axioms

MDPI

Article

Three Hybrid Scatter Search Algorithms for Multi-Objective Job Shop Scheduling Problem

Leo Hernández-Ramírez [1], Juan Frausto-Solís [1,*,†], Guadalupe Castilla-Valdez [1,†], Javier González-Barbosa [1] and Juan-Paulo Sánchez Hernández [2]

[1] Tecnológico Nacional de México/IT Cd Madero, Ciudad Madero 89440, Mexico; iscleo1@gmail.com (L.H.-R.); gpe_cas@yahoo.com.mx (G.C.-V.); juan.gb@cdmadero.tecnm.mx (J.G.-B.)
[2] Dirección de Informática, Electrónica y Telecomunicaciones, Universidad Politécnica del Estado de Morelos, Jiutepec 62574, Mexico; juan.paulosh@upemor.edu.mx
* Correspondence: juan.frausto@gmail.com
† These authors contributed equally to this work.

Abstract: The Job Shop Scheduling Problem (JSSP) consists of finding the best scheduling for a set of jobs that should be processed in a specific order using a set of machines. This problem belongs to the NP-hard class problems and has enormous industrial applicability. In the manufacturing area, decision-makers consider several criteria to elaborate their production schedules. These cases are studied in multi-objective optimization. However, few works are addressed from this multi-objective perspective. The literature shows that multi-objective evolutionary algorithms can solve these problems efficiently; nevertheless, multi-objective algorithms have slow convergence to the Pareto optimal front. This paper proposes three multi-objective Scatter Search hybrid algorithms that improve the convergence speed evolving on a reduced set of solutions. These algorithms are: Scatter Search/Local Search (SS/LS), Scatter Search/Chaotic Multi-Objective Threshold Accepting (SS/CMOTA), and Scatter Search/Chaotic Multi-Objective Simulated Annealing (SS/CMOSA). The proposed algorithms are compared with the state-of-the-art algorithms IMOEA/D, CMOSA, and CMOTA, using the MID, Spacing, HV, Spread, and IGD metrics; according to the experimental results, the proposed algorithms achieved the best performance. Notably, they obtained a 47% reduction in the convergence time to reach the optimal Pareto front.

Keywords: JJSP; Scatter Search; CMOSA; CMOTA; hybrid algorithms

Citation: Hernández-Ramírez, L.; Frausto-Solís, J.; Castilla-Valdez, G.; González-Barbosa, J.; Sánchez Hernández, J.-P. Three Hybrid Scatter Search Algorithms for Multi-Objective Job Shop Scheduling Problem. *Axioms* **2022**, *11*, 61. https://doi.org/10.3390/axioms11020061

Academic Editors: Rafael Alejandro Espin Andrade, Witold Pedrycz, Laura Cruz-Reyes and Darjan Karabasevic

Received: 30 November 2021
Accepted: 21 January 2022
Published: 31 January 2022

1. Introduction

The Job Shop Scheduling Problem (JSSP) consists of a set of jobs, formed by operations, which must be processed in a set of machines subject to restrictions of precedence and resource capacity. For a job to be completed, all of its operations must be processed in a given sequence. This problem belongs to the NP-hard class [1], is challenging of solving it, and has significant industrial applicability [2]. In JSSP, we must determine the order or sequence for processing a set of jobs through several machines minimizing one or more objective functions. An essential function of JSSP is the coordination and control of complex activities, both optimum resource allocation and the sequence in performing those activities [3].

In a real operation context, it is common to consider more than one criterion simultaneously, which defines a multi- objective optimization problem whose solution involves generating a set of non-dominated solutions. This set provides the decision-maker with several alternatives to choose the one according to the needs of the manufacturing process [4–6].

A detailed analysis of the state-of-the-art for multi-objective JSSP shows that few works approach the problem from the multi-objective perspective, in which at least three objectives are modeled, and that use more than two performance metrics. In addition, there are practically no works that publish the fronts of non-dominated solutions.

The Pareto optimal front can be studied from the Set and Vector Optimization point of view, where useful applications have been found for classical and fractional optimization problems. Moreover, new local search strategies for multiobjective optimization have been developed [7–9]. On the other hand, the high-level soft computing approach allows the developing of popular metaheuristics for JSSP problems. The present work is focused on the latter approach; even though it is too popular, one of their problems for solving multi-objective JSSP is the slow convergence to obtain the Pareto Optimal Front. This situation is more critical in algorithms such as NSGA-II, MOEA/D, MOMARLA, MOPSO, CMOSA, and CMOTA. This work aims to improve convergence by using a hybrid algorithm based on the Scatter Search metaheuristics [10], evolving over reduced populations to improve the convergence speed.

This research proposes three new hybrid algorithms for the multi-objective JSSP problem (MO-JSSP). A dataset of 70 benchmark instances is used to evaluate their performance, applying a set of five metrics. Additionally, the non-dominated solution fronts obtained by each algorithm are presented, and strategies are incorporated to improve the quality and the execution time results regarding the state-of-the-art algorithms with which they are compared [11].

The three proposed algorithms are: (1) Scatter Search/Local Search (SS/LS), (2) Scatter Search/Chaotic Multi-Objective Threshold Accepting (SS/CMOTA), and (3) Scatter Search/Chaotic Multi-Objective Simulated Annealing (SS/CMOSA). They use three objectives known as: makespan, total tardiness, and total flow time. The computational experiments indicated that the proposed algorithms provide high-quality solutions for the MOJSSP, having obtained competitive solutions relative to CMOEA/D, CMOSA, and CMOTA on a set of traditional JSSP benchmarks instances.

2. Related Literature

The Pareto Archived Simulated Annealing (PASA) algorithm was applied to a JSSP with two objectives: the makespan and the mean flow time [12]. This algorithm was evaluated with 82 instances from the literature. The results obtained are evaluated in terms of the number of non-dominated schedules generated by the algorithm and the proximity of the obtained non-dominated front to the Pareto front.

A successful algorithm based on Simulated Annealing (SA) for several objectives named AMOSA was proposed [13]; also, it was reported that this algorithm performed better than some MOEA algorithms, such as the NSGA-II [14].

A two-stage genetic algorithm (2S-GA) was proposed for JSSP. This algorithm is applied to minimize three objectives makespan, total weighted earliness, and total weighted tardiness [15]. This algorithm is composed of two Stages: Stage 1 applies parallel GA to find the best solution for each individual objective function, and in Stage 2 the populations are combined. The performance of the algorithm is tested with benchmark instances and compared with results from other published papers. The experimental results show that 2S-GA is efficient in solving instances of the job shop scheduling problem in terms of the quality solution.

The Contemporary Design and Integrated Manufacturing Technology (CDMIT) laboratory proposed the Improved Multiobjective Evolutionary Algorithm based on Decomposition (IMOEA/D) to minimize the makespan, tardiness, and total flow time [16]. This algorithm was evaluated with 58 instances using the performance metrics Coverage [17] and Mean Ideal Media (MID) [18] to the evaluation. This algorithm stands out from the rest of the literature because it considered three objectives, applied two performance metrics, and reported good results.

In 2017, was proposed a hybrid algorithm with NSGA-II and a linear programming approach [19] to solve the FT10 instance [20]. In this paper, the objectives are to minimize weighted tardiness and energy costs. Furthermore, the authors applied the Hypervolume metric to evaluate the performance. Furthermore, in 2019, was proposed MOMARLA; this is a new algorithm based on Q-Learning to solve MOJSSP [21]. In this work, each agent represents a specific objective and uses two action selection strategies to find a diverse and accurate Pareto front.

Furthermore, in 2021, two multi-objective algorithms for minimizing makespan, total tardiness, and total flow time, were published. These algorithms are Chaotic Multi-Objective Simulated Annealing (CMOSA) and Chaotic Multi-Objective Threshold Accepting (CMOTA) [11]. They incorporate dominance criteria and a chaotic perturbation strategy to improve their performance. Experimental evaluation results indicated that they overpassed the state-of-the-art algorithms MOMARLA, MOPSO, CMOEA, and SPEA [21]. The algorithms proposed in the present paper seek to enhance the best algorithms of this group.

Finally, the scheduling will probably be directed to increasingly automated and use intelligent systems in the future. Under the Industry 4.0 environment, the computational workload could be greatly reduced and the systems probably will become more flexible and agile [22].

3. Background

This section describes basic concepts and algorithms in the multi-objective area which are related to this work. Furthermore, we present the multi-objective Job Shop Scheduling formulation and the main performance metrics used in this work.

3.1. Multiobjective Optimization Concepts

The Multi-Objective optimization algorithms use the concept of domination where two solutions are compared to determine if one solution dominates the other or not. Key concepts for Multi-Objective optimization are described below.

Pareto Dominance: For any optimization problem, solution A dominates another solution B if the following conditions are met: A is strictly better than B on at least one objective, and A is not worse than B in all objectives [23].

Non-dominated set: Among a set of P solutions, the subset of non-dominated solutions P1 is integrated by solutions that accomplish the following conditions:

- Any pair of P1 solutions must be non-dominated (one regarding the other)
- Any solution that does not belong to P1 is dominated by at least one member of P1 [23].

Pareto front: It is the graphical representation of the non-dominated solutions in the space of the objectives of the multi-objective optimization problem [24].

3.2. Performance Metrics

In the case of Multi-Objective Optimization, defining quality is complicated because two or more conflicting objective functions could exist. Then in an experimental comparison of different optimization algorithms, it is necessary to have the notion of performance. Some of the performance metrics are shown in Table 1.

Table 1. Performance metrics.

Metric	Type	Formula	
Mean Ideal Distance	Accuracy	$MID = \frac{\sum_{i=1}^{Q} C_i}{Q}$	(1)
Spacing	Diversity	$S = \sqrt{\frac{\sum_{i=1}^{Q} \left(d_i - \bar{d}\right)^2}{Q}}$	(2)
Hypervolume	Accuracy/Diversity	$HV = volume\left(\cup_{i=1}^{\lvert Q \rvert} v_i\right)$	(3)
Spread	Diversity	$\Delta = \frac{\sum_{k=1}^{M} d_k^e + \sum_{i=1}^{Q} \lvert d_i - \bar{d} \rvert}{\sum_{k=1}^{M} d_k^e + Q\ x\ \bar{d}}$	(4)
Inverted Generational Distance	Accuracy/Diversity	$IGD = \frac{\left(\sum_{j=1}^{\lvert T \rvert} \hat{d}_j^p\right)^{\frac{1}{p}}}{\lvert T \rvert}$	(5)

A large number of performance metrics or quality indicators can be found. These metrics consider mainly the following three aspects of a non-dominated solution set [25]:

- Convergence: that is a feature related to the closeness to the theoretical Pareto optimal front.
- Diversity: this feature for any distribution of non-dominated solutions is measured by Spread and Spacing.
- The number of solutions.

It is difficult to find a single performance metric that encompasses all of the above criteria. However, according to the characteristics they measure, the metrics can be grouped as [25]:

- Cardinality metrics: refers to the number of solutions found. A larger number of solutions is preferred.
- Accuracy metrics: refers to the convergence of the solutions. In other words, it indicates how distant the solutions are from the theoretical true Pareto front (PF_{true}). When the PF_{true} is unknown, an approximate Pareto front (PF_{approx}) is considered instead [25].
- Diversity metrics: They measure how distributed the solutions are in the front, that is, the relative distance between the solutions and the range of values covered by the solutions [25,26].

The MID metric is calculated with Equation (1). This metric calculates the closeness of the calculated Pareto front (PF_{calc}) solutions with an ideal point [18]. In this equations, Q is the number of solutions in the PF_{calc}, $C_i = \sqrt{\left(f_1^i\right)^2 + \left(f_2^i\right)^2 + \left(f_3^i\right)^2}$, and $f_1^i,\ f_2^i\ ,\ and\ f_3^i$ are the values of the i-th non-dominated solution for their first, second, and third objective function, respectively.

In Equation (2) is showed the formula to calculate S; this metric evaluates the distribution of the non-dominated solutions in the PF_{calc}. The algorithm with the smallest S value is the best [26]; d_i measures the distance in the space of the objectives functions between the i-th solution and its nearest neighbor; that is the j-th solution in the PF_{calc} of the algorithm, $\overline{d}$ is the average of d_i, that is $\overline{d} = \sum_{i=1}^{Q} \frac{d_i}{Q}$ and $d_i = min\left(\left|f_1^i(x) - f_1^j(x)\right| + \left|f_2^i(x) - f_2^j(x)\right| + \cdots + \left|f_M^i(x) - f_M^j(x)\right|\right)$, where $f_1^j,\ f_2^j$ are the values of the j-th non-dominated solution for their first and second objective function, respectively. Furthermore, M is the number of objective functions and i,j = 1, . . . ,Q.

Equation (3) shows the formula to calculate HV. In this formula v_i represents a hypercube which is constructed with a reference point W (this can be found constructing a vector with the worst values of the objective function) and the solution i as the diagonal corners of the hypercube [23]. This metric calculates the volume in the objective space that is covered by all solutions in the non-dominated set [27]. Therefore, an algorithm that obtains the highest HV value is the best. The data should be normalized to calculate the HV by transforming the value in the range [0, 1] for each objective separately.

The Spread metric is calculated using Equation (4), which considers the distance to the extreme points of the True Pareto front (PF_{true}) and was proposed to have a more precise coverage value [14]. Where d_k^e measures the distance between the extreme point of the PF_{true} for the k-th objective function and the nearest point of the PF_{calc}.

Finally, Equation (5) shows the formula to calculate IGD; this metric finds the average distance between the points of the PF_{true} to the PF_{calc} [28]. Where $T = \left\{t_1, t_2, \ldots t_{|T|}\right\}$ that is, the solutions in the PF_{true} and ∣T∣ is the cardinality of T, p is an integer parameter, in this case, p = 2 and $\hat{d}_j$ is the Euclidian distance from t_j to its nearest objective vector q in Q. In this case $d_j = min_{q=1}^{|Q|}\sqrt{\sum_{m=1}^{M}\left(fm(t_j) - fm(q)\right)^2}$ where fm(t) is the m-th objective function value of the t-th member of T.

Another important metric is the number of non-dominated solutions generated by the algorithm. The greater the number of solutions, the greater the number of alternatives the decision maker will have to choose the desired solution [4–6].

3.3. MOJSSP Formulation

In JSSP, there are a set of n jobs, consisting of operations, which must be processed in m different machines. There are a set of precedence constraints for these operations, and there is a resource capacity constraint for ensuring that each machine should process only one operation at the same time. The processing time of each operation is known in advance.

The objective of JSSP is to determine the sequence of the operations in each machine (the start and finish time of each operation) to minimize certain objective functions. The most common objective is the makespan, which is the total time in which all the problem operations are processed. Nevertheless, real scheduling problems are multi-objective, and several objectives should be considered simultaneously.

This work tries to optimize three objectives simultaneously, makespan, total tardiness, and total flow time.

- Makespan: It is the maximum time of completion of all jobs.
- Total tardiness: It is the total positive difference between the makespan and the due date of each job.
- Total flow time: It is the summation of the completion times of all jobs.

The formal MO-JSSP model can be formulated as follows [29,30]:

$$Optimize\ F(x) = \left[f_1(x), f_2(x), \ldots f_q(x)\right]\ subject\ to\ x\ \in S \tag{6}$$

where q is the number of objectives, x is the vector of decision variables, and S represents the feasible region, defined by the next precedence and capacity constraints, respectively:

$$\begin{array}{ll} t_j \geq t_i + p_i & \text{For all } ij \in O \text{when } i \text{ precedes } j \\ t_j \geq t_i + p_i \text{ or } t_i \geq t_j + p_j & \text{For all } ij \in O \text{when } M_i = M_j \end{array}$$

where

t_i, t_j are the starting times for the jobs $i, j \in J$.
p_i and p_j are the processing times for the jobs $i, j \in J$.
J:{J_1, J_2, J_3, … ,J_n} it is the sets of jobs.
M:{M_1, M_2, M_3, … M_m} it is the set of machines.
O is the set of operations Oj,i (operation i of the job j).

The objective functions of makespan, total tardiness, and total flow time, are defined by Equations (7)–(9), respectively.

$$f_1 = \min\left(max_{j=1}^{n} C_j\right) \tag{7}$$

where C_j is the makespan of job j.

$$f_2 = \min\left(\sum_{i=1}^{n} T_j\right) = \min\left(\sum_{j=1}^{n} \max\left(0, C_j - D_j\right)\right) \tag{8}$$

where T_j = max $(0, C_j - D_j)$ is the tardiness of job j, and D_j is the due date of job j and is calculated with $D_j = \tau \sum_{i=1}^{m} p_{j,i}$ [31], where $p_{j,i}$ is the time required to process the job j in the machine i. In this case, the due date of the j job is the sum of the processing time of all its operations on all machines, multiplied by a narrowing factor (τ), which is in the range $1.5 \leq \mathrm{t} \leq 2.0$ [31,32].

$$f_3 = \min\left(\sum_{j=1}^{n} C_j\right) \tag{9}$$

4. Proposed Hybrid Algorithms MO

Three hybrid algorithms based on Scatter Search (SS) are proposed for the solution of the MO-JSSP problem. Figure 1 shows the Scatter Search framework showing the three different process which distinguish the three proposed hybrid algorithms.

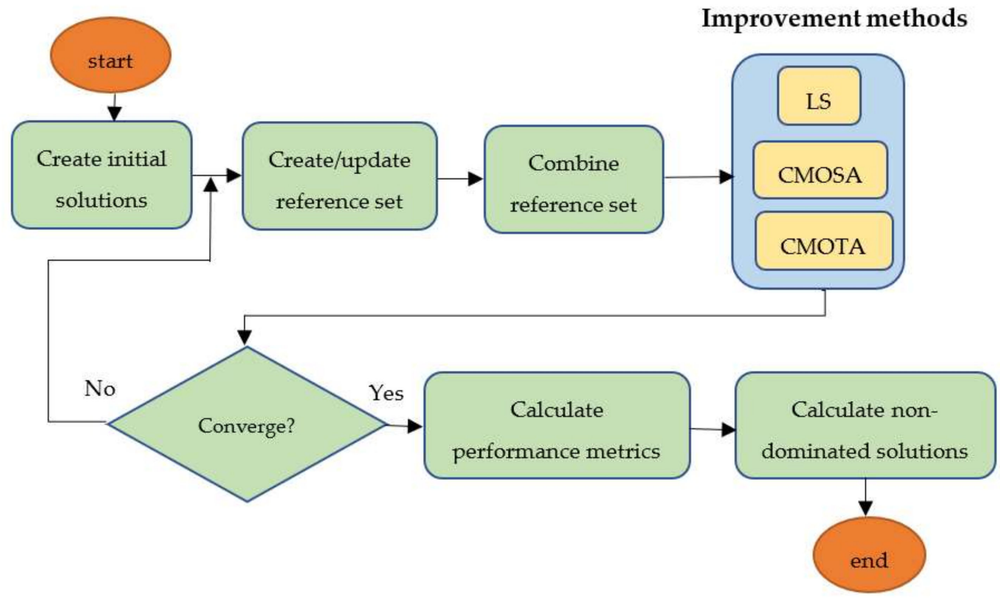

Figure 1. Scatter Search (SS) Framework.

Algorithm 1 contains the pseudocode of our Scatter Search Algorithm, which is described in detail in the next subsection. Notice that in line five, one of the algorithms CMOSA, CMOTA, or LS can be executed. The goal of Algorithm 1 is to improve the solutions of the reference set.

Algorithm 1. Scatter Search Algorithm

Input: *iterate=0*, MAXITERATIONS
Output: Non-dominated solutions front, metrics values
1: Generate initial solutions();
2: **while** (*iterate<=MAXITERATIONS*) **do**
3: Create/update reference set();
4: Combine reference set();
5: Improvement method(); //CMOSA, or CMOTA, or LS algorithm
6: *iterate++*
7: **End**
8: Calculate the non-dominate solutions front();
9: Calculate performance metrics;

4.1. Scatter Search General Framework

Scatter Search (SS) is an algorithm proposed by Glover [10], and it is composed of the following methods:

- Generator of diverse solutions, in which a set P of diverse solutions of size 30 is generated.
- Updater and creator of refSet, from the P solutions, the first three non-dominated and three most diverse are selected, using the Euclidean distance, to form the reference set (RefSet) of size 6.
- Combination of refSet. The six solutions in the refSet are mixed to obtain 30 new solutions. All possible combinations are generated in this process by taking the first half of one solution from the refSet and the second half from another.
- Improvement method. This process tries to improve each new solution created by the combination method. In this work, there are three different improvement methods implemented:

1. A Local Search (LS): consists of performing a set of iterations in which a regular perturbation is applied, which consists in exchanging two operations randomly selected from the current solution to generate a new one. The dominance criterion is applied at each iteration, and the new non-dominated solutions are stored.
2. A Chaotic Threshold Accepting Algorithm (CMOTA). Threshold Accepting (TA) is an algorithm proposed in [33]. In this enhanced method CMOTA, a version adapted to JSSP is used [11].
3. Chaotic Simulated Annealing Algorithm (CMOSA), SA was originally proposed in [34], and in 2021 a new version is implemented under the multiobjective approach [11].

In improvement processes 2) and 3), an analytical tuning process is performed for the algorithm parameters [35]. A regular perturbation is also applied to generate a new solution that is compared to the current one. From the previous comparison, the dominant one is selected, and the non-dominated is discarded. If both are not dominated, one is saved in the set of non-dominated solutions, and the other continues the search. When new non-dominated solutions are not found in both algorithms, a chaotic perturbation is applied. This perturbation consists of using the equation of the logistic maps [36] as a mechanism to escape from stagnation and search diversity in the solutions. A reheating process is also applied, which consists of raising the value of the current temperature parameter to be able to carry out a new scan. The improvement algorithms implemented are described in more detail in the following sections.

4.2. Hybrid Scatter Search with Local Search (SS/LS)

In this algorithm, a Local Search (LS) is applied in the solution improvement phase. Algorithm 2 shows the local search algorithm used. In this algorithm, a set of iterations is performed. In each of them, a regular perturbation is applied to the solution (exchange of two operations) to generate a new one. Dominance is verified between the current solution and the new one created with the perturbation in each iteration. In this verification, three possible cases are generated:

- Case A. If the new solution dominates the current one, then the new solution is saved and replaces the current one to continue the search process.
- Case B. If the current solution dominates the new one, then the new solution is discarded.
- Case C. If the current and the new solution are non-dominated, the current solution is saved, and the new one replaces the current one to continue the search.

4.3. Hybrid Scatter Search with Chaotic Simulated Annealing (SS/CMOSA)

The SS/CMOSA algorithm is based on Chaotic Simulated Annealing Algorithm (CMOSA) [11]. This hybrid algorithm, SS/CMOSA, receives the solutions obtained by a combination process as shown in Figure 1, while CMOSA is shown in Algorithm 3.: one that controls the stop condition and the other internal (Metropolis cycle) that controls the number of iterations carried out for each temperature parameter value.

In Algorithm 2, a perturbation is made to the current solution in the internal cycle to generate a new solution. Dominance is verified between the current solution and the new one in each iteration. In this verification, the same three possible previous cases are generated:

- Cases A and C are evaluated similarly to the SS/LS version.
- Case B. According to the Boltzmann probability distribution, when the current solution dominates the new one, the latter could be replaced by the former.

Additionally, a chaotic perturbation and a reheating are applied when stagnation occurs, consisting of a predetermined number of iterations without finding non-dominated solutions. The chaotic perturbation uses the equation of the logistic maps [37], whose main characteristic is that it generates different outputs even in small changes in its input data. Then chaos or chaotic perturbation is a process carried out to restart the search from another point in the space of solutions. Reheating is the process by which the current temperature parameter of the SA algorithm is high; this helps to perform reprocessing that allows reinitializing the search process.

Algorithm 2. Improvement method: Local Search (LS)

```
Input: iterate = 0, MAXITERATIONS
Output: Current solution, Non-dominated solutions
Current solution = Select one initial solution();
While (iterate ≤ MAXITERATIONS) do
   New solution = Perturbation(Current solution);
   Calculate makespan, tardiness, flowtime(New solution);
   if (New solution dominates Current solution) then
      Save(New solution);
      Current solution = New solution;
      NewSolutionDominatesCurrentSolution = Yes;
   end
   if (Current solution dominates New solution) then
      CurrentSolutionDominatesNewSolution = Yes;
   end
   if (Current and New solutions are non-dominated) then
      Current solution = New solution;
   end
   iterate++
end
```

4.4. Hybrid Scatter Search with Chaotic Threshold Accepting (SS/CMOTA)

The Chaotic Threshold Accepting (CMOTA) algorithm is used as an improvement method. In CMOTA (Algorithm 4), there are also two cycles such as CMOSA, one that controls the stop condition (temperature) and the other internal (Metropolis cycle) that controls the number of iterations that are carried out for each value of the temperature parameter.

The same three possible previous cases are generated. Cases A and C are evaluated in the same way as the SS/CMOSA version. In Case B, if the current solution dominates the new solution, the latter can replace the current one in the searching process by using a threshold established in the algorithm.

Similar to CMOSA, this algorithm also applies chaotic perturbation and a reheating process.

Algorithm 3. Improvement method: Chaotic Multi-Objective Simulated Annealing

```
Input: iterate = 0, MAXITERATIONS, MAXIMUM ALLOWED STAGNATION
SSSSSSTAGNATION
Output: Current solution, Non-dominated solutions
While (current temperature ≥ final temperature) do
      for each Metropolis cycle iteration do
         if stagnant = True then
               for each local search iteration do
                     if iteration = 1 then
                        New solution = chaoticPerturbation(Current solution);
                     Else
                        New solution = regularPerturbation(Current solution);
                     End
                     if (New solution dominates in all objectives to Current solution) then
                        Current solution = New solution;
                     End
               End
         Else
            New solution = regularPerturbation(Current solution);
         End
         if (New solution ≠ Current solution and it's not stored in the front) then
               if (New solution dominates Current solution) then
                     Save(New solution);
                     Current solution = New solution;
                     NewDominatesCurrent = True;
               End
```

Algorithm 3. Improvement method: Chaotic Multi-Objective Simulated Annealing

```
                if (Current solution dominates New solution) then
                    calculates decrement of objective functions;
                    if (random(0 - 1) < e^-decrementCurrenttemperature) then
                        Save(Current solution);
                        Current solution = New solution;
                        CurrentDominatesNew = True;
                    End
                End
                if (NewDominatesCurrent = False AND CurrentDominatesNew = False) then
                    Save(Current solution);
                    Current solution = New solution;
                End
            End
        End
        if (verifyCaught = True) then
            if (New solution is dominated by some stored solution) then
                stagnant = True;
                trappedCounter++;
                if (trappedCounter = MAXIMUM ALLOWED STAGNATION) then
                    verifyCaught = FALSE;
                End
            End
        End
        decrease current temperature;
    End
```

Algorithm 4. Improvement method: Chaotic Multi-Objective Threshold Accepting

```
    Input: iterate = 0, MAXITERATIONS, MAXIMUM ALLOWED STAGNATION
    SSSSSSTAGNATION
    Output: Current solution, Non-dominated solutions
    While (current temperature ≥ final temperature) do
        Threshold = current temperature
        for each Metropolis cycle iteration do
            if stagnant = True then
                foreach local search iteration do
                    if iteration = 1 then
                        New solution = chaoticPerturbation(Current solution);
                    else
                        New solution = regularPerturbation(Current solution);
                    end
                    if (New solution dominates in all objectives to Current solution) then
                        Current solution = New solution;
                    end
                end
            else
                New solution = regularPerturbation(Current solution);
            end
            if (New solution ≠ Current solution and it's not stored in the front) then
                if (New solution dominates Current solution) then
                    Save(New solution);
                    Current solution = New solution;
                    NewDominatesCurrent = True;
                end
                if (Current solution dominates New solution) then
                    if (random(0 - 1) < Threshold ) then
                        Save(Current solution);
                        Current solution = New solution;
                        CurrentDominatesNew = True;
                    end
                end
                if (NewDominatesCurrent = False AND CurrentDominatesNew = False) then
                    Save(Current solution);
                    Current solution = New solution;
                end
        end
        end
```

Algorithm 4. Improvement method: Chaotic Multi-Objective Threshold Accepting

```
            if (verifyCaught = True) then
                if (New solution is dominated by some stored solution) then
                    stagnant = True;
                    trappedCounter++;
                    if (trappedCounter = MAXIMUM ALLOWED STAGNATION) then
                        verifyCaught = FALSE;
                    end
                end
            end
            decrease current temperature;
        end
```

5. Computational Experimentation

This section describes the dataset used, the conditions of the experimentation as well as the results obtained.

5.1. Datasets

To perform the experimental evaluation of the algorithm, a benchmark of 70 instances of the problem are used. All the solutions of this dataset have different sizes and degrees of complexity. The instances are divided in six sets:

- Three instances denoted as FT06, FT10, FT20 of Fisher and Thompson [20],
- Ten instances denoted as ORB01–ORB10 of Applegate and Cook [38],
- 40 instances LA01-LA40 presented by Lawrence [39],
- Five instances ABZ5–ABZ9 taken from Baker [37],
- Four instances YN1, YN2, YN3 and YN4 taken from Yamada [40] and,
- Eight instances denoted as TA01, TA11, TA21, TA31, TA41, TA51, TA61, and TA71 taken fromTaillard [41].

The instance sizes of this dataset range from six jobs on six machines (the FT06 instance) to 100 jobs on 20 machines (the instance TA71).

5.2. Experiment Description

Two experiments were carried out with a different number of instances to evaluate the performance of the proposed algorithms.

The first experimentation was carried out with 58 common instances with only three algorithms (IMOEA/D [16], CMOSA, and CMOTA [11]) of the state-of-the-art that used the MID performance metric and the same three objectives. The second experimentation was carried out with 70 instances to compare the results with the CMOSA and CMOTA algorithms proposed in [11]. Each instance was executed 30 times using 30 initial solutions. The set of non-dominated solutions was obtained from the total solutions generated by the 30 executions

The performance metrics used are MID, Spacing, Spread, HV, IGD, Runtime, and the number of non-dominated solutions. Finally, we applied these metrics to the set of non-dominated solutions obtained by the algorithms at the end of their processes.

The execution of the proposed algorithms was carried out in a terminal of the Ehécatl cluster of the Technological Institute of Ciudad Madero, with the following characteristics: Intel® Xeon® processor at 2.30 GHz, Memory: 64 Gb (4×16 Gb) ddr4-2133, Linux operating system CentOS. C language was used for the implementation, and GCC compiler.

5.3. Comparative Results

Table 2 shows a comparison with the average results obtained by the MID metric for the 58 instances used in the algorithms IMOEA/D [16], CMOSA [11], CMOTA [11], and the proposed SS algorithms. We observed that our hybrid algorithms SS/LS, SS/CMOTA, and SS/CMOSA obtained the best results. Furthermore, the hybrid SS/CMOSA obtained the best result surpassing IMOEA/D by 17%.

Table 2. IMOEA/D, CMOSA, CMOTA, SS/LS, SS/CMOSA, and SS/CMOTA results using MID.

IMOEA/D [16]	CMOSA [11]	CMOTA [11]	SS/LS	SS/CMOSA	SS/CMOTA
18,727.04	15,729.65	16,567.07	15,579.30	15,509.06 *	15,600.19

* Best result obtained.

Table 3 shows the results obtained for each of the 70 instances by the three proposed algorithms, comparing them with the best of the state-of-the-art (CMOSA, CMOTA), taking as a reference the value obtained by the MID metric. The last row shows the average value of the MID metric for each of the algorithms. In this table, we observed that the SS/LS algorithm obtained a better performance since the value of the MID metric is smaller than the other algorithms analyzed.

Table 3. CMOSA, CMOTA, SS/LS, SS/CMOSA, and SS/CMOTA results using MID metric.

	INSTANCE	CMOSA [11]	CMOTA [11]	SS/LS	SS/CMOSA	SS/CMOTA
1	MT06	302.49	302.76	299.01	302.41	303.06
2	MT10	8968.94	9316.19	9514.63	8733.67	8841.43
3	MT05	18,599.1	18,789.48	18,095.69	17,334.15	18,036.81
4	ORB1	9423.07	9707.17	9263.34	9369.55	9400.38
5	ORB2	8129.62	8526.49	8207.86	8175.03	8050.28
6	ORB3	9443.81	9940.47	9618.31	9264.74	9491.26
7	ORB4	9205.69	9874.42	9552.22	9423.15	9407.45
8	ORB5	7983.66	8088.73	8211.05	7836.78	7898.4
9	ORB6	9489.54	9326.38	9639.36	8963.27	8962.67
10	ORB7	3630.48	3758.8	3526.24	3559.64	3605.42
11	ORB8	8268.95	8562.67	8242.37	8222.32	8150.52
12	ORB9	8778.61	9276.51	8872.7	8732.66	8993.75
13	ORB10	8672.67	9001.91	8796.23	8550.83	8568.04
14	LA01	5412.62	5597.42	5467.6	5479.76	5556.06
15	LA02	5031.77	5184.33	5006.39	4961.72	5013.23
16	LA03	4814.76	5134.95	4832.81	4918.78	5018.75
17	LA04	4861.93	5031.3	4929.76	4975.73	4928.21
18	LA05	4562.81	4714.16	4517.06	4555.52	4646.42
19	LA06	10,838.09	10,892.8	10,491.17	10,561.5	10,674.84
20	LA07	10,116.11	10,667.92	9562.58	10,137.43	10,186.86
21	LA08	10,042.56	10,009.01	9698.32	9896.07	10,389.5
22	LA09	11,036.3	11,411.96	10,893.66	10,861.87	11,408.09
23	LA10	11,202.9	11,365.93	10,898.95	11,333.39	11,302.51
24	LA11	19,027.38	19,802.15	17,161.79	19,450.04	19,350.97
25	LA12	15,911.86	16,330.94	14,623.73	15,892.3	15,600.23
26	LA13	17,928.81	18,139.11	17,160.73	17,743.58	17,990.65
27	LA14	20,538.67	20,433.48	18,234.8	20,062.12	20,181.51
28	LA15	19,316.53	20,217.82	18,501.01	19,201.56	19,108.72
29	LA16	8471.45	8504.98	8637.69	8250.89	8386.26
30	LA17	7360.64	7642.17	7393.33	7443.46	7349.59
31	LA18	7799.15	7970.16	7892.52	7762.4	7722.94
32	LA19	7886.52	8018.64	7928.09	7560.93	7822.05
33	LA20	8223.4	8504.94	8061.01	7976.07	8075.83
34	LA21	14,660.24	15,048.78	14,705	14,496.21	14,452.81
35	LA22	13,791.82	14,273.02	13,431.25	13,289.3	13,582.21
36	LA23	14,332.13	14,681.08	14,610.65	14,165.27	13,828.52
37	LA24	13,621.56	14,220.91	13,997.19	13,458.13	13,717.81
38	LA25	14,072.76	14,339.68	14,544.46	13,879.81	14,077.29
39	LA26	23,328.49	23,931.83	24,030.06	23,029.41	22,900.82
40	LA27	23,562.7	24,858.52	24,299.39	23,400.87	23,281.32
41	LA28	23,470.98	24,011.52	23,280.26	23,310.77	23,219.81
42	LA29	23,693.39	24,403.19	23,474.51	22,460.64	22,681.31
43	LA30	25,644.07	25,928.94	23,662.84	24,696.12	25,366.63

Table 3. *Cont.*

	INSTANCE	CMOSA [11]	CMOTA [11]	SS/LS	SS/CMOSA	SS/CMOTA
44	LA31	47,688.21	49,005.79	47,177.36	47,865.09	48,915.77
45	LA32	49,824.88	51,503.92	49,501.67	49,915.58	49,907.61
46	LA33	45,505.39	48,241.29	46,761.86	45,943.56	46,155.67
47	LA34	48,515.97	50,594.96	48,225.35	48,105.61	48,159.25
48	LA35	51,334.25	52,684.38	49,499.12	50,174.39	49,983.62
49	LA36	20,064.09	20,496.35	20,810.13	18,924.07	19,294.57
50	LA37	20,914.82	21,563.64	19,795.94	20,609.68	20,459.21
51	LA38	18,259.69	18,899.71	19,528.82	17,574.33	17,824.86
52	LA39	18,883.47	19,664.49	17,721.37	18,043.34	17,564.64
53	LA40	18,713.13	19,548.43	18,973.49	18,204.15	18,239.15
54	ABZ5	11,065.38	11,716.44	11,455.08	11,263.35	11,153.71
55	ABZ6	8562.26	8777.87	8547.61	8417.55	8412.77
56	ABZ7	13,456.23	14,117.93	13,666.34	13,533.63	13,433.57
57	ABZ8	13,876.7	14,964.84	13,829.45	13,659.58	13,917.87
58	ABZ9	14,196.03	14,482.01	14,338.05	13,611.68	13,857.5
59	YN01	19,600.96	19,081.94	18,852.58	18,614.34	18,513.43
60	YN02	19,478.97	29,387.6	20,381.93	18,911.92	19,185.31
61	YN03	19,373.68	37,958.48	19,617.2	18,652.67	18,646.75
62	YN04	20,795.84	20,085.05	20,537.93	20,019.8	20,302.96
63	TA01	18,854.25	21,077.84	17,900.21	18,028.25	18,372.28
64	TA11	28,456.37	19,829.68	27,650.77	27,426.51	26,732.44
65	TA21	36,784.31	21,706.29	38,361.95	35,354.63	35,020.33
66	TA31	57,276.28	58,910.97	55,016.34	51,624.54	55,253.5
67	TA41	67,727.19	70,740.02	65,540.4	66,060.72	67,477.21
68	TA51	149,060.96	147,166.01	142,621.68	145,129.97	140,875.79
69	TA61	163,794.79	165,040.63	157,115.67	156,896.57	159,441.06
70	TA71	634,090.21	633,330.29	593,742.24	614,082.8	604,620.66
AVG		30,680.19	31,233.15	29,727.69 *	29,861.83	29,846.47

* Best result obtained.

Table 4 summarizes the experiment results with 70 instances and five metrics; it contains the average values obtained by executing the algorithms 30 times. The first column has the name of the evaluated metric. The next two columns show the results of the two best state-of-the-art algorithms (CMOSA and CMOTA). Finally, the last three columns show the results for the three hybrid proposed algorithms (SS/LS, SS/CMOSA, and SS/CMOTA).

Table 4. Comparison of CMOSA, CMOTA and SS/LS, SS/CMOSA and SS/CMOTA results.

Metric	CMOSA [11]	CMOTA [11]	SS/LS	SS/CMOSA	SS/CMOTA
MID	30,680.19	31,233.15	29,727.69 *	29,861.83	29,846.47
Spacing	28,445.62	28,183.17	27,345.38	26,466.71	26,168.68 *
HV	0.42	0.42	0.42	0.42	0.44 *
Spread	24,969.31	23,401.88	23,935.95	20,961.48 *	21,307.87
IGD	1666.25	1870.94	1381.14 *	1449.77	1433.89
Runtime	495.22	229.42	133.65 *	1220.47	1202.18
Number of solutions	**10.57 ***	8.66	9.51	8.63	8.56

* Best result obtained.

In Table 4, the best values are highlighted and marked with an asterisk (*). An approximate Pareto front is used in the case of metrics in which it is necessary to use a True Pareto front [24]. The approximate Pareto front is generated by previous executions of developed algorithms throughout the study of this problem.

We can observe that SS/LS obtains the best result for MID and IGD metrics. This algorithm uses the shortest processing time, which means that it has the best convergence and generates the highest number of non-dominated solutions of the three proposed hybrid algorithms. The results indicate that the solutions found by SS/LS are closer to the origin point (0,0,0); they are closer on average to the approximate front and were achieved with the lowest amount of execution time. SS/CMOSA algorithm obtains the best Spread, which means that the generated solutions are very well distributed on the non-dominated solutions front. On the other hand, SS/CMOTA achieved the best Spacing and HV values, which indicate that this algorithm- has a more uniform Spacing and the best solution space coverage. In other words, the hybrid algorithms proposed in this paper obtained the best results for these datasets.

Finally, the non-dominated solution fronts obtained by the proposed algorithms for the 70 instances are included in Appendix A.

6. Conclusions

This paper presents three Hybrid Multi-Objective algorithms for JSSP, named SS/LS, SS/CMOSA, and SS/CMOTA. Three objectives are considered: makespan, total tardiness, and total flow time. Furthermore, we present an experimental evaluation applying six performance metrics.

Regarding the results from the comparison, we observe that SS/LS generates solutions closer to the origin point and PF_{approx}. It provides more solutions than the others' algorithms and uses the minimum runtime. SS/CMOSA generates solutions with a better distribution concerning PF_{approx}. Furthermore, SS/CMOTA generates better-distributed solutions in the PF_{calc} and better coverage, as shown by the HV metric. The results obtained by the proposed algorithms SS/LS, SS/CMOSA, and SS/CMOTA compared with some of the best algorithms in the literature show that they are among the best in the area. We highlight that our proposed SS/LS algorithm reduces processing time by 43% compared to the fastest in the state-of-the-art.

Author Contributions: Conceptualization, J.F.-S. and G.C.-V.; methodology, L.H.-R.; software, L.H.-R. and G.C.-V.; validation, G.C.-V. and L.H.-R.; formal analysis, J.F.-S. and J.-P.S.H.; investigation, G.C.-V. and L.H.-R.; resources, L.H.-R.; writing—original draft preparation, L.H.-R.; writing—review and editing, J.G.-B. and J.F.-S.; visualization, J.-P.S.H.; supervision, J.F.-S. and G.C.-V.; project administration, J.F.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: In this section, Data supporting are located in this paper.

Acknowledgments: We thank LanTI Lab for letting us use its computers and Conacyt for the scholarship to some of us while we studied our Ph.D.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Non-Dominated Solutions Obtained

The non-dominated solutions obtained by the proposed SS/LS, SS/CMOSA, and SS/CMOTA algorithms for the 70 instances used in this paper are shown in Tables A1–A6, Tables A7–A12 and Tables A13, A14 and A18–A21, respectively.

In these tables, MKS is the makespan, TDS is the total tardiness, and FLT is the total flow time. For each instance, the best value for each objective function is highlighted with an asterisk (*) and in bold type.

Table A1. Non-dominated solutions obtained by SS/LS for the JSSP instances proposed by [20].

		FT06			FT10			FT20	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**55** *	38.0	301	**1026** *	2258.0	9828	**1223** *	9170.0	16,791
2	55	33.0	306	1035	2174.0	9744	1224	9011.0	16,632
3	56	32.0	309	1037	2104.0	9674	1225	8651.0	16,296
4	57	27.0	297	1055	1815.0	9374	1228	8308.0	15,905
5	57	26.0	298	1055	1829.5	9288	1256	8110.0	15,769
6	57	24.5	306	1066	1386.0	8813	1261	8097.0	15,756
7	58	15.0	289	1202	**1322.5** *	8767	1270	8042.0	15,690
8	58	14.0	290	1202	1351.5	**8741** *	1274	7941.0	15,588
9	58	27.0	288				1279	**7914.0** *	**15,561** *
10	60	11.0	287						
11	60	10.0	288						
12	60	9.5	291						
13	62	9.5	284						
14	62	8.5	285						
15	69	**7.0** *	290						
16	73	14.5	283						
17	73	7.5	285						
18	75	13.5	283						
19	82	13.5	**280** *						

Table A2. Non-dominated solutions obtained by SS/LS for the JSSP instances proposed by [37].

		ORB1			ORB2			ORB3			ORB4			ORB5	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**1141** *	1401.5	9095	**930** *	768.5	8348	**1148** *	1764.0	9395	**1105** *	1156.5	9500	**1040** *	1517.0	8651
2	1144	1400.5	9094	942	752.5	8267	1159	**1710.5** *	9438	1106	1142.5	9487	1040	1511.0	8686
3	1146	**1376.5** *	**9070** *	942	758.5	8263	1159	1748.0	9391	1115	**1105.0** *	9412	1041	1406.0	8374
4				958	752.0	8373	1160	1776.0	**9335** *	1137	1171.0	**9322** *	1050	**682.0** *	7745
5				962	749.5	8306	1167	1727.0	9378	1155	1145.0	9362	1050	699.0	7725
6				963	666.0	8171							1052	690.0	7674
7				964	619.0	8143							1057	752.0	**7642** *
8				971	735.0	8066									
9				971	729.0	8134									
10				974	619.0	7883									
11				974	610.0	7952									
12				977	608.0	7893									
13				981	594.5	8192									
14				984	687.0	**7845** *									
15				988	**517.0** *	8119									
16				988	575.0	8071									
17				1001	558.0	8064									
		ORB6			**ORB7**			**ORB8**			**ORB9**			**ORB10**	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**1144** *	1639.0	9701	**411** *	108.5	3510	**1002** *	1577.5	8360	**997** *	1358.5	8906	**1021** *	1155.5	8985
2	1151	1510.0	9372	411	120.5	3499	1010	1569.5	8366	998	1336.0	8886	1030	992.0	8730
3	1154	1491.0	9380	411	118.5	3507	1012	1228.5	7872	1012	1185.0	8950	1047	812.0	8594
4	1157	1481.0	9644	411	113.5	3509	1014	1268.5	**7848** *	1012	1191.0	8915	1047	814.0	8589
5	1162	1441.0	9353	413	113.5	3493	1024	**1224.5** *	7868	1022	1204.5	8901	1057	769.0	8544
6	1163	1408.0	9320	413	106.5	3503				1022	1210.5	8866	1057	758.5	8826
7	1164	1394.0	**9306** *	413	110.5	3499				1024	1162.0	8927	1057	760.0	8719
8	1300	**1367.5** *	9557	413	117.5	**3489** *				1028	1175.5	8878	1060	763.0	8704
9				417	**100.0** *	3493				1063	1176.5	8866	1065	**754.0** *	**8536** *
10										1090	1150.5	8554			
11										1095	1197.5	8542			
12										1107	1136.0	8564			
13										1123	1133.0	8561			
14										1123	1168.5	8542			

Table A2. *Cont.*

	ORB6			ORB7			ORB8			ORB9			ORB10		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
15										1132	1144.0	**8483** *			
16										1138	1140.0	8500			
17										1142	**1122.5** *	8526			

Table A3. Non-dominated solutions obtained by SS/LS for the JSSP instances proposed by [38].

	LA01			LA02			LA03			LA04			LA05		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**666** *	1337.0	5414	**673** *	1241.0	5195	**627** *	1359.5	4837	**605** *	1466.5	5215	**593** *	1099.5	4507
2	666	1326.5	5505	676	1031.5	4954	628	1351.5	4831	607	1465.0	5183	593	1139.5	4359
3	666	1321.5	5509	678	1031.0	4921	630	1362.0	4796	610	1384.5	5133	600	1109.0	4490
4	666	1291.5	5565	679	1080.0	4902	630	1361.5	4801	611	1201.5	4950	605	1066.5	4334
5	668	1087.5	5275	694	981.0	4880	638	1187.0	4652	622	1142.5	4735	605	1097.5	4317
6	669	**1001.5** *	5275	694	1011.5	4867	640	1093.0	4558	622	1111.5	4807	607	1052.5	4294
7	673	1087.5	5246	694	972.5	4909	672	1039.5	4495	622	1121.5	4761	607	1023.0	4327
8	674	1071.5	5250	694	982.0	4872	672	1045.5	**4466** *	635	1075.0	4811	608	1025.0	4325
9	693	1074.0	5249	696	955.0	4845	691	1006.5	4475	638	1142.5	4720	626	1067.0	**4278** *
10	697	1074.0	5226	697	992.0	4782	730	**1005.5** *	4474	642	1046.5	4685	643	997.5	4296
11	716	1021.5	5189	697	959.0	4843				642	1091.5	4684	651	**996.5** *	4295
12	716	1006.5	5201	704	928.0	4812				652	923.0	4431			
13	716	1193.0	5161	705	986.0	4776				652	905.0	4457			
14	755	1105.0	**5153** *	709	912.0	4796				655	885.0	4441			
15				717	931.0	4794				660	**796.0** *	**4352** *			
16				717	970.0	**4752** *									
17				717	**894.0** *	4815									
18				717	960.5	4786									
19				796	964.0	4778									

	LA06			LA07			LA08			LA09			LA10		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**926** *	3857.5	9787	**890** *	3395.0	8955	**863** *	3475.5	9141	**951** *	3841.5	**10,108** *	**958** *	4104.0	10,058
2	926	3838.5	9804	890	3431.0	8929	863	3482.5	9134	951	3831.5	10,141	958	4094.5	10,086
3	926	3861.5	9704	890	3367.0	8972	863	3490.5	9126	951	3837.5	10,130	981	4116.0	9995
4	956	3826.0	9814	890	3366.5	8984	922	3257.0	8911	951	3827.5	10,163	1052	4097.0	**9976** *
5	965	3745.0	9733	967	3297.0	8833	929	**3232.0** *	**8906** *	953	**3826.5** *	10,221	1068	**4094.0** *	10,124
6	984	**3595.0** *	**9583** *	967	3350.5	**8825** *									
7				967	**3291.0** *	8845									

	LA11			LA12			LA13			LA14			LA15		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1222** *	7504.0	15,380	**1039** *	6408.5	13,419	**1150** *	7832.5	15,608	**1292** *	8325	**16,175** *	**1207** *	8422.5	16,499
2	1222	7486.0	15,438	1039	6489.5	13,398	1150	7843.5	15,586	1292	**8267.5** *	16,190	1207	8443.5	16,462
3	1222	7500.0	15,385	1044	6317.5	13,328	1160	7795.5	15,571	1292	8320.0	16,183	1223	8432.5	16,451
4	1338	**7478.0** *	**15,354** *	1048	6352.5	13,261	1182	7456.5	15,232				1223	8391.5	16,468
5				1051	6156.5	13,057	1191	7415.5	15,191				1317	8329.5	16,483
6				1056	6133.5	13,042	1199	7369.5	15,145				1326	8332.5	16,474
7				1060	6129.5	13,038	1208	**7345.5** *	**15,121** *				1334	**8180.5** *	**16,348** *
8				1134	**6083.5** *	13,094									
9				1134	6092.5	**13,001** *									

	LA16			LA17			LA18			LA19			LA20		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**981** *	1226.5	9053	**852** *	986.0	7650	**911** *	483.0	7864	**916** *	319.0	**7863** *	**952** *	364.5	8056
2	982	1185.5	9014	852	971.0	7652	911	464.0	7880	919	**312.0** *	7874	972	397.0	7928
3	985	951.5	8817	853	679.0	7250	911	490.0	7862				972	402.0	7863

Table A3. *Cont.*

		LA16			LA17			LA18			LA19			LA20	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
4	994	844.5	8713	853	654.0	7441	916	432.5	7792				983	352.5	8088
5	1009	873.5	8554	853	729.0	7245	921	415.5	7752				988	295.0	8020
6	1012	753.5	8435	856	691.0	7207	940	405.5	**7742** *				990	308.0	7984
7	1023	749.0	8490	860	682.0	7208	991	392.5	7841				990	301.0	8017
8	1049	747.0	8681	860	674.0	7255	991	**360.5** *	7865				997	322.0	7978
9	1050	658.5	8384	863	654.0	7371							1003	397.0	**7834** *
10	1050	684.5	8362	870	669.0	7345							1003	379.0	7874
11	1050	651.0	8436	871	708.5	7200							1094	290.5	8101
12	1052	653.0	8366	881	653.0	7255							1099	233.0	8036
13	1052	643.0	8430	882	577.5	7251							1112	**206.0** *	8022
14	1054	645.0	8365	883	511.5	7186							1113	262.0	8015
15	1065	630.0	8391	893	**506.5** *	**7120** *							1126	248.0	8014
16	1066	599.5	8356												
17	1069	**542.0** *	**8327** *												

		LA21			LA22			LA23			LA24			LA25	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**1156** *	2687.0	14,507	**998** *	2359.5	13,208	**1070** *	2564.5	14,459	**1040** *	2422.5	13,736	**1116** *	3976.0	15,131
2	1163	2620.0	14,440	998	2368.5	13,194	1077	2565.5	14,417	1040	2416.0	13,738	1118	3971.0	15,126
3	1168	2513.5	14,423	1001	2365.5	13,191	1077	2543.5	14,438	1040	2381.0	13,824	1121	3906.0	15,061
4	1169	2490.5	**14,389** *	1017	2470.0	13,169	1085	**2493.5** *	14,388	1040	2406.0	13,746	1133	3862.0	14,960
5	1170	**2480.0** *	14,425	1023	2363.0	13,178	1085	2515.5	14,367	1042	2394.5	13,712	1143	3737.0	14,772
6	1264	2481.5	14,423	1027	**2329.0** *	**13,144** *	1141	2593.5	14,343	1042	2395.5	13,694	1143	3832.0	14,759
7							1141	2585.5	14,351	1048	2332.5	**13,652** *	1149	3411.0	14,560
8							1185	2520.0	14,178	1048	2327.5	13,653	1167	3060.0	14,154
9							1205	2498.0	**14,156** *	1149	**2303.5** *	13,894	1171	2987.0	13,977
10										1149	2319.5	13,890	1171	3011.0	13,916
11													1174	2917.0	13,821
12													1176	2877.0	13,800
13													1179	2828.0	13,736
14													1182	2786.5	13,968
15													1183	2770.5	13,946
16													1183	2766.5	13,948
17													1187	2703.5 *	13,885
18													1187	2707.5	13,883
19													1200	2822.0	13,761
20													1220	2894.0	13,698
21													1220	2878.0	13,704
22													1220	2950.0	13,675
23													1228	2861.0	13,683
24													1244	2837.0	**13,646** *

		LA26			LA27			LA28			LA29			LA30	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**1367** *	8629.5	24,384	**1391** *	**7060.5** *	23,236	**1341** *	6677.5	22,696	**1318** *	8396.5	23,242	**1369** *	7102.0	23,122
2	1369	8571.5	24,326	1391	7083.5	23,189	1347	6611.5	22,630	1321	8323.5	23,169	1372	6977.0	22,997
3	1377	7612.5	23,174	1391	7100.5	**23,174** *	1353	6505.5	22,338	1324	8299.5	23,145	1374	6934.0	22,954
4	1386	7498.5	23,060	1391	7074.5	23,215	1363	6374.0	22,397	1329	8252.5	23,146	1382	6760.0	22,780
5	1392	6400.5	22,003				1364	6321.5	22,340	1334	8113.5	22,959	1384	6805.0	22,769
6	1395	6407.5	21,990				1368	6423.5	22,256	1341	7533.5	22,334	1386	6699.0	22,671
7	1403	6406.5	21,989				1368	6405.5	22,333	1341	7487.5	22,371	1394	6676.0	22,640
8	1470	**6398.5** *	**21,981** *				1373	**6226.0** *	22,249	1341	7500.5	22,357	1446	6608.0	22,572
9							1390	6385.5	22,218	1370	7207.5	22,013	1473	6324.0	22,281
10							1408	6299.5	**22,126** *	1370	7203.5	22,061	1477	6270.0	22,227
11							1408	6296.5	22,132	1382	7119.5	22,013	1480	**6259.0** *	**22,218** *
12										1387	7059.5	21,953			
13										1388	7052.5	21,946			
14										1390	7151.5	21,942			
15										1390	7155.5	21,914			
16										1399	7088.5	21,879			
17										1401	7011.5	21,905			
18										1403	7104.5	21,863			
19										1438	7010.5	21,904			
20										1448	6659.5	21,418			
21										1449	**6610.5** *	**21,369** *			

Table A3. *Cont.*

	LA31			LA32			LA33			LA34			LA35		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1784** *	20,361.5	42,992	**1850** *	21,329.0	45,992	**1719** *	21,064.0	43,370	**1746** *	21,499.5	44,511	**1908** *	23,093.5	46,321
2	1787	20,344.5	43,082	1851	21,304.0	45,967	1721	19,672.0	41,983	1747	21,420.5	44,432	1909	22,752.5	45,980
3	1803	20,227.5	42,928	1852	19,752.0	44,496	1727	19,668.0	41,979	1758	21,416.5	44,428	1911	22,690.5	45,918
4	1803	20,215.5	42,953	1852	19,733.0	44,561	1727	19,693.0	**41,971** *	1772	20,699.5	43,711	1937	22,640.5	45,868
5	1808	20,290.5	42,921	1855	**19,656.0** *	**444,00** *	1835	**19,661.0** *	41,972	1777	20,194.0	43,093	1938	20,907.5	44,132
6	1812	20,258.5	42,889							1777	20,181.5	43,193	1941	20,761.5	43,950
7	1827	19,744.5	42,438							1781	20,092.0	43,037	1945	20,540.5	43,765
8	1829	19,643.5	42,430							1791	19,928.5	42,890	1950	20,464.5	43,689
9	1832	19,603.5	42,390							1792	**19,886.5** *	**42,848** *	1954	20,441.5	43,666
10	1833	19,624.5	42,318										1956	20,437.5	43,662
11	1834	**195,92.5** *	**42,286** *										2016	**20,421.5** *	**43,646** *

	LA36			LA37			LA38			LA39			LA40		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1520** *	3470.5	21,071	**1537** *	1572.0	19,894	**1474** *	2716.0	19,396	**1358** *	1590.5	18,036	**1423** *	2368.0	18,972
2	1526	3348.0	20,673	1537	1563.0	19,901	1480	2706.0	19,435	1359	1419.5	17,909	1425	2307.0	18,929
3	1533	3344.5	20,793	1545	1537.0	19,735	1484	2693.5	19,407	1359	1417.5	17,911	1458	2240.0	18,847
4	1549	3300.0	20,625	1545	1552.0	19,538	1493	2639.5	19,318	1376	1307.0	17,698	1458	2244.0	18,763
5	1549	3274.5	20,804	1550	1523.5	19,666	1504	3015.0	19,085	1377	1127.0	17,539	1473	2139.0	18,639
6	1559	3244.0	20,569	1550	1539.0	19,637	1504	2976.0	19,225	1377	1130.0	17,523	1473	**2105.5** *	18,741
7	1566	3224.0	20,637	1550	1575.0	19,496	1531	2795.5	**18,977** *	1378	1118.0	17,548	1505	2116.0	**18,616** *
8	1566	3191.5	20,650	1551	1515.0	19,853	1542	**2616.5** *	19,336	1383	1073.0	17,536			
9	1566	3237.0	20,562	1552	1515.5	19,758				1383	1094.0	17,506			
10	1593	3186.0	20,653	1559	**1496.0** *	19,519				1391	1193.0	17,486			
11	1596	2940.5	20,405	1559	1511.0	**19,432** *				1422	1026.5	**17,386** *			
12	1596	2984.0	20,294							1422	**1020.5** *	17,424			
13	1596	2992.0	20,233												
14	1739	3197.5	20,149												
15	1758	2862.5	20,111												
16	1758	**2855.5** *	20,241												
17	1769	3181.5	**20,099** *												

Table A4. Non-dominated solutions obtained by SS/LS for the JSSP instances proposed by [39].

	ABZ5			ABZ6			ABZ7			ABZ8			ABZ9		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1286** *	365.0	11,502	**996** *	333.0	8597	**781** *	3157.5	14,129	**799** *	2922.0	14,229	**824** *	3002.0	**13,974** *
2	1312	434.0	11,447	997	336.0	8587	782	3129.5	14,101	801	2885.0	14,225	899	2999.5	14,006
3	1321	360.0	11,564	997	359.5	8535	785	3013.5	13,985	802	2887.0	14,210	899	2987.5	14,012
4	1324	302.0	11,423	1001	330.0	8634	786	2994.5	13,975	803	2836.0	14,169	912	**2960.0** *	13,990
5	1324	341.0	11,330	1009	270.0	8505	786	2997.5	13,966	803	2848.0	14,162			
6	1324	**260.0** *	11,444	1010	295.0	8362	800	2718.5	13,588	806	2840.0	14,141			
7	1326	465.5	11,273	1014	267.0	8600	803	2703.5	13,573	807	2705.0	14,049			
8	1326	445.5	11,297	1015	290.0	8455	853	2523.5	13,247	808	2699.0	14,043			
9	1332	310.0	11,314	1020	177.5	8395	853	2589.5	13,207	808	2787.0	14,029			
10	1332	349.0	**11,221** *	1025	162.5	8476	853	2525.5	13,210	810	2773.0	14,015			
11	1332	268.0	11,382	1026	**83.0** *	8536	854	2584.5	13,202	812	2372.5	13,516			
12	1333	315.0	11,277	1030	338.0	8355	855	2279.5	13,092	813	2185.5	13,329			
13				1038	210.5	8381	858	2272.5	13,054	825	2126.5	13,133			
14				1045	257.5	**8337** *	858	2278.5	13,041	828	2106.5	13,113			
15							868	**2200.5** *	12,942	829	2077.5	13,084			
16							868	2246.5	12,888	830	2031.0	13,037			
17							871	2229.5	12,934	831	2045.0	13,035			
18							874	2240.5	**12,882** *	839	1976.0	12,936			
19										869	1971.0	**12,931** *			
20										872	**1966.0** *	12,966			
21										875	1968.0	12,965			

Table A5. Non-dominated solutions obtained by SS/LS for the JSSP instances proposed by [40].

	YN01			YN02			YN03			YN04		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1165** *	2888.5	20,219	**1098** *	2807.0	20,198	**1172** *	3586.5	20,744	**1176** *	3057.0	21,124
2	1180	2816.5	20,149	1108	2798.0	20,189	1173	3551.5	20,697	1214	2596.5	20,442
3	1195	2308.0	19,714	1109	2679.0	20,136	1174	3388.0	20,564	1215	2596.5	20,433
4	1195	2299.5	19,732	1109	**2676.0** *	20,139	1175	3093.0	20,220	1221	**2579.5** *	20,425
5	1199	2257.0	**19,659** *	1109	2708.0	20,122	1178	3070.0	20,197	1222	2582.5	**204,19** *
6	1199	**2253.5** *	19,683	1112	2688.0	20,132	1179	3068.0	20,201			
7				1112	2689.5	20,131	1182	3062.0	20,211			
8				1112	2685.0	20,135	1183	3049.0	20,176			
9				1112	2717.0	20,108	1189	2739.5	**19,794** *			
10				1131	2686.0	20,130	1189	**2716.0** *	19,818			
11				1131	2715.0	**20,106** *						

Table A6. Non-dominated solutions obtained by SS/LS for the JSSP instances proposed by [41].

	TA01			TA11			TA21			TA31			TA41		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1534** *	2668.5	19,931	**1776** *	7002.0	28,574	**2216** *	7498.5	37,447	**2159** *	**18,102.0** *	**52,002** *	**2599** *	23,300.5	70,219
2	1534	2746.5	19,906	1802	6410.0	27,917	2228	7409.5	37,358				2609	23,287.5	70,206
3	1534	2612.0	19,962	1803	6408.0	27,915	2230	7301.5	37,120				2614	23,223.5	70,142
4	1534	2597.5	19,986	1810	**6325.0** *	**27,832** *	2243	7261.5	37,102				2625	23,211.5	70,130
5	1539	2577.5	19,908				2245	**7230.5** *	**37,035** *				2633	23,208.5	70,127
6	1539	**2551.0** *	19,920										2637	23,071.5	69,990
7	1570	2825.0	**19,888** *										2643	22,930.5	69,849
8													2649	**22,909.5** *	**69,828** *

	TA51			TA61			TA71		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**3153** *	72,768.0	129,645	**3478** *	72,727.0	148,963	**6064** *	363,427.5	514,764
2	3154	72,471.0	129,348	3479	72,655.0	148,891	6066	350,616.5	501,953
3	3162	72,307.0	129,184	3495	72,641.0	148,877	6071	350,572.5	501,909
4	3164	72,239.0	129,116	3545	72,603.0	148,975	6079	350,545.5	501,882
5	3212	71,599.0	128,476	3550	72,475.0	148,847	6096	350,395.5	501,732
6	3225	71,340.0	128,217	3559	72,411.0	148,783	6106	349,762.5	501,099
7	3229	71,313.0	128,190	3560	72,327.0	148,699	6110	**349,695.5** *	**501,032** *
8	3270	70,959.0	127,836	3604	72,125.0	148,571			
9	3317	70,584.0	127,461	3606	72,113.0	148,559			
10	3318	**70,445.0** *	**127,322** *	3611	72,085.0	148,531			
11				3614	72,058.0	148,504			
12				3629	72,046.0	148,492			
13				3633	71,979.0	148,425			
14				3780	**71,928.0** *	**148,374** *			

Table A7. Non-dominated solutions obtained by SS/CMOSA for the JSSP instances proposed by [20].

	FT06			FT10			FT20		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**55** *	38.0	301	**973** *	1190.0	8623	**1191** *	8027.0	15,642
2	55	30.0	305	973	1204.0	8592	1198	7859.0	15,474
3	56	29.0	308	1003	1163.0	8596	1200	7811.0	15,426
4	56	37.0	304	1016	1138.5	8720	1236	**7518.5** *	**15,181** *
5	57	23.5	305	1036	**991.5** *	**8474** *			
6	57	27.0	297						
7	57	26.0	298						
8	58	9.5	280						
9	62	8.5	285						
10	65	12.5	**278** *						
11	69	**7.0** *	290						

Table A8. Non-dominated solutions obtained by SS/CMOSA for the JSSP instances proposed by [37].

	ORB1			ORB2			ORB3			ORB4			ORB5		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1133** *	1549	9551	**912** *	678.5	8275	**1082** *	1576.5	9289	**1043** *	1152.5	9202	**935** *	916.5	8073
2	1135	1220.5	**8916** *	918	900.5	8242	1092	1530.5	9243	1046	1128.5	9238	938	684.5	7712
3	1147	1159.5	9188	941	650.5	8323	1102	1419.5	9093	1056	1102	9268	960	676	7881
4	1158	**1145.5** *	9191	943	693.5	8251	1107	1450.5	9040	1061	1143.5	**9174** *	968	545	7563
5				944	635.5	8244	1143	1376.5	9066	1069	1123	9266	998	**541.5** *	**7509** *
6				946	624.5	8434	1156	**1350.5** *	9040	1079	1097.5	9429			
7				947	584.5	8271	1158	1408	9033	1083	1015.5	9409			
8				948	680	8167	1160	1414	8970	1086	1088	9231			
9				952	547.5	8321	1180	1392	**8968** *	1112	1086.5	9327			
10				955	496.5	8185				1154	1075.5	9297			
11				957	418.5	8192				1156	**1014.5** *	9416			
12				966	486	7975									
13				968	415	7930									
14				968	542.5	7911									
15				972	365.5	7975									
16				974	315.5	7833									
17				980	382.5	7728									
18				982	376.5	**7724** *									
19				1020	**278.5** *	7919									

	ORB6			ORB7			ORB8			ORB9			ORB10		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1046** *	831	8980	**405** *	118.5	3597	**945** *	1557	8335	**971** *	1223	8907	**970** *	562	8460
2	1075	757	9003	416	114.5	**3499** *	946	1672	8292	972	1212	8896	974	566	8402
3	1075	788	8978	416	109.5	3512	948	1671	8278	976	1176.5	8797	979	790	8320
4	1083	751	8997	426	108.5	3510	949	1517.5	8289	978	1134.5	8691	983	646	8390
5	1084	775	**8716** *	429	**75.5** *	3575	963	1489.5	8257	984	1099	8815	985	661	8385
6	1104	756	8808	434	76.5	3506	971	1236	8005	988	1106	8730	1001	555	8637
7	1113	685	8783				973	1189.5	7892	997	1037.5	8670	1013	745.5	8353
8	1117	**558.0** *	8810				986	1198	7876	1012	1304.5	8635	1045	554.5	8730
9	1144	675.5	8718				992	**999.5** *	7639	1015	1021.5	8668	1055	535	8507
10							997	1036.5	**7629** *	1021	967.5	8591	1063	668	8297
11										1022	872	8490	1074	528.5	8658
12										1037	875	8434	1080	**501.0** *	8634
13										1038	827.5	8477	1081	704	**8289** *
14										1044	871	8474			
15										1055	**780.0** *	8383			
16										1148	860	8373			
17										1189	873	**8359** *			

Table A9. Non-dominated solutions obtained by SS/CMOSA for the JSSP instances proposed by [38].

	LA01			LA02			LA03			LA04			LA05		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**666** *	1419.0	5461	**655** *	1206.5	5156	**606** *	1406.0	4952	**590** *	1363.0	5069	**593** *	1094.5	4418
2	668	1138.0	5385	660	1273.5	5143	623	1446.5	4879	594	1407.0	5056	600	1071.5	4392
3	672	1302.0	5344	663	1050.0	4868	627	1404.5	4884	595	1324.0	4985	604	1063.0	4479
4	679	1275.0	5331	667	1024.0	4955	628	1361.5	4855	598	1214.5	4916	607	1069.0	4463
5	700	1233.0	5238	671	999.5	4904	628	1460.0	4828	598	1227.0	4854	608	1104.5	4372
6	701	1179.0	5274	677	914.5	4799	629	1333.0	4807	599	1182.0	4930	610	1164.0	4370
7	702	1157.0	5345	680	839.5	4724	634	1315.5	4764	608	1138.0	4725	611	1054.0	4378
8	715	1144.0	5346	680	800.5	4730	636	1346.0	4714	611	1122.5	4862	634	1089.0	4374
9	721	1095.0	5302	681	804.5	4717	638	1292.0	4785	614	1122.0	4870	648	**1005.0** *	4299
10	743	1169.0	5287	688	796.5	**4659** *	643	1280.5	4792	614	1123.5	4851	672	1008.0	**4293** *
11	768	1026.0	5277	712	790.5	4703	646	1295.0	4778	616	1032.5	4630			
12	769	1056.0	5265	729	**729.0** *	4683	648	1328.0	4724	625	1025.0	4667			
13	771	1029.0	5255	730	781.0	4680	650	1293.5	4690	629	1012.5	4682			
14	771	**1017.0** *	5268				650	1174.5	4749	642	1011.5	4678			
15	771	1221.5	5239				652	1163.0	4629	658	982.0	4726			
16	850	1212.5	**5230** *				657	1157.5	4731	661	991.5	4609			
17							660	1158.5	4668	663	**917.5** *	**4575** *			
18							662	1209.0	4625						
19							664	1128.5	4637						
20							666	1053.5	4558						
21							679	1011.0	4516						

Table A9. *Cont.*

		LA01			LA02			LA03			LA04			LA05	
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
22							719	1001.5	4541						
23							722	1010.5	4533						
24							750	**988.5** *	**4511** *						
		LA06			**LA07**			**LA08**			**LA09**			**LA10**	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**927** *	3972.5	9939	**890** *	4047.0	9541	**863** *	3722.5	9460	**951** *	3922.5	10,287	**958** *	4539.5	10,554
2	935	3814.0	9782	891	3849.5	9467	864	3706.5	9444	960	3852.5	10,140	958	4588.5	10,528
3	940	3780.0	9748	892	3909.5	9410	867	3614.0	9350	1005	3711.0	**10,029** *	959	**4403.0** *	10,433
4	944	**3754.0** *	**9722** *	904	3700.0	**9166** *	868	3608.5	9346	1005	**3708.0** *	10,058	960	4058.0	**10,088** *
5				927	3690.5	9292	869	3603.5	9341						
6				967	**3674.5** *	9216	870	3615.0	9310						
7							879	3539.5	9277						
8							889	3485.5	9223						
9							891	3474.5	9154						
10							900	3453.5	9124						
11							900	3460.5	9097						
12							901	3422.5	9109						
13							945	3201.5	8939						
14							954	**3075.5** *	**8813** *						
		LA11			**LA12**			**LA13**			**LA14**			**LA15**	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**1222** *	9206.0	17,155	**1039** *	7244.0	14,217	**1150** *	8442.0	16,221	**1292** *	**9630.5** *	**17,552** *	**1207** *	9213.5	17,373
2	1227	9176.5	17,203	1039	7233.0	14,247	1153	8128.0	15,907				1209	8874.5	16,937
3	1228	9091.5	17,118	1051	7265.0	14,216	1161	7806.0	15,585				1244	8797.0	**16,836** *
4	1232	**9031.5** *	**17,052** *	1054	7195.0	14,209	1186	**7665.0** *	**15,444** *				1264	8784.0	16,915
5				1059	7091.0	14,105							1288	**8772.5** *	16,940
6				1061	7174.5	14,054							1289	8774.0	16,905
7				1064	7074.0	14,080									
8				1069	**7038.0** *	**14,044** *									
		LA16			**LA17**			**LA18**			**LA19**			**LA20**	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**946** *	787.5	8253	**796** *	912.0	7633	**855** *	558.5	7932	**861** *	140.0	**7444** *	**908** *	519.0	8156
2	967	673.5	8212	801	805.0	7617	859	553.5	7894	870	136.0	7471	910	487.0	8153
3	979	485.0	8238	805	797.0	7493	865	471.5	7833	873	135.0	7628	911	442.5	7795
4	1035	510.5	8235	805	762.0	7496	872	571.5	7808	892	**133.0** *	7493	911	409.5	7831
5	1038	430.0	8237	809	683.5	7312	873	567.5	7803				918	389.0	8128
6	1050	**272.0** *	8067	809	698.5	7305	874	545.5	7829				921	396.0	8043
7	1054	488.5	8066	841	627.5	**7175** *	876	445.5	7890				922	407.0	8037
8	1109	569.0	**8043** *	857	603.5	7256	879	497.0	7730				922	407.5	7680
9				863	548.5	7311	880	390.0	7593				925	376.0	7981
10				878	569.5	7206	881	370.0	7591				925	400.5	**7671** *
11				898	547.5	7304	882	352.0	7671				926	262.0	7875
12				925	**481.5** *	7265	884	366.5	7573				931	316.5	7873
13							892	334.0	7720				933	297.0	7829
14							898	329.0	7567				942	**227.5** *	8042
15							919	257.0	7623				959	383.5	7700
16							924	318.0	7603				986	260.0	8021
17							928	292.0	**7531** *				1027	348.0	7798
18							929	245.5	7545				1032	352.0	7792
19							945	244.0	7708						
20							985	242.5	7659						
21							986	**240.5** *	7582						
		LA21			**LA22**			**LA23**			**LA24**			**LA25**	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**1102** *	2769.5	14,487	**984** *	2337.5	13,192	**1039** *	2200.5	14,235	**993** *	2225.5	13,648	**1029** *	2956.5	14,010
2	1115	2756.5	14,610	988	2309.5	13,166	1041	**1819.0** *	**13,730** *	998	1885.5	13,334	1032	2875.5	13,668
3	1120	2713.5	14,469	992	**2146.0** *	12,986				999	1876.0	13,196	1033	2845.0	13,995
4	1122	2654.5	14,414	1046	2222.5	12,979				1008	1874.0	13,344	1040	2844.5	13,637
5	1124	2386.5	14,317	1104	2281.5	**12,955** *				1009	1860.5	13,309	1040	2824.5	13,773
6	1125	2374.0	14,281							1010	1832.0	13,302	1041	2807.5	13,864
7	1129	2345.5	14,114							1014	1831.0	13,259	1048	2517.5	13,517
8	1134	2275.5	14,072							1015	1707.0	13,188	1052	2469.5	13,462
9	1138	2331.5	13,979							1024	**1705.0** *	13,179	1056	2567.0	13,408
10	1150	2232.0	14,223							1025	1726.0	**13,153** *	1057	2266.5	13,223

Table A9. *Cont.*

		LA21			LA22			LA23			LA24			LA25	
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
11	1191	2183.0	14,057										1075	2256.5	13,308
12	1200	**1952.0** *	**13,938** *										1102	**2207.0** *	**13,192** *
		LA26			**LA27**			**LA28**			**LA29**			**LA30**	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**1242** *	6684.5	22,457	**1286** *	6708.0	22,911	**1264** *	7023.0	23,046	**1248** *	7318.5	22,212	**1386** *	7671.0	23,691
2	1244	6698.5	22,425	1287	6337.0	22,540	1280	6668.0	22,691	1259	7195.0	22,040	1393	7649.0	23,669
3	1250	6359.5	22,067	1301	6193.0	22,441	1284	6461.5	22,474	1264	6847.5	21,731	1402	**7380.0** *	23,400
4	1287	6429.5	21,981	1321	6187.5	22,412	1289	6434.5	22,447	1271	6743.5	21,627	1528	7426.0	23,334
5	1306	6329.5	22,102	1322	**5990.5** *	**22,212** *	1291	6328.0	22,351	1274	6556.5	21,423	1546	7410.0	**23,318** *
6	1307	6361.0	22,033				1295	6107.0	22,062	1277	6516.5	21,399			
7	1308	6239.5	22,012				1296	6079.0	22,034	1279	6533.5	21,337			
8	1310	6255.0	21,992				1306	**5988.0** *	**21,942** *	1282	6439.5	21,327			
9	1315	6232.0	21,969							1293	6465.5	21,272			
10	1385	**6172.5** *	**21,870** *							1314	6363.5	21,257			
11										1317	6333.5	21,227			
12										1333	6327.5	21,221			
13										1338	6381.5	21,188			
14										1354	**6070.0** *	**20,839** *			
		LA31			**LA32**			**LA33**			**LA34**			**LA35**	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**1784** *	20,682.5	43,469	**1850** *	20,891.5	45,745	**1719** *	**19,234.5** *	**41,688** *	**1721** *	21,725.5	44,737	**1888** *	22,157.5	45,385
2	1786	20,647.5	43,434	1851	20,716.5	45,570				1723	20,594.5	43,606	1900	21,987.5	45,135
3	1787	20,586.5	43,373	1855	20,520.5	45,374				1729	20,578.5	43,590	1906	21,994.0	45,106
4	1793	20,451.5	43,238	1859	20,513.5	45,367				1740	20,173.5	43,147	1910	21,783.0	**44,894** *
5	1796	20,415.5	43,202	1882	20,494.5	45,348				1748	20,030.5	43,042	1932	**21,772.5** *	44,897
6	1806	20,417.5	42,953	1888	20,482.5	45,336				1750	**19,808.5** *	**42,820** *			
7	1818	20,419.5	42,952	1891	**20,458.5** *	**45,312** *									
8	1859	**20,335.5** *	**43,122** *												
		LA36			**LA37**			**LA38**			**LA39**			**LA40**	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**1351** *	1724.0	19,137	**1558** *	2472.0	20,884	**1330** *	1226.5	17,545	**1334** *	1213.5	17,855	**1337** *	1710.5	18,569
2	1353	1587.0	18,872	1559	2070.5	20,550	1348	**1027.5** *	17,503	1337	1194.5	18,144	1338	1686.0	18,546
3	1378	1479.5	18,817	1565	2023.5	20,628	1351	1172.0	**17,445** *	1346	1176.0	18,119	1340	1623.0	18,483
		LA36			**LA37**			**LA38**			**LA39**			**LA40**	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
4	1380	1471.5	18,856	1565	2013.5	20,653	1376	1135.5	17,447	1364	1189.5	17,958	1349	1593.5	18,481
5	1387	1438.5	18,809	1566	2063.5	20,495				1365	887.0	**17,837** *	1350	1589.5	18,485
6	1391	1437.5	18,808	1570	1957.0	20,563				1365	**839.0** *	17,843	1352	1509.5	18,588
7	1397	1313.0	18,714	1590	1886.5	20,610							1355	1740.5	18,450
8	1399	1316.0	18,707	1593	2043.0	20,357							1359	1419.5	18,037
9	1405	**1265.0** *	**18,637** *	1594	1992.5	20,550							1373	1417.5	18,542
10				1600	1807.0	20,573							1388	1195.0	17,804
11				1609	1778.0	20,366							1389	1123.0	17,596
12				1621	1762.0	20,350							1389	1106.0	17,673
13				1623	1756.0	20,599							1391	1108.0	17,637
14				1625	2048.5	20,261							1393	1127.0	17,584
15				1632	1946.5	**20,159** *							1393	1110.0	17,621
16				1633	1848.5	20,262							1403	**1033.0** *	**17,482** *
17				1653	**1743.0** *	20,288									
18				1681	1832.5	20,246									
19				1691	1895.5	20,234									

Table A10. Non-dominated solutions obtained by SS/CMOSA for the JSSP instances proposed by [39].

		ABZ5			ABZ6			ABZ7			ABZ8			ABZ9	
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1254** *	177.5	11,280	**947** *	289.0	8473	**734** *	2472.0	13,486	**746** *	2186.0	13,532	**767** *	2367.0	13,380
2	1255	108.0	11,213	961	207.5	8512	739	2421.0	13,440	747	2183.0	13,529	767	2354.0	13,406
3	1280	231.0	11,180	963	282.5	8486	739	2471.0	13,435	763	2220.0	13,437	788	2336.0	13,388
4	1281	174.0	11,191	966	306.5	8405	740	2418.0	13,437	764	2299.0	13,425	788	2365.0	13,378

Table A10. *Cont.*

	ABZ5			ABZ6			ABZ7			ABZ8			ABZ9		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
5	1285	**157.5** *	11,160	967	253.5	8461	743	2365.0	13,188	767	**2165.5** *	13,516	789	2349.0	13,362
6	1296	255.0	11,148	970	173.0	8389	745	2394.0	13,184	768	2252.5	**13,302** *	791	2326.5	13,452
7	1319	243.5	**11,147** *	971	321.0	8326	750	2335.5	13,325				806	**2270.5** *	**13,338** *
8				976	154.5	8289	752	2309.5	13,299						
9				982	139.0	8419	769	2180.5	13,146						
10				982	166.5	8282	773	**2157.5** *	**13,123** *						
11				983	114.0	8365									
12				984	138.0	8306									
13				984	65.5	8435									
14				991	117.5	8229									
15				994	67.0	8395									
16				996	50.5	8207									
17				996	21.5	8229									
18				998	161.0	**8179** *									
19				1003	52.5	8189									
20				1097	**20.5** *	8577									

Table A11. Non-dominated solutions obtained by SS/CMOSA for the JSSP instances proposed by [40].

	YN01			YN02			YN03			YN04		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1021** *	1355.0	**18,401** *	**1037** *	1567.0	18,942	**1029** *	1739.0	18,783	**1136** *	2134.0	19,837
2	1060	**1233.0** *	18,679	1046	1533.0	18,877	1032	1502.0	18,671	1138	2110.0	**19,813** *
3				1048	1525.0	18,869	1036	1518.0	18,520	1139	2073.0	19,842
4				1054	1604.5	18,735	1038	1464.0	18,449	1181	**2011.5** *	20,020
5				1056	**1502.5** *	18,766	1051	**1426.5** *	**18,380** *			
6				1086	1596.5	**18,721** *						

Table A12. Non-dominated solutions obtained by SS/CMOSA for the JSSP instances proposed by [41].

	TA01			TA11			TA21			TA31			TA41		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1335** *	1372.5	18,209	**1535** *	5652.5	27,205	**1915** *	5306	35,250	**1944** *	15,386.50	49,247	**2364** *	17,422.50	64,248
2	1346	1302	18,110	1537	5589.5	27,142	1916	5229	35,256	1951	**15,375.5** *	**49,236** *	2385	17,264.50	64,090
3	1351	1229.5	18,073	1538	5661.5	26,977	1929	4896.5	35,136				2423	17,247.50	64,073
4	1354	1143.5	18,037	1551	5407.5	27,055	1940	5077.5	35,122				2438	17,231.50	64,057
5	1359	881.5	17,807	1569	5297.5	26,743	1969	4868.5	35,039				2454	16,909.00	63,800
6	1359	**821.5** *	17,918	1594	5194	26,582	2000	4432	34,499				2459	16,893.00	63,784
7	1368	859.5	17,917	1596	5193.5	26,787	2007	**4347.0** *	**34,421** *				2511	16,866.50	63,785
8	1380	1031	17,792	1604	5105	26,765							2518	16,603.50	63,522
9	1389	1138.5	**17,784** *	1604	5121	26,698							2521	16,509.50	63,428
10	1392	928.5	17,791	1613	**4998.0** *	**26,630** *							2522	**16,358.5** *	**63,277** *

	TA51			TA61			TA71		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**3036** *	70,728.00	127,605	**3175** *	68,199.00	144,645	**5813**	360,343.50	511,680
2	3037	70,693.00	127,570	3194	65,524.00	141,970	5825	350,203.50	501,540
3	3062	69,876.00	126,753	3196	**64,945.0** *	141,391	5842	348,580.50	499,917
4	3100	69,835.00	126,712	3200	64,939.00	**141,385** *	5852	**348,429.5** *	**499,766** *
5	3110	**69,572.0** *	**126,449** *						

Table A13. Non-dominated solutions obtained by SS/CMOTA for the JSSP instances proposed by [20].

	FT06			FT10			FT20		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**55** *	38.0	301	**974** *	1441.5	8911	**1202** *	8107.0	15,752
2	55	30.0	305	980	1392.5	8666	1218	**7977.0** *	**15,576** *
3	56	37.0	304	983	1389.5	8663			
4	56	29.0	308	1004	1211.0	8706			
5	57	23.5	305	1039	**1140.0** *	8676			
6	57	26.0	298	1058	1384.0	8642			
7	57	27.0	297	1071	1346.0	8604			

Table A13. *Cont.*

	FT06			FT10			FT20		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
8	58	9.5	280	1101	1289.5	8669			
9	60	8.5	**274** *	1104	1346.0	**8576** *			
10	69	**8.0** *	291						

Table A14. Non-dominated solutions obtained by SS/CMOTA for the JSSP instances proposed by [37].

	ORB1			ORB2			ORB3			ORB4			ORB5		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1111** *	1331.5	9164	**926** *	564.5	8131	**1096** *	1856.5	9724	**1044** *	1067.0	9367	**914** *	834.5	7721
2	1135	1268.0	9315	929	514.5	8139	1100	1811.5	9636	1054	1063.5	9366	962	829.5	7909
3	1145	1310.5	9301	936	466.0	8066	1110	1757.0	9391	1056	1157.0	9349	969	786.5	8027
4	1160	1290.5	9273	944	461.0	8028	1111	1666.5	9491	1063	1126.5	9224	980	790.0	7894
5	1160	1245.5	9289	957	353.5	8038	1117	1600.5	9425	1072	1061.0	9342	981	757.0	7927
6	1161	1279.5	9262	957	465.0	7903	1121	1504.5	9329	1085	1081.0	9333	982	792.5	7754
7	1168	**1227.0** *	9310	959	416.0	7830	1122	1533.5	9091	1095	1125.0	9325	990	798.5	7737
8	1196	1242.5	9132	960	404.0	7805	1139	1393.0	9270	1141	900.0	9131	1000	781.0	7840
9	1197	1242.0	**9128** *	979	389.5	8026	1146	1399.5	9186	1225	**886.0** *	**9117** *	1003	**657.5** *	7688
10				1010	521.0	**7800** *	1154	1379.5	9166				1037	740.0	7645
11				1017	**317.5** *	8014	1164	1432.5	9080				1045	731.0	**7636** *
12							1170	1414.5	9088						
13							1228	**1287.5** *	**8955** *						

	ORB6			ORB7			ORB8			ORB9			ORB10		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1058** *	724.5	**8797** *	**413** *	143.0	3646	**948** *	1498.5	8317	**978** *	1637.0	9373	**974** *	875.5	8921
2	1114	**703.5** *	8931	417	111.0	3544	966	1373.5	8144	979	1470.0	9232	976	767.0	8318
3	1118	711.0	8872	424	144.5	3538	1003	1311.0	7925	984	1419.0	9126	979	758.0	8420
4				445	164.5	**3525** *	1003	1304.0	7957	986	1514.0	9121	979	744.0	8472
5				445	**110.0** *	3633	1004	1257.0	7910	988	1248.0	8956	1021	692.0	8744
6							1006	1236.0	7889	992	1223.5	8950	1029	752.0	8395
7							1010	1258.0	7872	1003	1298.5	8826	1031	601.0	8467
8							1021	**1233.0** *	**7847** *	1006	1196.5	8899	1058	535.0	8409
9										1006	1191.0	8922	1059	511.0	8385
10										1008	1178.5	8905	1068	**483.5** *	**8268** *
11										1011	1173.5	8892			
12										1017	1140.0	8816			
13										1020	1271.5	8801			
14										1021	1140.0	8610			
15										1029	1076.0	8841			
16										1029	1100.5	8765			
17										1040	1084.0	8787			
18										1048	932.5	8627			
19										1057	1123.5	**8581** *			
20										1058	1117.5	8607			
21										1112	923.0	8694			
22										1140	**920.5** *	8674			
23										1180	960.5	8625			

Table A15. Non-dominated solutions obtained by SS/CMOTA for the JSSP instances proposed by [38].

	LA01			LA02			LA03			LA04			LA05		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**666** *	1302.5	5562	**655** *	1230.5	5180	**614** *	1741.5	5301	**595** *	1212.0	4915	**593** *	1177.5	4520
2	666	1525.5	5536	665	1284.5	5135	619	1641.5	5137	598	1196.0	4880	593	1218.0	4519
3	668	1178.5	5426	668	1015.0	4883	622	1582.0	5137	610	1150.0	4846	595	1186.5	4510
4	668	1276.5	5409	681	1003.0	4947	623	1489.5	5001	612	1182.0	4775	598	1148.5	4517
5	670	1112.0	5220	686	1168.0	4851	625	1536.0	4968	618	1106.0	4809	598	1199.0	4443
6	740	1093.0	5283	687	924.5	4839	627	1478.0	4984	623	1020.5	4695	600	1125.0	4426
7	764	**1068.5** *	**5182** *	687	981.0	4758	627	1471.0	5034	642	945.5	4620	600	1172.0	4416
8				698	922.0	4806	628	1435.5	4952	663	**906.5** *	**4581** *	607	1114.5	4539
9				720	948.5	**4696** *	630	1451.5	4897				607	1115.5	4528
10				729	928.5	4778	630	1451.0	4902				610	1102.5	4433
11				831	918.0	4799	632	1365.5	4837				613	1095.5	4442
12				834	925.0	4736	633	1161.5	4657				621	1181.0	4413
13				855	**850.0** *	4716	645	1151.0	4625				647	1077.5	4458
14							651	1136.5	4711				648	**1071.5** *	4479
15							658	1115.5	4611				649	1124.5	4432
16							669	1183.0	4588				649	1072.5	4440
17							671	1122.5	4594				654	1124.5	4353 *

Table A15. *Cont.*

		LA01			LA02			LA03			LA04			LA05	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
18							672	995.5	4542						
19							699	**927.0** *	4442						
20							701	1012.5	**4421** *						
21							766	940.0	4441						
		LA06			**LA07**			**LA08**			**LA09**			**LA10**	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**926** *	3995.0	9983	**890** *	3930.5	9547	**864** *	4293.5	10,031	**951** *	4358.0	10,543	**958** *	4465.0	10,495
2	926	4013.5	9965	898	3949.0	9523	864	4350.0	9986	951	4285.5	10,680	959	4464.0	10,494
3	932	4020.5	9886	904	3823.0	9396	865	3852.5	9590	952	4326.0	10,594	971	4424.0	10,419
		LA06			**LA07**			**LA08**			**LA09**			**LA10**	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
4	933	3971.5	9919	905	3812.0	9385	866	3777.5	9515	954	**4029.0** *	**10,358** *	981	4404.0	10,399
5	938	3857.0	9845	910	3825.0	9365	871	3778.0	9511				986	4392.0	10,387
6	939	3920.5	**9763** *	912	3785.5	9401	873	3759.5	9497				994	4330.0	10,360
7	942	**3793.0** *	9781	927	**3720.0** *	9286	874	3735.5	9379				1042	4266.0	10,296
8				974	3765.5	**9268** *	878	**3520.5** *	**9258** *				1046	**4246.0** *	10,276
9													1052	4299.5	**10,269** *
		LA11			**LA12**			**LA13**			**LA14**			**LA15**	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**1222** *	8996.0	16,945	**1039** *	7090.0	14,104	**1150** *	8297.0	16,076	**1292** *	9616.0	17,635	**1207** *	9033.0	17,154
2	1241	**8911.5** *	**16,938** *	1042	7066.0	14,080	1158	8291.0	16,032	1294	**9583.0** *	**17,602** *	1213	8905.5	16,962
3				1050	**6637.0** *	**13,626** *	1166	8197.0	15,976				1214	8703.5	**16,818** *
4							1169	**8019.0** *	**15,798** *				1260	8684.5	16,832
5													1261	**8659.0** *	16,826
		LA16			**LA17**			**LA18**			**LA19**			**LA20**	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**979** *	673.5	8402	**793** *	781.0	7553	**870** *	410.5	7735	**875** *	286.0	7914	**912** *	578.0	8217
2	1017	632.5	8493	795	768.0	7534	880	476.5	7671	875	337.0	7869	914	457.5	8146
3	1024	622.5	8490	795	798.0	7518	882	333.5	7773	875	263.0	7929	914	490.5	8055
4	1025	566.0	8452	800	763.0	7431	883	298.5	7766	880	192.0	7720	916	472.5	8077
5	1028	629.5	8376	801	738.5	7537	883	396.5	7645	888	176.0	7783	917	512.0	8014
6	1038	269.0	8090	802	733.0	7384	885	310.5	7750	891	188.5	7634	918	320.0	8140
7	1050	284.5	**8070** *	802	686.5	7450	886	294.5	7724	894	**166.0** *	7738	920	287.0	8015
8	1076	**228.0** *	8080	802	671.5	7457	887	393.5	7717	897	182.5	7698	926	293.0	8002
9				804	623.5	7126	887	427.5	7564	935	193.5	**7626** *	926	309.5	7943
10				811	623.0	7244	905	440.5	7547				932	385.0	7935
		LA16			**LA17**			**LA18**			**LA19**			**LA20**	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
11				813	548.0	7173	909	383.0	7675				934	343.0	7891
12				821	616.5	7115	914	264.5	7733				940	283.0	8053
13				821	622.5	7024	915	248.5	7707				940	301.0	7966
14				829	654.5	6999	916	320.0	**7487** *				943	347.5	7888
15				846	647.5	7009	934	277.0	7513				958	282.0	8011
16				851	587.5	**6997** *	936	276.5	7706				962	319.0	**7836** *
17				851	496.5	7242	945	274.0	7506				973	294.5	7976
18				881	583.5	7167	972	243.0	7633				979	280.0	8065
19				941	**474.0** *	7233	972	228.0	7745				996	**237.5** *	8010
20							979	**196.5** *	7734						
21							993	234.0	7555						

Table A16. Non-dominated solutions obtained by SS/CMOTA for the JSSP instances proposed by [38].

		LA21			LA22			LA23			LA24			LA25	
	MKS	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**	**MKS**	**TDS**	**FLT**
1	**1103** *	**2088.5** *	**13,708** *	**993** *	2598.0	13,518	**1035** *	1738.5	13,672	**1000** *	2168.5	13,581	**1047** *	3160.5	14,250
2				995	2574.0	13,492	1108	1830.0	**13,651** *	1009	2156.5	13,569	1053	3107.5	14,250
3				999	2567.0	13,550	1122	**1612.5** *	13,707	1011	2135.5	13,591	1055	2846.0	13,843
4				1004	2514.0	13,373				1013	2135.5	13,548	1059	2754.5	13,768
5				1013	2524.0	13,359				1013	2115.5	13,584	1066	2630.5	13,850
6				1018	2422.0	13,340				1015	2134.5	13,528	1068	2617.5	13,789
7				1041	2331.5	13,081				1016	**1866.0** *	**13,224** *	1072	2616.5	13,753
8				1041	2325.0	13,132							1074	2713.5	13,732
9				1045	**2179.0** *	**13,040** *							1079	2689.5	13,696
10													1081	2489.5	13,652
11													1082	2354.0	13,474
12													1128	**2306.5** *	13,564
13													1216	2520.0	**13,471** *

Table A16. *Cont.*

		LA26			LA27			LA28			LA29			LA30	
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1237** *	6743.5	22,466	**1290** *	**5765.5** *	**21,988** *	**1279**	6587.0	22,573	**1261** *	7052.0	21,877	**1390** *	8350.5	24,343
2	1243	6722.5	22,445				1284	6445.0	22,468	1263	7023.0	21,844	1398	8278.5	24,271
3	1245	6103.5	21,867				1286	6340.0	22,363	1276	6974.0	21,817	1402	8225.5	24,218
4	1281	6238.5	21,822				1289	6205.0	22,228	1296	6587.5	21,312	1404	8004.0	24,024
5	1297	5973.5	21,670				1291	**6107.5** *	**22,099** *	1315	**6334.0** *	**21,155** *	1409	7978.0	23,998
6	1303	**5934.5** *	**21,631** *										1433	8189.5	23,994
7													1434	7791.0	23,811
8													1439	7738.0	23,758
9													1442	**7735.0** *	**23,755** *

		LA31			LA32			LA33			LA34			LA35	
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1786** *	22,019.5	44,806	**1850** *	20,717.0	45,507	**1719** *	19,800.5	42,156	**1721** *	20,276.5	43,288	**1888** *	**20,980.5** *	**44,156** *
2	1787	21,998.5	44,785	1852	20,531.5	45,385	1721	19,732.5	42,186	1724	**20,265.5** *	**43,277** *			
3	1789	21,274.5	44,061	1873	**20,525.5** *	**45,379** *	1723	19,751.5	42,144						
4	1790	20,568.5	43,355				1724	19,672.5	42,126						
5	1810	**20,337.5** *	**43,124** *				1725	19,328.5	41,782						
6							1726	19,068.5	41,424						
7							1729	19,057.5	41,413						
8							1731	**19,050.5** *	**41,406** *						

Table A17. Non-dominated solutions obtained by SS/CMOTA for the JSSP instances proposed by [38].

		LA36			LA37			LA38			LA39			LA40	
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1367** *	2046.0	19,295	**1533** *	2291.0	20,930	**1323** *	1640.0	17,834	**1328** *	1036.5	17,585	**1318** *	1024.5	**17,718** *
2	1369	2016.5	19,362	1536	1919.5	20,430	1329	1509.0	17,538	1366	1031.5	17,512	1326	**944.0** *	17,768
3	1377	2012.0	19,249	1548	1828.0	20,435	1331	1498.0	17,878	1368	904.5	17,426			
4	1378	1768.0	19,164	1553	1830.0	20,433	1345	1473.5	17,653	1368	961.5	**17,422** *			
5	1380	1897.0	19,108	1554	1827.0	20,430	1357	1457.5	**17,459** *	1379	**888.0** *	17,527			
6	1394	1684.5	**19,003** *	1567	1926.0	20,294	1373	1450.0	17,836	1392	893.0	17,440			
7	1409	1675.0	19,041	1571	1919.0	20,287	1376	**1426.0** *	17,778						
8	1473	**1667.0** *	19,134	1577	1986.0	20,218									
9	1478	1674.5	19,048	1581	1769.0	20,605									
10				1581	1797.0	20,289									
11				1584	1550.5	20,032									
12				1587	**1454.5** *	19,923									
13				1648	1676.5	**19,800** *									

Table A18. Non-dominated solutions obtained by SS/CMOTA for the JSSP instances proposed by [39].

		ABZ5			ABZ6			ABZ7			ABZ8			ABZ9	
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1239** *	241.5	**10,829** *	**961** *	190.5	8426	**742** *	2425.5	13,462	**751** *	2604.5	13,961	**780** *	2968.0	13,900
2	1250	150.0	11,038	970	177.0	8416	745	2395.5	13,441	751	2607.5	13,954	783	2665.5	13,798
3	1269	**110.0** *	10,852	976	137.0	8464	755	2502.0	13,346	755	2566.0	13,904	787	2562.0	13,622
4				981	115.0	8427	757	2126.5	13,113	762	2533.5	13,903	789	2538.0	13,598
5				981	171.5	8288	768	2125.5	13,112	763	2524.5	13,894	790	2537.0	13,611
6				987	74.0	8382	768	2114.0	13,126	774	2326.0	13,702	791	2492.0	13,621
7				994	59.0	8342	774	2100.0	13,112	781	2258.5	13,492	794	2434.0	13,491
8				998	149.0	8235	777	**2097.5** *	**13,077** *	782	2256.5	13,518	827	**2235.5** *	13,384
9				1007	144.0	8307				784	2247.5	13,475	839	2443.0	**13,358** *
10				1014	122.5	8333				785	2120.0	13,406			
11				1030	185.5	**8217** *				786	**2105.0** *	**13,391** *			
12				1066	**39.5** *	8390									

Table A19. Non-dominated solutions obtained by SS/CMOTA for the JSSP instances proposed by [40].

		YN01			YN02			YN03			YN04	
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1036** *	1582.0	18,928	**1063** *	1757.0	19,058	**1037** *	1635.0	18,813	**1133**	2375.5	19,979
2	1036	1593.0	18,907	1084	1709.5	19,133	1051	1579.5	18,876	1155	2373.0	20,359
3	1041	1252.5	18,619	1087	1685.0	19,084	1056	1691.5	18,591	1157	2247.0	20,236

Table A20. Non-dominated solutions obtained by SS/CMOTA for the JSSP instances proposed by [38].

	YN01			YN02			YN03			YN04		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
4	1051	1427.5	18,468	1095	**1643.0** *	**19,042** *	1057	1455.0	18,786	1163	2258.0	**19,898** *
5	1058	1142.5	18,215				1058	1619.5	18,519	1177	**2244.5** *	20,223
6	1059	1144.5	18,188				1064	1512.0	18,656			
7	1068	1090.5	18,208				1072	1494.5	18,419			
8	1072	1091.5	**18,164** *				1081	1481.5	18,406			
9	1092	**1055.5** *	18,254				1082	1463.5	18,406			
10							1089	1421.5	18,331			
11							1093	**1364.5** *	**18,289** *			

Table A21. Non-dominated solutions obtained by SS/CMOTA for the JSSP instances proposed by [41].

	TA01			TA11			TA21			TA31			TA41		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**1330** *	1620.0	18,799	**1550** *	4800.5	26,342	**1910** *	5510.5	35,751	**1978** *	18,629.0	52,529	**2375** *	**16,031.5** *	**62,497** *
2	1335	1557.0	18,483	1552	4752.0	26,321	1911	4527.0	34,593	1980	18,479.0	52,379			
3	1352	1504.0	18,649	1553	4639.0	26,208	2071	4701.0	34,566	1987	18,437.0	52,337			
4	1366	1407.0	18,478	1568	**4635.0** *	**26,204** *	2075	4633.0	34,575	1999	18,338.0	52,238			
5	1371	1516.0	18,403				2086	4451.0	**34,186** *	2007	18,036.0	51,936			
6	1374	1369.5	18,603				2096	**4365.5** *	34,188	2028	17,872.0	51,772			
7	1377	1330.5	18,467							2134	**17,768.0** *	**51,668** *			
8	1378	1220.0	18,078												
9	1381	1385.5	18,066												
10	1382	1369.5	18,050												
11	1389	1219.0	18,084												
12	1396	1285.5	**17,972** *												
13	1408	1184.5	18,075												
14	1413	1180.5	18,048												
15	1417	1251.5	18,000												
16	1421	**1170.5** *	18,068												

	TA51			TA61			TA71		
	MKS	TDS	FLT	MKS	TDS	FLT	MKS	TDS	FLT
1	**3053** *	68,059.0	124,902	**3225** *	**66,175.0** *	**142,621** *	**5814** *	345,366.5	496,703
2	3062	67,494.0	124,335				5815	344,999.5	496,336
3	3067	66,576.0	123,417				5830	**344,914.5** *	**496,251** *
4	3069	66,551.0	123,392						
5	3070	**66,462.0** *	**123,303** *						

References

1. Garey, M.R.; Johnson, D.S.; Sethi, R. PageRank: The complexity of flowshop and jobshop scheduling. *Math. Oper. Res.* **1976**, *1*, 117–129. [CrossRef]
2. Pinedo, M. *Scheduling Theory Algorithm, and Systems*, 5th ed.; Springer: Berlin/Heidelberg, Germany; New York, NY, USA, 2016; ISBN 978-3-319-26578-0.
3. Yang, Y.B. Methods and Techniques Used for Job Shop Scheduling, MSc. Research Project, Florida Technological University. 1972. Available online: https://stars.library.ucf.edu/cgi/viewcontent.cgi?article=1389&context=rtd (accessed on 29 November 2021).
4. Xing, L.N.; Chen, Y.W.; Yang, K.W. Multi-objective flexible job shop schedule: Design and evaluation by simulation modeling. *Appl. Soft Comput.* **2009**, *9*, 362–376. [CrossRef]
5. Yuan, Y.; Xu, H. Multiobjective flexible job shop scheduling using memetic algorithms. *IEEE Trans. Autom. Sci. Eng.* **2015**, *12*, 336–353. [CrossRef]
6. Kaplanoglu, V. An object-oriented approach for multi-objective flexible job-shop scheduling problem. *Expert Syst. Appl.* **2016**, *45*, 71–84. [CrossRef]
7. Bao, Q.; Soubeyran, A. Variational principles in set optimization with domination structures and application to changing jobs. *J. Appl. Numer. Optim.* **2019**, *1*, 217–241.
8. Luu, D.V.; Linh, P.T. Optimality and duality for nonsmooth multiobjective fractional problems using convexificators. *J. Nonlinear Funct. Anal.* **2021**, *2021*, 1.
9. Pereira, O.; Júnior, V.L.d.; Soubeyran, A. Inexact Multi-Objective Local Search Proximal Algorithms: Application to Group Dynamic and Distributive Justice. *J. Optim. Theory Appl.* **2018**, *177*, 181–200.
10. Glover, F. Heuristics for integer programming using surrogate constraints. *Decis. Sci.* **1977**, *8*, 156–166. [CrossRef]

11. Frausto-Solis, J.; Hernández-Ramírez, L.; Castilla-Valdez, G.; González-Barbosa, J.; Sánchez, J. Chaotic multi-objective simulated annealing and threshold accepting for job shop scheduling problem. *Math. Comput. Appl.* **2021**, *26*, 8. [CrossRef]
12. Suresh, R.K.; Mohanasundaram, M. Pareto archived simulated annealing for job shop scheduling with multiple objectives. *Int. J. Adv. Manuf. Technol.* **2006**, *29*, 184–196. [CrossRef]
13. Bandyopadhyay, S.; Saha, S.; Maulik, U.; Deb, K. A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA. *Evol. Comput. IEEE Trans.* **2008**, *12*, 269–283. [CrossRef]
14. Deb, K.; Agrawal, S.; Pratap, A.; Meyarivan, T. A fast elitist non-dominated sorting genetic algorithm for multiobjective optimization: NSGA-II. In *International Conference on Parallel Problem Solving from Nature*; Spring: Berlin/Heidelberg, Germany, 2000; Volume 1917.
15. Kachitvichyanukul, V.; Sitthitham, S. A two-stage genetic algorithm for multiobjective job shop scheduling problems. *J. Intell. Manuf.* **2009**, *22*, 355–365. [CrossRef]
16. Zhao, F.; Chen, Z.; Wang, J.; Zhang, C. An improved MOEA/D for multiobjective job shop scheduling problem. *Int. J. Comput. Integr. Manuf.* **2016**, *30*, 616–640. [CrossRef]
17. Zitzler, E.; Deb, K.; Thiele, L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evol. Comput.* **2000**, *8*, 173–195. [CrossRef]
18. Karimi, N.; Zandieh, M.; Karamooz, H. Bi-objective group scheduling in hybrid flexible flowshop: A multi-phase approach. *Expert Syst. Appl.* **2010**, *37*, 4024–4032. [CrossRef]
19. González, M.; Oddi, A.; Rasconi, R. Multiobjective optimization in a job shop with energy costs through hybrid evolutionary techniques. In Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling, Pittsburgh, PA, USA, 18–23 June 2017; pp. 140–148.
20. Fisher, H.; Thompson, G.L. Probabilistic learning combinations of local job-shop scheduling rules. *Ind. Sched.* **1963**, *1*, 225–251.
21. Méndez-Hernández, B.; Rodriguez Bazan, E.D.; Martinez, Y.; Libin, P.; Nowe, A. A Multiobjective Reinforcement Learning Algorithm for JSSP. In Proceedings of the 28th International Conference on Artificial Neural Networks, Munich, Germany, 17–19 September 2019; pp. 567–584. [CrossRef]
22. Zhang, J.; Ding, G.; Zou, Y.; Qin, S.; Fu, J. Review of job shop scheduling research and its new perspectives under Industry 4.0. *J. Intell. Manuf.* **2019**, *30*, 1809–1830. [CrossRef]
23. Deb, K. *Multiobjective Optimization Using Evolutionary Algorithms*; Wiley: New York, NY, USA, 2001.
24. Coello, C.; Veldhuizen, D.; Lamont, G. *Evolutionary Algorithms for Solving Multiobjective Problems*, 2nd ed.; Springer: Berlin, Germany, 2007; ISBN 978-0-387-36797-2.
25. Okabe, T.; Jin, Y.; Sendhoff, B. A critical survey of performance indices for multiobjective optimisation. In *Proceedings of the 2003 Congress on Evolutionary Computation, 2003. CEC '03, Canberra, ACT, Australia, 8–12 December 2003*; IEEE: Piscataway, NJ, USA, 2003; Volume 2, pp. 878–885.
26. Schott, J.R. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.
27. Veldhuizen, D.A.V. Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. Ph.D. Thesis, Air Force Institute of Technology, Wright-Patterson AFB, Dayton, OH, USA, 1999.
28. Coello, C.; Cruz, N. Solving Multiobjective Optimization Problems Using an Artificial Immune System. *Genet. Program. Evolvable Mach.* **2005**, *6*, 163–190. [CrossRef]
29. Sawaragi, Y.; Nakagama, H.; Tanino, T. *Theory of Multiobjective Optimization*; Springer: Boston, MA, USA, 1985.
30. Bakuli, D.L. A Survey of Multiobjective Scheduling Techniques Applied to the Job Shop Problem (JSP). In *Applications of Management Science: In Productivity, Finance, and Operations*; Emerald Group Publishing Limited: Bingley, UK, 2015; pp. 51–62.
31. Baker, K.R. Sequencing rules and due-date assignments in job shop. *Manag. Sci.* **1984**, *30*, 1093–1104. [CrossRef]
32. Yazid, M.; Dauzère-Pérès, S.; Chams, L. A general approach for optimizing regular criteria in the job-shop scheduling problem. *Eur. J. Oper. Res.* **2011**, *212*, 33–42. [CrossRef]
33. Dueck, G.; Scheuer, T. Threshold Accepting: A General Purpose Algorithm Appearing Superior to Simulated Annealing. *J. Comput. Phys.* **1990**, *90*, 161–175. [CrossRef]
34. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. Am. Assoc. *Adv. Sci.* **1983**, *220*, 671–680.
35. Sanvicente, S.H.; Frausto, J. A method to establish the cooling scheme in simulated annealing like algorithms. In Proceedings of the International Conference on Computational Science and Its Applications, Assisi, Italy, 14–17 May 2004; pp. 755–763.
36. May, R. Simple Mathematical Models with Very Complicated Dynamics. *Nature* **1976**, *26*, 457. [CrossRef]
37. Adams, J.; Balas, E.; Zawack, D. The shifting bottleneck procedure for job shop scheduling. *Manag. Sci.* **1988**, *34*, 391–401. [CrossRef]
38. Applegate, D.; Cook, W. A computational study of the job-shop scheduling problem. *ORSA J. Comput.* **1991**, *3*, 149–156. [CrossRef]
39. Lawrence, S. *Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques (Supplement)*; Graduate School of Industrial Administration, Carnegie-Mellon University: Pittsburgh, PA, USA, 1984.
40. Yamada, T.; Nakano, R. A genetic algorithm applicable to large-scale job-shop problems. In Proceedings of the Second International Conference on Parallel Problem Solving from Nature, Brussels, Belgium, 28–30 September1992; pp. 281–290.
41. Taillard, E. Benchmarks for basic scheduling problems. *Eur. J. Oper. Res.* **1993**, *64*, 278–285. [CrossRef]