	(+)	1	(-)
ta72	0.8858	(+)	0.8858	(+)	0.8858	(+)	0.8858	(+)	1	(-)
ta73	0.8858	(+)	0.8845	(+)	0.8858	(+)	0.8845	(+)	1	(-)
ta74	0.8858	(+)	0.8845	(+)	0.8858	(+)	0.8845	(+)	1	(-)
ta75	0.8858	(+)	0.8858	(+)	0.8858	(+)	0.8845	(+)	1	(-)

6. Conclusion and Future Works

In this study, the ATSA approach, which is an improved version of the TSA algorithm, is proposed. In the ATSA approach, three different mutation operators are gradually included in the seed generation mechanism of the basic TSA algorithm. To demonstrate the success of the ATSA approach, trials have been carried out on large-sized JSP problems.

For these experimental studies, 40 different and large-sized JSP benchmarks were selected. In experimental studies, basic versions of well-known meta-heuristic algorithms (PSO, ABC, and GWO) are discussed along with the TSA algorithm in the literature. The runtime environment (computer in use, experimental work done on the same software-MATLAB, etc.), run conditions (number of FEs, number of populations, etc.) were kept the same so that all metaheuristics were evaluated fairly. The results of the experimental studies and statistical approaches were evaluated in detail. It has been observed that the ATSA proposed in this study provides a statistically significant improvement over both the TSA algorithm and the meta-heuristic algorithms well known in the literature. In addition, the elapsed time by the ATSA algorithm in generating results has been significantly reduced compared to other meta-heuristic algorithms.

In future studies, new approaches can be suggested to improve the performance of JSPs by the discrete TSA algorithm, or other state-of-the-art algorithms which can run in the discrete search space.

Conflict Of Interest Statement

No conflict of interest was declared by the author.

References

[1] H. Mousavipoor, H. Farughi, and F. Ahmadizar, "Job shop scheduling problem based on learning effects, flexible maintenance activities and transportation times," *Journal of Industrial and Systems Engineering,* vol. 12, no. 3, pp. 107-119, 2019.

[2] Y. Yu, "A Research Review on Job Shop Scheduling Problem," in *E3S Web of Conferences*, 2021, vol. 253: EDP Sciences, p. 02024, doi:10.1051/e3sconf/202125302024

[3] M. R. Garey, D. S. Johnson, and R. Sethi, "The complexity of flowshop and jobshop scheduling," *Mathematics of operations research,* vol. 1, no. 2, pp. 117-129, 1976, doi:10.1287/moor.1.2.117

[4] R. Buddala and S. S. Mahapatra, "An integrated approach for scheduling flexible job-shop using teaching–learning-based optimization method," *Journal of Industrial Engineering International,* vol. 15, no. 1, pp. 181-192, 2019, doi:10.1007/s40092-018-0280-8

[5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, 1995, vol. 4: IEEE, pp. 1942-1948, doi:10.1109/ICNN.1995.488968

[6] Y. Fan, P. Wang, A. A. Heidari, H. Chen, and M. Mafarja, "Random reselection particle swarm optimization for optimal design of solar photovoltaic modules," *Energy,* vol. 239, p. 121865, 2022, doi:10.1016/j.energy.2021.121865

[7] L. Zhang, C. P. Lim, Y. Yu, and M. Jiang, "Sound classification using evolving ensemble models and Particle Swarm Optimization," *Applied Soft Computing,* vol. 116, p. 108322, 2022, doi:10.1016/j.asoc.2021.108322

[8] N. Karasekreter, M. A. Şahman, F. Başçiftçi, and U. Fidan, "PSO-based clustering for the optimization of energy consumption in wireless sensor network," *Emerging Materials Research,* vol. 9, no. 3, pp. 776-783, 2020, doi:10.1680/jemmr.20.00107

[9] M. A. Şahman, A. A. Altun, and A. O. Dündar, "A new MILP model proposal in feed formulation and using a hybrid-linear binary PSO (H-LBP) approach for alternative solutions," *Neural Computing and Applications,* vol. 29, no. 2, pp. 537-552, 2018, doi:10.1007/s00521-016-2467-5

[10] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," vol. 200, pp. 1-10, 2005.

[11] T. Ye, W. Wang, H. Wang, Z. Cui, Y. Wang, J. Zhao and M. Hu, "Artificial bee colony algorithm with efficient search strategy based on random neighborhood structure," *Knowledge-Based Systems,* vol. 241, p. 108306, 2022, doi:10.1016/j.knosys.2022.108306

[12] G. Yavuz, B. Durmuş, and D. Aydın, "Artificial Bee Colony Algorithm with Distant Savants for constrained optimization," *Applied Soft Computing,* vol. 116, p. 108343, 2022, doi:10.1016/j.asoc.2021.108343

[13] M. S. Kiran, "The continuous artificial bee colony algorithm for binary optimization," *Applied Soft Computing,* vol. 33, pp. 15-23, 2015, doi:10.1016/j.asoc.2015.04.007

[14] M. S. Kiran, H. Hakli, M. Gunduz, and H. Uguz, "Artificial bee colony algorithm with variable search strategy for continuous optimization," *Information Sciences,* vol. 300, pp. 140-157, 2015, doi:10.1016/j.ins.2014.12.043

[15] E. Kaya, S. A. Uymaz, and B. Kocer, "Boosting galactic swarm optimization with ABC," *International Journal of Machine Learning and Cybernetics,* vol. 10, no. 9, pp. 2401-2419, 2019, doi:10.1007/s13042-018-0878-6

[16] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software,* vol. 69, pp. 46-61, 2014, doi:10.1016/j.advengsoft.2013.12.007

[17] J. Adhikary and S. Acharyya, "Randomized Balanced Grey Wolf Optimizer (RBGWO) for solving real life optimization problems," *Applied Soft Computing,* vol. 117, p. 108429, 2022, doi:10.1016/j.asoc.2022.108429

[18] L. Yin and Z. Sun, "Distributed multi-objective grey wolf optimizer for distributed multi-objective economic dispatch of multi-area interconnected power systems," *Applied Soft Computing,* vol. 117, p. 108345, 2022, doi:10.1016/j.asoc.2021.108345

[19] K. Tütüncü, M. A. Şahman, and E. Tuşat, "A hybrid binary grey wolf optimizer for selection and reduction of reference points with extreme learning machine approach on local GNSS/leveling geoid determination," *Applied Soft Computing,* vol. 108, p. 107444, 2021, doi:10.1016/j.asoc.2021.107444

[20] M. S. Kiran, "TSA: Tree-seed algorithm for continuous optimization," *Expert Systems with Applications,* vol. 42, no. 19, pp. 6686-6698, 2015, doi:10.1016/j.eswa.2015.04.055

[21] X. Song, Y. Cao, and C. Chang, "A hybrid Algorithm of PSO and SA for Solving JSP," in *2008 Fifth international conference on fuzzy systems and knowledge discovery*, 2008, vol. 1: IEEE, pp. 111-115, doi:10.1109/FSKD.2008.430

[22] P. Pongchairerks and V. Kachitvichyanukul, "A comparison between algorithms VNS with PSO and VNS without PSO for job-shop scheduling problems," *International Journal of Computational Science,* vol. 1, no. 2, pp. 179-191, 2007.

[23] P. Pongchairerks and V. Kachitvichyanukul, "A two-level particle swarm optimisation algorithm on job-shop scheduling problems," *International Journal of Operational Research,* vol. 4, no. 4, pp. 390-411, 2009.

[24]D. Sha and C.-Y. Hsu, "A hybrid particle swarm optimization for job shop scheduling problem," *Computers & Industrial Engineering,* vol. 51, no. 4, pp. 791-808, 2006, doi:10.1016/j.cie.2006.09.002

[25] B. Z. Yao, C. Y. Yang, J. J. Hu, G. D. Yin, and B. Yu, "An improved artificial bee colony algorithm for job shop problem," in *Applied Mechanics and Materials*, 2010, vol. 26: Trans Tech Publ, pp. 657-660, doi:10.4028/www.scientific.net/AMM.26-28.657

[26] 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, doi:10.1109/ACCESS.2018.2833552

[27] T. Jiang, "A hybrid grey wolf optimization for job shop scheduling problem," *International Journal of Computational Intelligence and Applications,* vol. 17, no. 03, p. 1850016, 2018, doi:10.1142/S1469026818500165

[28] H. Xiong, S. Shi, D. Ren, and J. Hu, "A survey of job shop scheduling problem: the types and models," *Computers & Operations Research,* vol. 142, p. 105731, 2022, doi:10.1016/j.cor.2022.105731

[29] D. Yang, M. Wu, D. Li, Y. Xu, X. Zhou, and Z. Yang, "Dynamic opposite learning enhanced dragonfly algorithm for solving large-scale flexible job shop scheduling problem," *Knowledge-Based Systems,* vol. 238, p. 107815, 2022, doi:10.1016/j.knosys.2021.107815

[30] Y. Sun, J.-S. Pan, P. Hu, and S.-C. Chu, "Enhanced Equilibrium Optimizer algorithm applied in job shop scheduling problem," *Journal of Intelligent Manufacturing,* vol. 33, pp. 1-27, 2022, doi:10.1007/s10845-021-01899-5

[31] J. C. Bean, "Genetic algorithms and random keys for sequencing and optimization," *ORSA journal on computing,* vol. 6, no. 2, pp. 154-160, 1994, doi:doi.org/10.1287/ijoc.6.2.154

[32] J. Jiang, R. Han, X. Meng, and K. Li, "TSASC: tree–seed algorithm with sine–cosine enhancement for continuous optimization problems," *Soft Computing,* vol. 24, pp. 18627-18646, 2020, doi:10.1007/s00500-020-05099-w

[33] J. Jiang, M. Xu, X. Meng, and K. Li, "STSA: A sine Tree-Seed Algorithm for complex continuous optimization problems," *Physica A: Statistical Mechanics and its Applications,* vol. 537, p. 122802, 2020, doi:10.1016/j.physa.2019.122802

[34] A. C. Cinar, "Training Feed-Forward Multi-Layer Perceptron Artificial Neural Networks with a Tree-Seed Algorithm," *Arabian Journal for Science and Engineering,* vol. 45, no. 12, pp. 10915-10938, 2020, doi:10.1007/s13369-020-04872-1

[35] A. C. Cinar, H. Iscan, and M. S. Kiran, "Tree-Seed algorithm for large-scale binary optimization," *KnE Social Sciences,* vol. 2017, pp. 48-64, 2018, doi:10.18502/kss.v3i1.1396

[36] M. A. Sahman and A. C. Cinar, "Binary tree-seed algorithms with S-shaped and V-shaped transfer functions," *International Journal of Intelligent Systems and Applications in Engineering,* vol. 7, no. 2, pp. 111-117, 2019, doi:10.18201/ijisae.2019253192

[37] A. C. Cinar and M. S. Kiran, "Similarity and logic gate-based tree-seed algorithms for binary optimization," *Computers & Industrial Engineering,* vol. 115, pp. 631-646, 2018, doi:10.1016/j.cie.2017.12.009

[38] A. C. Cinar, S. Korkmaz, and M. S. Kiran, "A discrete tree-seed algorithm for solving symmetric traveling salesman problem," *Engineering Science and Technology, an International Journal,* vol. 23, no. 4, pp. 879-890, 2020, doi:10.1016/j.jestch.2019.11.005

[39] E. Taillard, "Benchmarks for basic scheduling problems," *European Journal Of Operational Research,* vol. 64, no. 2, pp. 278-285, 1993, doi:10.1016/0377-2217(93)90182-M

[40] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in statistics*: Springer, 1992, pp. 196-202.

[41] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics,* vol. 11, no. 1, pp. 86-92, 1940.