Online at: https://researchtrendsjournal.com ISSN No: 2584-282X

INTERNATIONAL JOURNAL OF TRENDS IN EMERGING RESEARCH AND DEVELOPMENT

Volume 1; Issue 1; 2023; Page No. 167-174

Received: 07-10-2023
Accepted: 18-11-2023

Desert sparrow optimization algorithm

[1]NSV Kiran Kumar and [2]Dr. Manish KR Singh

[1]Research Scholar, Sunrise University, Alwar, Rajasthan, India
[2]Associate Professor, Sunrise University, Alwar, Rajasthan, India

DOI: https://doi.org/10.5281/zenodo.12663258

Corresponding Author: NSV Kiran Kumar

Abstract

The Desert Sparrow Optimization (DSO) algorithm is a nature-inspired metaheuristic optimization technique that mimics the foraging behavior of sparrows in desert environments. It leverages the strategies employed by sparrows to survive and thrive in harsh, resource-constrained habitats. The algorithm is characterized by its simplicity, efficiency, and effectiveness in solving complex optimization problems across various domains. This research aims to better understand desert sparrows by Analysing their cooperative work allocation behaviour and developing an algorithm to Minimise Makespan.

Keywords: Allocation, developing, algorithm, makespan, DSO

Introduction

Among the several subfields of scheduling that make up industrial optimisation, production scheduling is crucial. Extensive research on production scheduling has been conducted by practitioners in order to fulfil the demands of large-scale market rivalry and increase production size in industrial and manufacturing firms. In order to carry out the production plan, the production scheduling takes into account the organisation of all production resources. In order to accomplish one or more performance metrics, tasks are distributed throughout the available machine system during this procedure. Many large-scale manufacturing businesses regularly use the flow shop scheduling method of production scheduling. The field of industrial mechanism is rife with combinatorial optimisation issues, one of which is permutation ow shop scheduling problems (PFSSP).
Research into the permutation ow shop has implications for both small- and large-scale manufacturing environments, as it provides a basic model of many real ow production lines. In addition, PFSSP gained popularity for its emphasis on mathematical modelling in order to find the best solutions for optimisation issues including both single and multiple objectives. Nevertheless, problems encountered by large-scale industrial businesses cannot be adequately addressed by this strategy. The per-mutation ow shop scheduling issues for systems with more than three machines are NP-hard combinatorial optimisation problems, as described by Garey, Johnson, and Sethi [GJS76]. These problems cannot be satisfactorily addressed using typical optimisation solution approaches. As a result, finding the best computational methods to address these issues has long been a hot subject in academia and business.

Literature and Review

Yinggao Yue *et al.* (2023) [1] There has been a lot of study and focus on swarm intelligence algorithms as of late. Optimisation issues are often addressed using swarm intelligence algorithms, a kind of biological heuristic. Combinatorial optimisation, job scheduling, process control, engineering prediction, and image processing are just a few areas that have benefited from the innovative ideas and novel approaches provided by classical swarm intelligence algorithms. Specifically, the sparrow search algorithm mimics the group foraging and anti-predation behaviours of sparrows to conduct local and global searches, and it is a novel kind of group intelligence optimisation algorithm. Researchers both at home and abroad have worked to improve upon the original sparrow search algorithm, which had a number of drawbacks that led to its practical applicability in other domains. These include a poor convergence accuracy, a sluggish convergence speed, and an easy fall into local optimum. This paper begins by introducing the sparrow search algorithm and its basic principle. It then analyses the factors affecting the algorithm's performance and proposes an improvement strategy. Finally, it compares and analyses the sparrow search algorithm's performance with other algorithms, such as particle swarm optimisation, monarch butterfly, colony spider, and pigeon swarm optimisation. Next, we will go into the creation and use of the sparrow search algorithm in various domains such as image processing, route tracking, defect detection, wireless sensor network routing performance optimisation, and power grid load forecasting. By combining the sparrow search algorithm's performance features with its application orientation, we can finally look forward to its future study and development.

Vishal Sharma *et al*. (2018) [2] Cooperative task allocation and decision are important aspects of networks that involve heterogeneous nodes operating in ad hoc mode like flying ad hoc networks (FANETs). The task allocation can be either mission based or simple utilization of available resources. In networks, including mission critical resources, cooperative task allocation and rendezvous are the key factors that drive the mission as well as optimize the performance. Many optimization algorithms have been designed and developed which focus on the cooperative behavior of nodes and also handle resources efficiently. Cooperative allocation and rendezvous both can be achieved by taking an example from biological world. In this paper, a new hill Myna and desert Sparrow optimization algorithm, namely HMADSO, is proposed for cooperative rendezvous and efficient task allocation. The application and analysis of proposed algorithm are shown for FANETs. To validate the proposed HMADSO, onboard processors, as well as simulation-based analysis, are carried out.

Meenakshi Sharma *et al*. (2022) [3] In this study, we tackle the bicriteria flow shop scheduling issue with sequence-independent setup time. Relative to the minimal value of makespan, the goal of the scheduling issue is to minimise the time that the system is used. The formulation of a mixed-integer programming paradigm allows for the separate handling of sequence-independent setup time and task processing time. In this work, we present a modified heuristic based on the nature-inspired Desert Sparrow Optimisation (DSO) algorithm that incorporates a backward to forward shift mechanism, a novel initial feasible solution technique, and a tie-breaking strategy. This heuristic is designed to solve the NP-complete flow shop scheduling problems with sequence-independent setup time. In addition, the suggested scheduling goals are optimised by formulating the delay time for the available machine system. To test how well the suggested heuristic works with up to twenty computers and five hundred tasks, a computational experiment is run. By comparing the suggested method to alternative constructive heuristics for the flow shop's mentioned scheduling issue, the study using the specified response variable average relative percentage deviation (ARPD) confirms that the strategy is an effective approach.

Kewal Krishan Nailwala, Deepak Gupta and Kawal Jeet (2016) [4] Jobs are continuously flowed through several equipment in a no-wait flow shop. Once begun, the work should be processed continuously via the machines without any waiting. This happens when workloads are processed sequentially on two different computers but no intermediate storage is available. Because NP-hardness is a difficulty in minimising makespan in flow shop scheduling, heuristic algorithms are essential for finding an ideal solution or a simple way to get closer to the optimal solution. In this work, we provide a heuristic approach for minimising makespan by modifying an existing heuristic, and a second heuristic algorithm for sequencing n-jobs through m-machines in a flow shop under a no-wait requirement. We compare the suggested heuristic algorithms to the NEH under no-wait and the MNEH heuristic for no-wait flow shop problem on 120 of Taillard's benchmark problems that have been published in the literature. By increasing the performance of NEH by 27.85%, MNEH by 22.56%, and the suggested constructive heuristic algorithm by 24.68%, the improvement heuristic surpasses all other heuristics on the Taillard's examples. The publication also includes numerical examples to clarify the algorithm's computing process. To arrive at these results, statistical tests of significance are conducted.

Janaki Elumalai *et al*. (2023) [5] At the operational decision-making level, job scheduling is a crucial responsibility of production logistics that helps organisations stay competitive. Using flow shop scheduling without task block criteria, processing times for multi machines are correlated with their probabilities. The end objective is to reduce the overall duration of all tasks. Finding the best or almost best sequence is Johnson's technique for reducing total elapsed time. It's easy to understand. The method is better understood with the assistance of a numerical demonstration.

Desert Sparrow-Cooperative Task Allocation Be Havior

Members of the bird family known as "sparrows" are able to work together on tasks such as constructing nests, looking for mates, and Recognising one another. There are two main types of these birds based on their unique traits: aggressive and protective. The sparrows who act as protectors shield the other sparrows from harm and tend to the young, while the sparrows that act as aggressors seek vengeance. Cooperative nest building is a unique skill of these birds. Task allocation during nest construction is dependent on cooperative reference to navigation, reconnaissance, and surveillance. In Table 1, we can see the desert sparrow traits that contribute to the allocation of cooperative tasks during nest construction.

Table 1: Cooperative task allocation functions and desert sparrow features

Practical requirements	Supporting features
Interactive cooperation	cooperative building of nests
Localization	vision
Protect and attack	coordination and identification
Task allocation	vision and common decision

As a group, the desert sparrows carry out the many responsibilities assigned to them with little confusion or conflict. In addition, while working together, these birds are most notable for their localization abilities; that is, they are fully aware of their current positions and the responsibilities that are allocated to them. In light of these inherent biological characteristics, these birds have developed a visual range set that allows them to detect hostile creatures and decide how to best defend themselves. Taking these traits into account, we create a cooperative position allocation optimisation method that aims to distribute resources rather than use them, based on the fact that each desert sparrow is controlled over its movement and route tracking.

Desert sparrow optimization algorithm for min Imization of makespan

Because of its close relationship to throughput and resource utilisation maximization, makespan is often considered a crucial scheduling criterion. In order to solve the multi-machine permutation ow shop scheduling model with the requirements of lowest makespan, the desert sparrow optimisation (DSO) method is devised and implemented in this part. Rather of treating setup time as a distinct scheduling parameter, this approach treats it as a component of processing time. Another way to depict the issue we've been talking about here is as: $F_m|perm, p_{ij}|C_{max}$.

Problem formulation

In a permutation-ow shop scheduling setting, think about a system with n tasks and m machines. Let J = {1, 2,.., n} and I = {1, 2, 3,..., m} serve as the job index sets and machine index sets, correspondingly. Moreover, every machine i in the set I carries out a distinct action. Additionally, O_{ij} on job j ∈ J. The fixed course of operations is followed by all the tasks as: $O_{1j} \rightarrow O_{2j} \rightarrow O_{3j} \rightarrow \ldots\ldots \rightarrow O_{mj}$ about the accessible machine system for every j in J. No machine will allow a work to be stopped while it is processing. Additionally, until the present operation of a task is finished, it cannot exit that computer. Two separate processes cannot run in parallel on a same task. In addition, no two tasks may do the same action in parallel. On top of that, the tasks go through the system in a predetermined order, but not at the same pace, so some machines may sit idle until the job after it is ready to be processed.

Allow for processing time (p_{ij}) as well as the time needed to finish (C_{ij}) of every task j in the set J executed on the machine i in the set I are positive parameters that are input and output to the scheduling model, respectively. Let x_{jk} and y_{ijk} are two-state variables in the ow shop scheduling model that use the minimal makespan criterion. This is, x_{ij} is set to 1 in the event that task j is scheduled at k^{th} if the task schedule is finalized, and 0 otherwise. This also applies to the binary variable y_{ijk} is set to 1 if task k is processed on machine i after job j in the available system of machines, and has no value otherwise. Subsequently, the MILP model is used for $F_m|perm, p_{ij}|C_{max}$ The issue is stated in the following manner:

Purposeful method

$$\min C_{max} = \min\ (\max_j\ C_{mj})$$

Depending on

$$C_{max} \geqslant C_{ij} \quad \forall (i,j) \in I \times J$$

$$C_{ij} \geqslant C_{(i-1)j} + \sum_{k=1}^{n} (p_{ij} \times x_{jk}) \quad \forall (i,j) \in I \times J$$

$$C_{ij} \geqslant C_{i(j-1)} + \sum_{k=1}^{n} p_{ij} \times x_{jk} \quad \forall (i,j) \in I \times J$$

$$\sum_{k=1}^{n} x_{jk} = 1 \ \forall j \in J$$

$$\sum_{j=1}^{n} x_{jk} = 1 \ \forall k = 1, 2, 3, \ldots, n$$

$$y_{ijk} + y_{ijk} = 1 \ \forall i \in I; j \neq k; j,k \in J$$

$$x_{jk}, y_{jki} \in \{0,1\} \forall i \in I; j,k \in J; j \neq k$$

$$p_{ij} \geqslant 0 \ \forall (i,j) \in I \times J$$

$$C_{0j} = 0 \quad \forall j \in J$$

$$C_{i0} = 0 \quad \forall i \in I.$$

Minimising the value of makespan, as shown in equation, is the goal of the mathematical model. Job j's makespan and completion time on machine i are seen in the relation. No task may leave the present machine until its current operation is finished, as shown in restriction. The current job will not be processed on machine i until the current work is finished, according to restriction. It is guaranteed that the work schedule is permutational by equations. There can be no two jobs scheduled at the same position in the final schedule, as shown by equation and by equation, which ensure that each job j has a unique position k. According to Equation, the sequence in which tasks are processed is fixed across all computers in the system that are accessible. The constraint presents the scope of decision variables of the scheduling model. Input scheduling parameters are non-negatively restricted by the constraint. The scheduling model incorporates dummy variables, which are represented by equations and.

Proposed computational technique

In order to find the optimal processing sequence for the available machine system, an algorithm for cooperative position allocation is developed based on the traits of the desert sparrow. This approach has the potential to provide a minimal approximation value of makespan. In two steps, the suggested heuristic is implemented. Initially, a modified DSO method is used to find a suitable task schedule. Then, to build the final solutions, the NEH heuristic's insertion technique is used. Also, a new rule for breaking ties among competing work schedules is created in the second phase. The DSO algorithm was created by Sharma, Reina, and Kumar [SRK17] to address the issue of resource allocation in mobile networks. Nevertheless, this optimisation method is well-suited to solving the permutation ow shop scheduling issue using the shortest makespan criterion, thanks to the desert sparrow's controllable movement and route tracing characteristics.

The following is a description of the components that make up the proposed algorithm: a backward to forward shift mechanism, a tie-breaking mechanism, and an initial viable task schedule approach based on vision indices.

First workable timetable based on vision index

The following procedures are conducted in order to ascertain the first workable timetable using the vision index as a biological property of the desert sparrow's cooperative behaviour:

1. Assign the machines the role of guide lines that will be followed by a predetermined series of tasks that are related to a group of active desert sparrows.
2. The amount of guider lines that the desert sparrows will sketch out is set to m.
3. Determine the visual range (Drange) that each desert sparrow should be able to see, which may also be used to specify the range of nodes that each sparrow should follow for transmission. The entire processing time, or range of transmission, is calculated by treating desert sparrows as tasks to be processed on available guider lines.

$$Drange_j = \sum_{i=1}^{m} p_{ij} \quad \forall j = 1, 2, 3, \ldots.., n$$

4. To execute work localization, we may think of machine C as an allocation centre that sits at the origin and calculates the placement of each job from there. This is how you may choose the hub of the machines that are currently available:

$$C = \begin{cases} m \;/\; 2, & m \text{ is even} \\ (m+1) \;/\; 2, & m \text{ is odd} \end{cases}$$

Let $desertsparrow_{dist}(j)$ signifies the separation of the j-th task from the hub.

$$desertsparrow_{dist}(j) = \sum_{i=1}^{(C)} p_{ij} \quad \forall j = 1, 2, 3, \ldots..,$$

Finding the middle ground distance $(DSintermediate_{j,k})$ between the j^{th} and k^{th} position with relation to the whole surface area that has to be traversed in order to build the incidence matrix or connection matrix using Drange.

$$DSintermediate_{j,k} = abs(Drange_j - Drange_k), \quad \forall j, k = 1, 2, 3, \ldots, n; j \neq k$$

Put together the incidence matrix with only one or zero entries. Rare desert sparrow accompanied by $0 < DSintermediate_{j,k} \leq Drange_j$ the value is denoted as 1 or 0 if it is not.

Find out how far the guider is $Dguider_j$ for j^{th} job. Rejecting tasks with the shortest processing times on the available machine system yields this maximum track length for each job.

$$Dguider_j = Drange_j - min(Dist(j, GL))$$

A guider line, or GL for short, is a machine representation.

Put a value on it W_j to j^{th} task by summing up all the incidence matrix elements that match to j^{th} row.

Determine each task's vision index by associating them with the dependence function provided by:

$$V_j = W_j / Dguider_j$$

To get a first workable schedule, sort all the tasks by vision index and then arrange them in decreasing order π.

Numerical illustration

A scheduling environment with 12 jobs and 4 machines is used to demonstrate the technique of implementing the suggested algorithm. The processing time values that are predictable and fixed p_{ij} Table 5 displays this information for every task j on machine i.2.

Beginning stage

Here, the machines serve as guide lines and the jobs as desert sparrows, and the issue is confined to that one area.

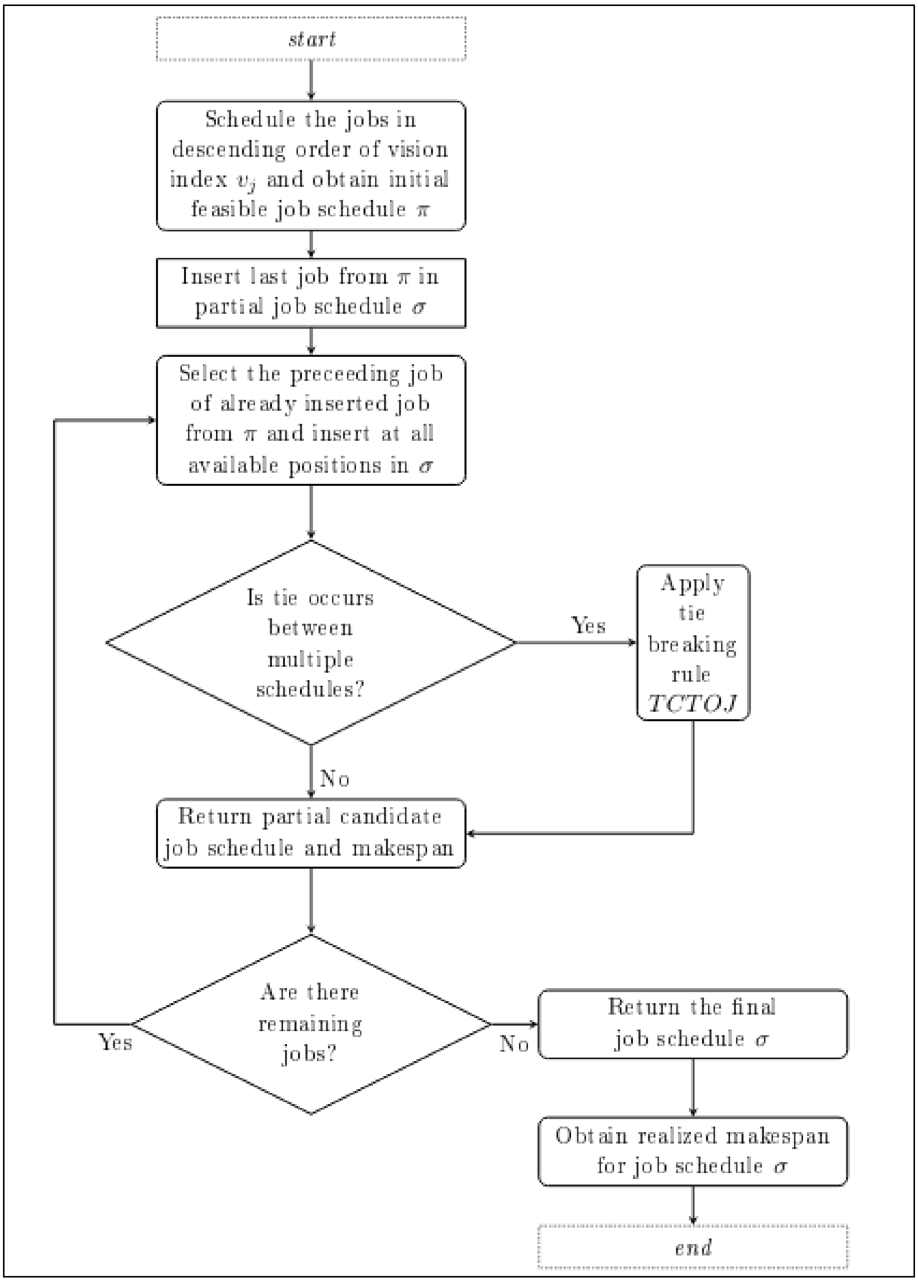

Fig 1: Flow chart of proposed heuristic

Table 2: Illustration of initial phase

Machines→ Jobs↓	M1	M2	M3	M4	Drange	DS_{dist} (i)
J_1	5	56	35	4	100	61
J_2	69	22	43	14	148	91
J_3	55	86	47	10	198	141
J_4	14	53	1	67	135	67
J_5	34	78	94	60	266	112
J_6	65	59	12	8	144	124
J_7	14	90	4	48	135	104
J_8	27	89	18	4	138	116
J_9	76	12	59	13	160	88
J10	70	9	40	90	209	79
J11	28	60	1	52	141	88
J12	57	23	73	1	154	80

The computation of the intermediate distance between the sparrows using their visual range yields the incidence matrix, as shown in Table 3.

Table 3: Incidence matrix

Jobs(i)	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J10	J11	J12	W(i)	Dguider	V(i)
J_1	0	0	0	1	0	0	1	0	0	0	0	0	2	96	0.0308
J_2	1	0	1	1	1	1	1	1	1	1	1	1	11	134	0.0764
J_3	1	1	0	1	1	1	1	1	1	1	1	1	11	188	0.0567
J_4	1	1	1	0	1	1	0	1	1	1	1	1	10	134	0.0758
J_5	1	1	1	1	0	1	1	1	1	1	1	1	11	232	0.0415
J_6	1	1	1	1	1	0	1	1	1	1	1	1	11	136	0.0786
J_7	1	1	1	0	1	1	0	1	1	1	1	1	10	131	0.0763
J_8	1	1	1	1	1	1	1	0	1	1	1	1	11	134	0.0821
J_9	1	1	1	1	1	1	1	1	0	1	1	1	11	148	0.0696
J10	1	1	1	1	1	1	1	1	1	0	1	1	11	119	0.0531
J11	1	1	1	1	1	1	1	1	1	1	0	1	11	140	0.0797
J12	1	1	1	1	1	1	1	1	1	1	1	0	11	153	0.0733

By sorting the tasks in decreasing order of visual range, the following first schedule may be achieved: π= (J_8, J_{11}, J_6, J_2, J_7, J_4, J_{12}, J_9, J_3, J_{10}, J_5, J_1).

Stage of advancement

The following procedures make up the improvement phase of the suggested computational technique's second stage:

Step 1: The first step is to schedule the jobs in reverse chronological order J_5 and J_1 as: (J_5, J_1) and (J_1, J_5). The obtained optimal sequence is (J_5, J_1) using the bare minimum of makespan, C_{max} = 270 units.

Step 2: Think about the following job, job J_{10}, and place it into the partial candidate job schedule in reverse chronological order. The resulting work schedules are: (J_5, J_1, J_{10}), (J_5, J_{10}, J_1), (J_{10}, J_5, J_1). The best time to work is (J_{10}, J_5, J_1) throughout the course of 350 units. When work J9 has been added to every open spot in the partial job schedule, the process is repeated in the same manner (J_{10}, J_5, J_1, J_3) A connection is made between the work schedules (J_{10}, J_5, J_9, J_1, J_3) and (J_{10}, J_9, J_5, J_1, J_3), after which the suggested rule for breaking ties is used to settle the matter. At last, the optimal work schedule is determined by: (J_{10}, J_5, J_9, J_3, J_1) with makespan, C_{max} = 396 pieces.

Following the stages of the suggested DSO heuristic yielded the incomplete task schedules presented in Table 4. Iterations 4 and 7 of the proposed heuristic algorithms reportedly use the tie-breaking approach.

Table 4: The best partial optimal job schedules constructed by the DSO heuristic after inserting each job of π

Iteration	Inserted job	σ	Cmax	Tie
1	J_5	(J_5, J_1)	270	No
2	J10	(J_{10}, J_5, J_1)	340	No
3	J_3	(J_{10}, J_5, J_3, J_1)	350	No
4	J_9	(J_{10}, J_5, J_9, J_3, J_1)	396	Yes
5	J12	(J_{12}, J_{10}, J_5, J_9, J_3, J_1)	453	No
6	J_4	(J_4, J_{12}, J_{10}, J_5, J_9, J_3, J_1)	467	No
7	J_7	(J_4, J_{12}, J_7, J_{10}, J_5, J_9, J_3, J_1)	485	Yes
8	J_2	(J_4, J_{12}, J_7, J_{10}, J_5, J_9, J_3, J_2, J_1)	532	No
9	J_6	(J_4, J_{12}, J_7, J_{10}, J_5, J_9, J_3, J_6, J_2, J_1)	550	No
10	J11	(J_4, J_{12}, J_7, J_{10}, J_5, J_{11}, J_9, J_3, J_6, J_2, J_1)	578	No
11	J_8	(J_4, J_8, J_{12}, J_7, J_{10}, J_5, J_{11}, J_9, J_3, J_6, J_2, J_1)	658	No

Table 5: Makespan value obtained for heuristics

Instances	Best soln	NEH	NEH-D	NEHKK1	NEHKK2	CLwts	DSO	Instances	Best soln	NEH	NEH-D	NEHKK1	NEHKK2	CLwts	DSO
20x5	1278	1286	1297	1296	1297	1305	1313	50X10	2991	3135	3177	3169	3130	3155	3130
	1359	1365	1383	1365	1366	1371	1367		2867	3032	3008	3058	3062	3076	2986
	1081	1159	1132	1132	1159	1135	1132		2839	2986	3036	3046	3025	3013	2929
	1293	1325	1306	1312	1314	1323	1314		30063	3198	3176	3184	3187	3156	3135
	1235	1305	1283	1305	1278	1305	1305		2976	3160	3136	3121	3146	3185	3139
	1195	1228	1264	1231	1228	1210	1225		3006	3178	3156	3158	3169	3168	3148
	1234	1278	1251	1278	1273	1270	1278		3093	3277	3271	3287	3289	3259	3289
	1206	1223	1221	1223	1222	1224	1227		3037	3123	3162	3157	3184	3147	3168
	1230	1291	1289	1263	1273	1292	1262		2897	3002	3001	3040	3032	3047	3025
	1108	1151	1131	1151	1151	1127	1120		3065	3257	3179	3239	3215	3204	3122
20X10	1582	1680	1692	1680	1622	1646	1654	50X20	3850	4082	4064	4051	4051	4069	4059
	1659	1729	1718	1723	1732	1711	1735		3704	3921	3937	3993	3975	3958	3901
	1496	1557	1538	1529	1563	1559	1535		3640	3927	3820	3892	3855	3882	3882
	1377	1439	1427	1428	1429	1455	1437		3723	3969	3948	3948	3927	3998	3931
	1419	1502	1500	1502	1504	1502	1502		3611	3835	3827	3859	3860	3834	3863
	1397	1453	1447	1434	1445	1433	1451		3681	3914	3844	3855	3979	3859	3848
	1484	1562	1529	1562	1539	1526	1497		3704	3952	3975	3987	3934	3931	3931
	1538	1609	1593	1648	1601	1610	1620		3691	3938	3982	3939	3926	3925	3908
	1593	1647	1663	1647	1648	1647	1647		3743	4052	3941	3877	3941	3949	4015
	1591	1653	1656	1656	1684	1649	1649		3756	4079	3972	4017	3961	4012	3960
20X20	2297	2410	2380	2404	2394	2397	2410	100X5	5493	5519	5514	5514	5504	5514	5519
	2099	2150	2162	2137	2181	2150	2162		5268	5348	5297	5284	5291	5289	5284
	2326	2411	2387	2414	2386	2411	2416		5175	5219	5215	5227	5195	5216	5207
	2223	2262	2248	2264	2262	2290	2262		5014	5023	5027	5023	5029	5023	5030
	2291	2397	2363	2375	2353	2394	2390		5250	5266	5255	5255	5255	5256	5281
	2226	2349	2378	2349	2283	2349	2353		5135	5139	5139	5139	5139	5139	5146
	2273	2362	2366	2383	2386	2360	2382		5246	5259	5283	5256	5246	5284	5307
	2200	2249	2279	2249	2283	2249	2249		5094	5120	5110	5130	5101	5123	5151
	2237	2320	2292	2306	2360	2323	2396		5448	5489	5470	5489	5454	5482	5507
	2178	2277	2308	2220	2260	2270	2283		5322	5341	5346	5345	5346	5344	5346

Table 6: Cont. Makespan value obtained for heuristics

Instances	Best soln	NEH	NEH-D	NEHKK1	NEHKK2	CLwts	DSO	Instances	Best soln	NEH	NEH-D	NEHKK1	NEHKK2	CLwts	DSO
50X5	2724	2733	2724	2724	2729	2729	2729	100X10	5770	5846	5845	5846	5899	5831	5868
	2834	2843	2882	2882	2848	2848	2882		5349	5453	5426	5427	5522	5431	5405
	2621	2640	2654	2626	2628	2633	2647		5676	5824	5740	5755	5780	5777	5755
	2751	2782	2768	2780	2773	2762	2782		5781	5929	5935	5926	5997	6006	5969
	2863	2868	2890	2876	2864	2886	2864		5467	5679	5579	5649	5643	5633	5540
	2829	2850	2846	2841	2836	2839	2835		5303	5375	5389	5358	5414	5363	5403
	2725	2758	2736	2766	2747	2732	2743		5595	5704	5736	5699	5688	5699	5730
	2683	2791	2804	2805	2784	2788	2735		5617	5760	5784	5733	5719	5727	5778
	2552	2576	2571	2561	2590	2565	2568		5871	6032	5993	5978	5986	5990	5960
	2782	2790	2790	2787	2782	2804	2794		5845	5918	5903	5903	5903	5903	5909
100X20	6241	6541	6540	6541	6574	6588	6550	200X20	11195	11594	11593	11596	11603	11530	11601
	6183	6523	6506	6497	6544	6482	6444		11203	11677	11708	11736	11713	11698	11676
	6271	6639	6535	6572	6547	6584	6575		11281	11852	11811	11797	11804	11788	11772
	6269	6557	6635	6594	6606	6597	6586		11275	11803	11627	11660	11852	11695	11601
	6314	6695	6600	6637	6601	6642	6637		11259	11685	11668	11668	11672	11668	11623
	6364	6664	6666	6742	6728	6694	6690		11176	11629	11601	11612	11688	11676	11645
	6268	6632	6682	6611	6634	6617	6608		11360	11833	11793	11850	11762	11809	11793
	6401	6739	6769	6818	6771	6818	6789		11334	11913	11791	11825	11817	11753	11714
	6275	6677	6599	6599	6660	6650	6583		11192	11673	11669	11732	11726	11678	11662
	6434	6877	6658	6706	6662	6699	6695		11288	11869	11753	11776	11806	11838	11713
200X10	10862	10992	10942	10952	10942	10992	10976	500X20	26059	26770	26564	26689	26629	26659	26562
	10480	10716	10642	10734	10665	10741	10633		26520	27232	27127	27175	27187	27145	27111
	10922	11025	11033	11050	11045	11027	11066		26371	26848	26878	26864	26968	26835	26831
	10889	11057	11057	11057	11096	11067	11057		26456	27055	26849	26994	27030	26935	26829
	10524	10645	10628	10695	10601	10669	10634		26334	26727	26768	26762	26767	26890	26719
	10329	10458	10458	10423	10477	10467	10454		26477	27992	26983	26972	27035	26990	26952
	10854	10989	10972	10991	10963	10962	10904		26389	26797	26668	26745	26708	26726	26701
	10730	10829	10798	10852	10847	10857	10850		26560	27138	27026	27129	27007	27165	27077
	10438	10574	10598	10527	10487	10558	10511		26005	26631	26438	26539	26591	26555	26433
	10675	10807	10789	10830	10814	10790	10815		26457	26984	26926	26938	27001	26877	26863

Conclusion

the desert sparrow optimization algorithm to solve the multimachine flow shop scheduling problems which was originally developed by Sharma, Reina and Kumar for highly dynamic cooperative network node formation and nonredundant task handling with these network nodes. the various characteristics of desert sparrow that act as the base to develop the desert sparrow optimization algorithm to solve the combinatorial optimization problems. the widely studied permutation ow shop scheduling problem of minimum makespan with complete procedure of implementation proposed optimization algorithm to solve this problem.

Reference

1. Yue Y, Cao L, Lu D, Hu Z, Minghai X, Wang S, Li B, Ding H. Review and empirical analysis of sparrow search algorithm. Artificial Intelligence Review. 2023;56:1-53. https://doi.org/10.1007/s10462-023-10435-1.
2. Sharma V, Gutiérrez D, Kumar R. HMADSO: A Novel Hill Myna and Desert Sparrow Optimization Algorithm for Cooperative Rendezvous and Task Allocation in FANETs. Soft Computing. 2018;22. https://doi.org/10.1007/s00500-017-2686-4.
3. Sharma M, Sharma M, Sharma S. Desert sparrow optimization algorithm for the bicriteria flow shop scheduling problem with sequence-independent setup time. Oper Res Int J. 2022. https://doi.org/10.1007/s12351-021-00675-w.
4. Nailwala KK, Gupta D, Jeet K. Heuristics for no-wait flow shop scheduling problem; c2016.
5. Elumalai J, Appasamy T, Ramalingam B, Joseph JPM, Radhakrishnan K. Optimizing the completion time for flow shop scheduling model for multi machine and multi job without job block criteria, 2023.
6. Taillard E. Benchmarks for basic scheduling problems. Eur. J Oper Res. 1993;64(2):278-285.
7. Vanchipura R, Sridharan R. Development and analysis of constructive heuristic algorithms for flow shop scheduling problems with sequence-dependent setup times. Int J Adv Manuf Technol. 2013;67(1):1337-1353.
8. Viswanath A, Sridharan R, Ram Kumar PN. Hybrid genetic algorithm for multi-objective flow shop scheduling problem with sequence dependent setup time: parameter design using Taguchis robust design method. Int J Process Manag Benchmark. 2019;9(4):419.
9. Raj S, Panchal VK, Chopra R. Cooperative Optimization Algorithm Based on Desert Sparrow for FANETs. March 31, 2020. Available at SSRN: https://ssrn.com/abstract=3565250 or http://dx.doi.org/10.2139/ssrn.3565250.
10. Sharma M, Sharma M, Sharma S. Desert sparrow optimisation algorithm for permutation flowshop scheduling problems. Int J Math Oper Res. 2020;17(2):253-277.
11. Sharma S, Gupta D, Nailwal KK. Bi-criteria multistage flow shop scheduling with sequence-dependent setup

times. Int J Oper Res. 2017;29(1):127-147.
12. Sharma V, Reina DG, Kumar R. HMADSO: a novel hill Myna and desert Sparrow optimization algorithm for cooperative rendezvous and task allocation in FANETs. Soft Comput. 2017;22(18):6191-6214.
13. Ribas I, Companys R, Martorellc XT. An iterated greedy algorithm for solving the total tardiness parallel blocking flow shop scheduling problem. Exp Syst Appl. 2019;121(1):347-361.
14. Pargar F, Zandieh M, Kauppila O, Kujala J. The effect of worker learning on scheduling jobs in hybrid flow shop: a bicriteria approach. J Syst Sci Syst Eng. 2018;27(3):265-291.
15. Meng T, Pan QK, Li JQ, Sang HY. An improved migrating birds optimization for an integrated lot-streaming flow shop scheduling problem. Swarm Evolut Comp. 2018;38(1):64-78.