Hybridization Simulated Annealing Algorithm in a Single Machine Scheduling Problem

HAFED M. MOTAIR
Department of Mathematics, Distinguished Secondary School
Ministry of Education, Diwaniya, IRAQ

Abstract: In this paper, we investigate a single machine scheduling problem (SMSP). We try to reach the optimal or near optimal solution which minimize the sum of three objective functions: total completion times, total tardiness and total earliness. Firstly, we solve this problem by Branch and bound algorithm (BAB alg) to find optimal solutions, dominance rules (DR)s are used to improve the performance of BAB alg, the resulting is BABDR, secondly, we solve this problem by simulated annealing algorithm (SA alg) as metaheuristic algorithm (MET alg). It is known that combining MET alg with other algorithms can improve the resulting solutions. In this paper we developed the concept of insertion preselected jobs one by one through all positions of remaining jobs of considered sequence, the proposed MET alg called Insertion Metaheuristic Algorithm (IMA). This procedure improves the performance of SA alg in two directions: in the first one, we use the IMA to generate initial solution for SA alg, in the second one, we use the IMA to improve the solution obtained through the iterations of SA alg. The experiments showed that IMA can improve the performance of SA alg in these two directions.

Key-Words:- Hybrid, Single Machine, Branch and bound, Dominance rules, Simulated annealing, Metaheuristics.
Received: May 3, 2021. Revised: October 11, 2021. Accepted: October 26, 2021. Published: November 12, 2021.

1 Introduction

In this work, we consider the problem (Q) of minimization the sum of three objective functions: total completion times ($\sum_i C_i$), total tardiness ($\sum_i T_i$) and total earliness ($\sum_i E_i$) in single machine framework, the objective is to reach the optimal solution for the problem (Q). In view of the fact that the problem has been considered as NP-hard problem because the minimization of $\sum_i T_i$ in single machine problem is NP-hard [1]. Most of the works of researchers in scheduling for many years focused on a single (objective) performance measure, the focused works are on Bi-objective or Tri-objective scheduling problems. Using multi-objective against single objective in fact, makes the scheduling problem more realistic. In single objective problems, only one schedule's performance aspect is considered, and the others are not, where the multi-objective scheduling consider more than one aspect so that tradeoffs between conflicting objectives can be achieved. Due to the hardness of the problem, it is very hard to solve the problem (Q) by exact methods in which all possible solutions are considered to find the optimal one. These methods guarantee finding the best solution, but the computational times are exponentially increase. Another alternative search strategies are the use of heuristic (metaheuristic) methods. Various MET alg were proposed in literature: SA alg [2], Tabu search algorithm [3], Iterated local search [4], variable neighborhood search [5] etc. Although these methods offer good results, they do guarantee to reach optimal solutions. In the other hand, using MET alg alone can rather restrictive for advancing optimization problems [6]. Therefore, researchers tended to combine MET alg with other algorithms to obtain more efficient algorithms especially for large sized problems called hybrid MET alg. [7], combined an iterated local search algorithm with an evolutionary algorithms and then the comparison made with to local search algorithms proposed in the literature. [8] hybridize GRASP with evolutionary path relinking to find approximate solution for maximum diversity problem (MMDP). [9] hybridize GA algorithm with Tabu search algorithm to find the best locations for installing back to back (BtB) converters in a power grid to decrease fault current levels. [10] proposed a hybrid algorithm based on CS and GSA algorithms, the objective is to develop the exploration capability of gravitational search algorithm. Simulation results show that the proposed algorithm better solutions than both CS and GSA algorithms. [11] hybridize the IWSSr method and Shuffled Frog Leaping Algorithm (SFLA), the objective is to reach effective features in a large-scale gene dataset, the

results show effectiveness of the combination of the two algorithms.

In this work, we propose an insertion heuristic algorithm (IMA) which is use the idea of NEH algorithm (NEH alg.) [12], then it is combined with SA alg. The IMA procedure uses the idea of insertion of preselected jobs and insert these jobs one by one to all positions of the remaining jobs, the development of this idea is by using this idea through several runs and at each run the neighborhood used to perturbate the current sequence. With the simplicity of the algorithm, its performance was reasonable, especially if it is combined with another algorithm such as SA alg. Several approaches in literature proposed a hybrid SA alg. [13] combined two algorithms which are genetic algorithm and cross entropy algorithm with the SA alg, and the proposed algorithm compared with NSGA-II and GA-SA algorithms. [14] embedded SA alg in a whale optimization algorithm (WOA), the experimental results showed the effectiveness of the proposed algorithm. [15] proposed a hybrid SA alg and reduced variable neighborhood search to find the near optimal solutions in a mixed-integer linear programming formulation, experimental results on a large set of benchmarks demonstrate the efficiency of the proposed algorithm. [16] combine SA alg with ant colony optimization (ACO) for dynamic traveling salesman problem, the results compared with four other MET algs and showed that the proposed algorithms out performed these algorithms. [17] combined a genetic-algorithm and SA alg for prediction of the ultimate bearing capacity of the pile. [18] combined an atom search optimization with SA alg. [19] proposed a hybrid bio inspired clustering routing protocol using Cuckoo Search and SA alg..

The rest of this paper is organized as follows: In section 2 we discuss the problem formulation and state some notations and definitions. Section 3 include the proposed BAB alg and dominance rule, while section 4 introduced proposed MET alg. Computational experiments (results and discussions) are included in section5. Conclusion and some recommendations are placed in section 6.

2 Problem Formulation and Some Notations.

We investigate a SMSP with a set of n jobs and considering the following assumption:

- Processing times contains the set-up times.
- All jobs sequence is ready at time zero.
- No Precedence relationships between jobs
- Preemption is not allowed.
- The idle time of machine is not allowed.

The notations used to describe the scheduling problems:

- n : Is the number of jobs to be processed.
- p_i: "Processing time" of the job i
- d_i: "Due date" of the job i
- J_i : Is the job in the $i\ th$ position.
- S_i "Slack time" of the $i\ th$ job, $S_i = d_i - p_i$.
- C_i: "Completion time" of job i.
- T_i: "Tardiness" of job i, $T_i = \max(C_i - d_i, 0)$.
- E_i: "Earliness" of job i, $E_i = \max(d_i - C_i, 0)$.

Let $\beta = (J_1, J_2, \dots, J_n)$ be a schedule of n jobs, then:
$\sum_j C_j$ is the total completion times.
$\sum_j T_j$ is the total tardiness.
$E_{max} = \max(E_i) = \max(d_i - C_i, 0)$.
The objective is to minimize the function $g(\beta)$ of the sum of the three objective functions:
$g(\beta) = \sum_j C_j(\beta) + \sum_j T_j(\beta) + \sum_j E_j(\beta)$.
The mathematical form of the problem (Q) can be written as follows:
$g(B) = \min(\sum_j C_j + \sum_j T_j + \sum_j E_j)$.
s.t
$C_j \geq p_j,\ j = 1,2, \dots, n$
$C_j = C_{j-1} + p_j,\ \ j = 2,3, \dots, n.$
$T_j \geq C_j - d_j,\ \ j = 1,2, \dots, n.$
$T_j \geq 0,\ \ j = 1,2, \dots, n.$
$E_j \geq d_j - C_j,\ \ j = 1,2, \dots, n.$
$E_j \geq 0, j = 1,2, \dots, n.$
According to the three field notation, the problem (Q) can be written: $1\ |\ |\sum_j C_j + \sum_j T_j + \sum_j E_j$.

3 Proposed BAB algorithm

Finding optimal solution for NP-hard discrete optimization problems needs to use very efficient algorithms. One of the main tools to solve these problems is the BAB alg. In order to use the BAB alg and to avoid exponentially increasing number of potential solutions, the BAB alg use two types of bounds (lower and upper bounds) for the optimized function combined with the value of the current best solution which enables the algorithm to search parts of the solution space [20]. To implement BAB alg, the problem (Q) can be decomposed into three sub-problems Q_i, the first sub-problem is : Q_1: $1||Z_1$, the second sub-problem is Q_2: $1||Z_2$ and the third sub-problem is Q_3: $1||Z_3$, where $Z_1 = min \sum_j C_j$, , $Z_2 = min \sum_j T_j$, and $Z_3 = \sum_j E_j$.

To construct upper and lower bounds, we start with the following basic definitions:

Definition(1): Shortest Processing Times (SPT) [21]**:** the jobs are sorted in non-decreasing order of job processing times (p_i).

Definition(2): Earliest Due Dates (EDD) [22]**:** The sequence ordered in non-decreasing of jobs due dates (d_i).

Definition(3): Minimum slack time (MST) [23]**:** The problem $1||E_{max}$ can be minimized by sorting the jobs sequence in non-decreasing of slack times $s_j = d_j - p_j$.

These rules are used to compute the first and second upper bounds as follows:

$UR1 = \sum_j C_j\,(SPT) + \sum_j T_j\,(SPT) + \sum_j E_j\,(SPT)$

$UR2 = \sum_j C_j\,(EDD) + \sum_j T_j\,(EDD) + \sum_j E_j\,(EDD)$

And the upper bound (UR) for problem (Q) is the minimum value of these bounds.

To construct the lower bound for problem (Q), we use the following theorem:

Theorem (1): Suppose M_1, M_2, M_3 be the lower bounds for the sub-problem Q_1, Q_2, Q_3 respectively and let M be the lower bound of the problem (Q), then $M \geq M_1 + M_2 + M_3$.

Proof: Suppose the schedule β be the optimal of problem (Q), then $M = M_1(\beta) + M_2(\beta) + M_3(\beta)$, on the other hand the schedule β is a feasible for each of sub-problems: Q_i, $i = 1,2,3$.

Then $M_i(\beta) \geq M_i\,, \forall\, i = 1,2,3$.

Then $M = M_1(\beta) + M_2(\beta) + M_3(\beta) \geq M_1 + M_2 + M_3$ Hence $M \geq M_1 + M_2 + M_3$. ■

We use the following initial lower bound (ILB): $ILB = \sum_j C_j\,(\rho_1) + \sum_j T_j\,(\rho_2) + \sum_j E_j\,(\rho_3)$, where ρ_1 is the jobs, sequence obtained by SPT rule to minimize the objective function $\sum_j C_j$. ρ_2 is the jobs sequence obtained by EDD rule to get the minimum objective function T_{max} and then use the relation $T_{max}(\rho_2) \leq \sum_j T_j$. ρ_3 is the jobs sequence obtained by MST rule to get the minimum objective function E_{max} and then use $E_{max}(\rho_3) \leq \sum_j E_j$. The following theorem help to reduce the search space by using the DR:

Theorem 2: If $p_i \leq p_j$ and $d_i \leq d_j$ for each job i and j from 1 to n, then job (i) precede job (j) in optimal solution when solving problem (Q) .

Proof: Suppose we have a sequence $\beta = \beta_1 i\, j\, \beta_2$ and let $\acute{B} = B_1\, ji\, B_2$ be a sequence obtained by interchange the position of jobs i and j.

We have two cases for the sequence β and β́:

Case 1: If $p_i \leq p_j$ and $d_i \leq d_j$ implies $s_i \leq s_j$ for every $i, j = 1,2, \dots, n$

From $p_i \leq p_j$ we have:$\sum_k C_k(\beta) \leq \sum_k C_k(\acute{\beta})$

From the condition of slack time $s_i \leq s_j$, we have$\sum_k E_k\,(\beta) \leq \sum_k E_k\,(\acute{\beta})$.

From $p_i \leq p_j$ and $d_i \leq d_j$, we have $\sum_k T_k\,(\beta) \leq \sum_k T_k\,(\acute{\beta})$.

Hence, we have: $\sum_k C_k(\beta) + \sum_k T_k(\beta) + \sum_k E_k\,(\beta) \leq \sum_k C_k(\acute{\beta}) + \sum_k T_k(\acute{\beta}) + \sum_k E_k\,(\acute{\beta})$.

Case 2: If $p_i \leq p_j$ and $d_i \leq d_j$ implies $s_i > s_j$ for every $i, j = 1,2, \dots, n$

From $p_i \leq p_j$ we have:

$\sum_k C_k(\beta) \leq \sum_k C_k(\acute{\beta})$ (1)

Equation (1) satisfied by the condition on processing times, and the addition in cost which is obtained from (1) is equal to $p_j - p_i$, this gives:

$\sum_k C_k(\beta) + p_j - p_i = \sum_k C_k(\acute{\beta})$. (2)

The slack times condition $s_i > s_j$ implies $\sum_k E_k\,(\beta) > \sum_k E_k\,(\acute{\beta})$.

Also, the addition in cost $s_i - s_j$ gives:

$\sum_k E_k\,(\acute{\beta}) + s_i - s_j = \sum_k E_k\,(\beta)$ (3)

$$s_i - s_j = (d_i - p_i) - (d_j - p_j)$$
$$= (d_i - d_j) + (p_j - p_i)$$
$$\leq p_j - p_i \quad (4)$$

Adding $\sum_k E_k\,(\acute{\beta})$ to both side of (4) we have:

$\sum_k E_k\,(\acute{\beta}) + s_i - s_j \leq \sum_k E_k\,(\acute{\beta}) + p_j - p_i$ and from (3) we have

$\sum_k E_k\,(\beta) \leq \sum_k E_k\,(\acute{\beta}) + p_j - p_i$ (5)

Adding $\sum_k C_k(\beta)$ to both side of (5) and by (2) we have

$\sum_j C_j(\beta) + \sum_k C_k(\beta) \leq \sum_j C_j(\acute{\beta}) + \sum_k E_k\,(\acute{\beta})$. (6)

From the conditions $p_i \leq p_j$ and $d_i \leq d_j$ we have $\sum_k T_k\,(\beta) \leq \sum_k T_k\,(\acute{\beta})$. By adding this result to (6):

$\sum_k C_k(\beta) + \sum_k T_k(\beta) + \sum_k E_k\,(\beta) \leq \sum_k C_k(\acute{\beta}) + \sum_k T_k(\acute{\beta}) + \sum_k E_k\,(\acute{\beta})$.

Hence σ is better than the sequence σ́ in the two cases and a job i proceed job j in the optimal solution. ■

4. Proposed Metaheuristic Algorithms

4.1 Insertion Metaheuristic Algorithm (IMA).

One of the most powerful algorithm that proposed to minimize makespan objective function in two machine permutation flow shop is the NEH alg which proposed by [12]. The algorithm uses the concept of job insertion technique after sorting the considered sequence in descending order of the total processing times. The first two jobs considered as a partial sequence and the other jobs are inserted between the jobs of this partial sequence one by one to obtain a final complete sequence. We use the

idea of NEH alg to develop a new metaheuristic algorithm called Insertion Metaheuristic Algorithm (IMA). The IMA algorithm run three times, in the first run it use the SPT sequence as initial sequence for NEH alg, in the second run it uses the EDD sequence and in the last run it use the MST sequence. In each run (ith run) the NEH alg uses the initial sequences to find the (ith) best solution for problem (Q), and the obtained sequence perturbed using swap neighborhood in which one job removed from one position randomly in a solution and reinserted in another position chosen randomly, the resulting solution then used as new initial solution for NEH alg, this process repeated three times in each run. There are three best solutions resulting in three runs and the final best solution is the best one of them. Figure (1) present the flow chart of IMA alg.

NEH Algorithm.

1. Sort the initial solution in SPT rule.
2. Set W=2, from the obtained solution select the first two jobs, then select the best one that minimize the objective function, and set this solution as the current solution.
3. Set W=W+1, and generate W partial solutions by insert the first job from the selected set of jobs into each position of the current solution, from these solutions choose the best one. Set the obtained partial solution as the new current solution.
4. If W=n go to step (5), otherwise go to step .(3)
5. Stop.

4.2 Combining IMA with SA alg.

The IMA used to improve the performance of SA alg by generate the initial solution (Isol) of SA alg, and the resulted algorithm is HIM-SA (Hybrid Insertion Metaheuristic-SA). Also, the SA alg with initial solution (SPT, EDD, and MST respectively) used to generate the initial solution (Isol) for NEH alg in each of the three runs of IMA, the resulting algorithm is HSA-IM (Hybrid SA-Insertion Metaheuristic) Figure (2). Where F ( ) is the value of the objective function,

SA Algorithm

1. Choose Initial solution Isol
2. Set t_0 Initial temperature
3. Set h_i=Isol, calculate $F(h_i)$
4. Repeat
   i. Perturbation h_i to generate h_j.
   ii. Calculate $F(h_j)$
   iii. if $F(h_i) < F(h_j)$ then $h_i = h_j$
   iv. else
   v. If Exp $(|F(h_i) - F(h_j)|/t_k\ ) < rand\ (0,1)$ then $h_i = h_j$.
   vi. Set $t_k = \alpha\, t_{k-1}$
5. Return h_i and $F(h_i)$

rand (0, 1) is a random number on the open interval (0,1), t_k is a non-negative number called (temperature) of SA alg, α is a physical annealing parameter, Exp ( ) is exponential function value and | | denote the absolute value.

5 Computational Experiments

In this section, tables (1, 2, and 3) contains results of experiments. The processing times sampled from discrete uniform distributions on the interval [1, 99], each job has due dates generated from uniform distribution on $[(1 - T - R/2)P, (1 - T + R/2)P]$ where R and T are hardiness factors belongs to sets: $\{0.2, 0.6, 1.2\}$ and $\{0.2, 0.4, 0.8\}$ respectively, $P = \sum_{j=1}^{n} a_j$. We use MATLAB R2014a program to evaluate all the algorithms which are executed on LENOVO machine Intel (R) Core ™ (i7) CPU @ 2.50 GHz, and 8 GB of RAM. We generate 9 instances for each pair of R and T for each n. For $n \leq 10$, we use the complete enumeration method to generate optimal solutions for problem (Q).

5.1 Parameter Setting

All algorithms are executed on nine instances problem for each n and results reported are the mean values obtained from these instances. For SA alg, initial temperature is set to 10, α (a physical annealing) is set to 0.99, the number of iterations is equal to 2000 iterations when $4 \leq n \leq 15$ and equal to 5000 iterations when $n = 20, 30, 40, 50, 75, 100$.

For IMA, the algorithm runs three times in each run the number of iterations is equal to three.

5.2. Results

From table 1, the results show the efficiency of using DR in BAB alg, the comparison made between two algorithms BAB and BABDR compared with optimal solutions obtained by CEM method for $4 \leq n \leq 15$, the results showed that DR reduce the search space and then computational times when solving problem
(Q), for n=12 BAB one problem not optimal (ex 9).

Also, the results show that BAB alg failed to reach optimal solutions within the 1800 seconds for some problem instances and solved about 81%, while BABDR solved 97 % of all problem instances for $4 \leq n \leq 15$. The mean of computational results and mean values of execution times of SA alg, IMA, HIM-SA, and HSA-IM are summarized in table (2), the results showed that for $4 \leq n \leq 14$ where the BABDR reach optimal solution for problem Q, SA alg and IMA does not give optimal solution in some instances (n=12 for SA alg and $10 \leq n \leq 14$ for IMA). For $15 \leq n \leq 100$, the results shows that the three algorithms SA alg, IMA-SA, and HAS-IM give best solution with the preference of HAS-IM followed by IMA-SA. This mean that when we combined IMA and SA algs, the combined algorithms give better results than the results obtained by each algorithm alone, also we see that the
results obtained when we use the SA alg to improve the best solution obtained by NEH alg in IMA in table 2 shows reasonable computational times for all algorithms with the preference of HAS-IM when

5.3. Discussions

When solving the problem (Q) by BAB alg, the results in table (1) show that using dominance rules can improve the performance of BAB alg but within certain limits of the number of jobs, this calls us to use more flexible algorithms, through which the problem can be solved for a greater number of jobs, for example in this research (100 jobs). We use the so-called hybrid metaheuristics algorithms and propose the IMA procedure combined it with SA alg in two directions. In this paper, we see that the concept of inserting some jobs on one by one on

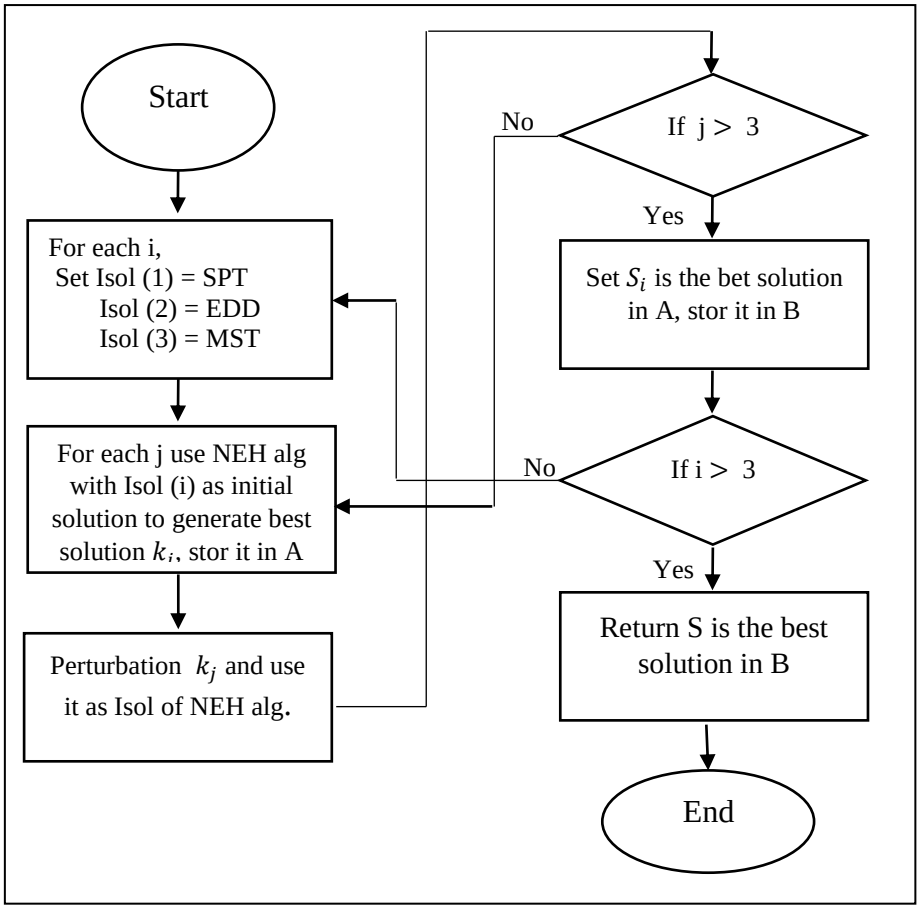

Fig 1 Proposed IMA algorithm.

remaining jobs of considered sequence which firstly proposed by [12] can be developed to use through can improve the performance of the original algorithms. Some relevant studies can be found in [24] and [25].

6. Conclusions and future research

In this work, we study a SMSP, the objective is to reach the optimal solution that minimize the sum of three objective functions. BAB without and with dominance rules were used to find optimal solution which compared with CEM method. We proposed a metaheuristic algorithm IMA based on the NEH alg, and then combined this algorithm with SA alg in two ways and the resulting is two MET algs (HIM-SA) and (HAS-IM). These two MET algs compared with the original algorithms: SA alg and IMA algorithm, the results show that this combination gives better performance than the original algorithm in reasonable execution times. The proposed algorithms can be developed to multi-objective scheduling to find non dominating solutions through multi-objective insertion algorithm based on the concept of NEH alg and multi-objective simulated annealing proposed in literature.

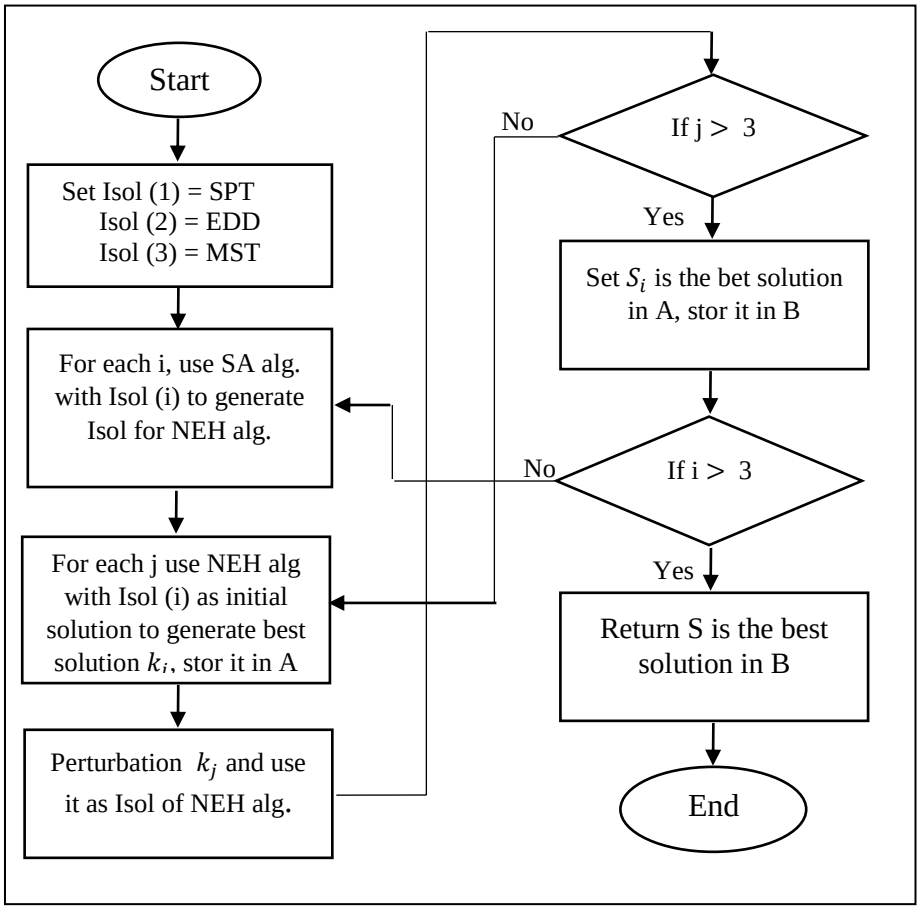

Fig 2 Proposed HSA-IM algorithm.

Table 1 The mean values of results of BAB and BABDR algorithms compared with CEM method.

n	CEM	BAB	NODS	T.BAB	BABDR	NODS	T. BABDR
4	688.33	688.33	22.89	0.028	688.33	12.89	0.0187
5	918.89	918.89	103.78	0.009	918.89	31.78	0.0034
6	1276.00	1276.00	266.33	0.018	1276.00	40.11	0.0034
7	1723.33	1723.33	1507.78	0.087	1723.33	193.44	0.0141
8	1877.33	1877.33	9772.22	0.643	1877.33	537.00	0.0375
9	2274.67	2274.67	124330.44	8.190	2274.67	1098.22	0.0821
10	3234.67	3234.67	1107720.44	73.014	3234.67	7342.11	0.5568
11		3397.11	2753986.44	254.679	3397.11	16851.33	1.947
12		4256.56	10171043.78	1022.570	4253.44	45029.56	5.287
13		5057.33	15408451.78	1474.084	5055.56	1098000.44	131.597
14					5521.000	1926510.222	212.031
15					6774.000*	6908525.778	803.709

NODS: Total active nodes. **T.BAB**: Mean value of execution time of BAB Alg. **T. BABDR**: Mean value of execution time of BABDR Algorithm

Table 2 The mean values of BAB.DR, SA, IMA, HIM-SA, and HSA-IM algorithms of objective function for problem instances size $4 \leq n \leq 15$, and $n = 20,30,40,50,75,100$.

n	BABDR	SA	IMA	HIM-SA	HSA-IM
4	688.33	688.33	688.33	688.33	688.33
5	918.89	918.89	918.89	918.89	918.89
6	1276.00	1276.00	1276.00	1276.00	1276.00
7	1723.33	1723.33	1723.33	1723.33	1723.33
8	1877.33	1877.33	1881.33	1877.33	1877.33
9	2274.67	2274.67	2274.67	2274.67	2274.67
10	3234.67	3234.67	3234.89 *	3234.67	3234.67
11	3397.11	3397.11	3401.11 *	3397.11	3397.11
12	4253.44	4256.78 *	4261.00 *	4253.44	4253.44
13	5055.56	5055.56	5065.11 *	5055.56	5055.56
14	5521.000	5521.00	5546.56 *	5521.00	5521.00
15		6770.89 **	6817.78	6770.89 **	6770.89 **
20		12127.22 **	12144.11	12127.22 **	12127.22 **
30		25646.89	25757.56	25629.33 **	25629.33 **
40		40979.44	40993.89	40922.78	40906.11 **
50		68591.22	68470.33	68395.22	68336.78 **
75		150198.89	149052.00	148967.11	148830.89 **
100		256401.89	253150.11	253412.11	252957.00 **

The symbol (*) mean that the mean value is near optimal. (i.e., some problem instances cannot be solved within 1800 second of CPU time). (*) The result is not optimal for some instances. (**) The result is the best.

Table 3 The mean execution times in seconds (T.) for SA, IMA, HIM-SA, HAS-IM.

n	T.SA	T.IMA	T.HIM-SA	T.HSA-IM
4	0.0886	0.0157	0.2480	0.0851
5	0.0797	0.0330	0.2673	0.0851
6	0.0867	0.0538	0.2814	0.0870
7	0.0830	0.0764	0.3056	0.0869
8	0.0834	0.1093	0.3192	0.0781
9	0.0818	0.1456	0.3611	0.0814
10	0.0851	0.1786	0.3768	0.0868
11	0.0848	0.2100	0.4167	0.0869
12	0.0886	0.2553	0.4372	0.0798
13	0.0813	0.2969	0.4738	0.0869
14	0.0817	0.3401	0.4964	0.0884
15	0.0763	0.3487	0.4758	0.0677
20	0.2083	0.7133	1.1197	0.2097
30	0.2188	1.5620	1.8274	0.2152
40	0.2117	2.8291	2.6934	0.2051
50	0.2206	4.3112	4.0161	0.2378
75	0.3663	10.8913	8.7844	0.2363
100	0.2724	20.5304	16.0068	0.2724

References

[1] J. Du and J. Y.-T. Leung, "Minimizing total tardiness on one machine is NP-hard," *Math. Oper. Res.*, vol. 15, no. 3, pp. 483–495, 1990.

[2] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science (80-.).*, vol. 220, no. 4598, pp. 671–680, 1983.

[3] F. Glover, "Tabu search—part II," *ORSA J. Comput.*, vol. 2, no. 1, pp. 4–32, 1990.

[4] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated local search," in *Handbook of metaheuristics*, Springer, 2003, pp. 320–353.

[5] P. Hansen and N. Mladenović, "Variable neighborhood search: Principles and applications," *Eur. J. Oper. Res.*, vol. 130, no. 3, pp. 449–467, 2001.

[6] C. Blum, A. Roli, and M. Sampels, *Hybrid metaheuristics: an emerging approach to optimization*, vol. 114. Springer, 2008.

[7] M. Lozano and C. García-Martínez, "Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report," *Comput. Oper. Res.*, vol. 37, no. 3, pp. 481–497, 2010.

[8] M. G. C. Resende, R. Martí, M. Gallego, and A. Duarte, "GRASP and path relinking for the max–min diversity problem," *Comput. Oper. Res.*, vol. 37, no. 3, pp. 498–508, 2010.

[9] R. J. Labios, S. Kim, H. Song, and B. Lee, "Hybrid multistarting GA-Tabu search method for the placement of BtB converters for Korean metropolitan ring grid," *Math. Probl. Eng.*, vol. 2016, 2016.

[10] M. K. Naik, L. Samantaray, and R. Panda, "A hybrid CS–GSA algorithm for optimization," in *Hybrid Soft Computing Approaches*, Springer, 2016, pp. 3–35.

[11] J. Pirgazi, M. Alimoradi, T. E. Abharian, and M. H. Olyaee, "An Efficient hybrid filter-wrapper metaheuristic-based gene selection method for high dimensional datasets," *Sci. Rep.*, vol. 9, no. 1, pp. 1–15, 2019.

[12] M. Nawaz, E. E. Enscore Jr, and I. Ham, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *Omega*, vol. 11, no. 1, pp. 91–95, 1983.

[13] D. M. Utama, D. S. Widodo, W. Wicaksono, and L. R. Ardiansyah, "A new hybrid metaheuristics algorithm for minimizing energy consumption in the flow shop scheduling problem," *Int. J. Technol.*, vol. 10, no. 2, pp. 320–331, 2019.

[14] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, 2017.

[15] S. Zheng, Z. Yang, Z. He, N. Wang, C. Chu, and H. Yu, "Hybrid simulated annealing and reduced variable neighbourhood search for an aircraft scheduling and parking problem," *Int. J. Prod. Res.*, vol. 58, no. 9, pp. 2626–2646, 2020.

[16] P. Stodola, K. Michenka, J. Nohel, and M. Rybanský, "Hybrid algorithm based on ant colony optimization and simulated annealing applied to the dynamic traveling salesman problem," *Entropy*, vol. 22, no. 8, p. 884, 2020.

[17] W. Yong *et al.*, "A new hybrid simulated annealing-based genetic programming technique to predict the ultimate bearing capacity of piles," *Eng. Comput.*, vol. 37, no. 3, pp. 2111–2127, 2021.

[18] K. K. Ghosh, R. Guha, S. Ghosh, S. K. Bera, and R. Sarkar, "Atom Search Optimization with Simulated Annealing--a Hybrid Metaheuristic Approach for Feature Selection," *arXiv Prepr. arXiv2005.08642*, 2020.

[19] M. DEMRI, S. FEROUHAT, S. ZAKARIA, and M. E. BARMATI, "A hybrid approach for optimal clustering in wireless sensor networks using cuckoo search and simulated annealing algorithms," in *2020 2nd International Conference on Mathematics and Information Technology (ICMIT)*, 2020, pp. 202–207.

[20] J. Clausen, "Branch and bound algorithms-principles and examples," *Dep. Comput. Sci. Univ. Copenhagen*, pp. 1–30, 1999.

[21] W. E. Smith, "Various optimizers for single-stage production," *Nav. Res. Logist. Q.*, vol. 3, no. 1–2, pp. 59–66, 1956.

[22] J. Lampinen, "Multiobjective nonlinear pareto-optimization," *Pre-investigation report, Lappeenranta Univ. Technol. Lab. Inf. Process.*, 2000.

[23] K. R. Baker and D. Trietsch, *Principles of sequencing and scheduling*. John Wiley & Sons, 2013.

[24] T. Hussein Elmenfy, Design of Velocity PID-Fuzzy Power System Stabilizer Using Particle Swarm Optimization, *WSEAS Transactions on Systems*, pp.9-14, Volume 20, 2021

[25] Ghada AL-Rawashdeh, Rabiei Bin Mamat, Jawad Hammad Rawashdeh, Evaluation of the Performance for Popular Three Classifiers on Spam Email without using FS methods, *WSEAS Transactions on Systems and Control*, pp.121-132 Volume 16, 2021