**Periodicals of Engineering and Natural Sciences** ISSN 2303-4521
Vol. 8, No. 3, August 2020, pp.1555-1565

# Improved scatter search algorithm based on meerkat clan algorithm to solve NP-hard problems

**Saif Ali Abd Alradha Alsaidi[1], Dena Kadhim Muhsen[2], Sura Mazin Ali[3]**
[1] Computer science, College of Education for Pure Science, Wasit University
[2] Computer science, University of Technology
[3] Computer science, College of Political Science, Al Mustansirya University

## ABSTRACT

A modified Scatter Search (SS) algorithm based on Meerkat Clan Algorithm (MCA) has been presented in this paper. SS is one of the important metaheuristic algorithms, while the MCA is one of the recent swarm intelligence algorithms. The modified SS algorithm, including the main steps of MCA, through it the diversity and exploration of SS-MCA's solutions, have improved. The proposed algorithm has been applied to two important NP-Hard problems (Travelling Salesman Problem (TSP) and Flexible Job Shop Scheduling Problem (FJSSP)) to verify the performance of SS-MCA. The experimental results show that the performance of SS-MCA is better than both SS and MCA, respectively.

**Keywords**: Meerkat Clan Algorithm, Scatter Search Algorithm, NP-Hard Problem, Traveling Salesman Problem, Flexible Job Shop Scheduling Problem.

***Corresponding Author:***

Saif Ali Abd Alradha Alsaidi,
Computer science, College of Education for Pure Science,
Wasit University,
Address.
E-mail: salsaidi@uowasit.edu.iq

## 1. Introduction

Metaheuristic algorithms are nature and swarm-inspired; there are many types of this algorithm such as Particle swarm optimization algorithm, Ant colony optimization, Bat algorithm, Cuckoo search, Firefly algorithms[1]. Many metaheuristic algorithms are developed to use in various fields of optimization such as network design, planning, Traveling Salesman Problem (TSP), Flexible Job Shop Scheduling Problem (FJSSP), data mining, and others [2]. A metaheuristic is has substantiated to very efficient methods for solving different types of NP-hard problems. The NP refers to "non-deterministic polynomial time", which mean one move rather through a given configuration [3]. In NP-hard problems, the TSP and FJSSP are the well-known problems which means that there is not promising to get the optimal route and no exact algorithm to solve it in polynomial time [4]. Also, some other NP-Hard problems such as decision problems, resource-constrained project scheduling, and channel assignment where the efficient method used to check the solution which is already obtained in a good manner [5, 6]. Scatter search (SS) is a good evolutionary methodology based metaheuristic, the effective strategies of combination solution vectors have been used and reducing the randomization in different problem, to achieve the main goal which reaches to the better solutions [7].

In different types of optimization problems, the SS produces a good solution through using the diversification and intensification strategies. The SS begins with the initial set of solutions from which a subset of solutions are chosen for the reference set RefSet which develops by intensification and diversification mechanisms, update the reference set by the guided combination of the solution in this set to produce new solutions [8].

In this paper, the scatter search has been improved and using the meerkat clan algorithm, then solving TSP and FJSSP to evaluate the proposed algorithm. The results of tow problems are compared to test the improved algorithm. The paper structure is ordered as, some of the related work submitted in section 2. The standard Scatter Search Algorithm illustrated in section 3. Section 4 presents the basic Meerkat Clan Algorithm. The proposed algorithm shows in section 5. Section 6 explains the Experiments results of the improved algorithm on TSP and FJSSP. Section 7 includes Conclusions.

## 2. Related works

There are many studies in the field of metaheuristic algorithms improvement across many years and in particular when considering in scatter search. Some of the asymptotic research are submitted in this section.

In [9] (2011), the improved scatter search algorithm has been presented to predicting all-atoms protein structures using the CHARMM22 energy model. The improved algorithm; product 3D structure of the whole protein, by decreasing the energy function associated with protein folding. The experiments showed that the improved SS produces Three - dimensional structure with satisfactory and robust root mean square deviations from the protein reference.

In [10] (2012), the improved SS has been submitted using the Bees Algorithm, the SS improved by apply random exploration on the search of a problem and increasing to produce good solutions. The experimental results indicated that the proposed SS algorithm is better than the existing SS algorithm when it comes to the nearest optimal solutions.

In [11] (2012), the improvement of a new Scatter Search algorithm for the stochastic travel- time vehicle routing problem with simultaneous pick-ups and deliveries through merge a new chance-constrained programming method. To a comparison for a performance, the genetic algorithm methods are used. The experiment results show the solutions of SS is very good and best than the GA solutions.

In [12] (2013), presents an improved Scatter Search algorithm by adding some of Bees Algorithm concepts. The existing and improved Scatter Search algorithms were compared and tested on local 4–Colors Mapping test problems. The experiment results show that the improved SS algorithm is better in performance than the original SS. In [13] (2013), the scatter search algorithm has been improved by using Cuckoo Search (CS), the search is done randomly, which led to many better solutions. The results in a new improved algorithm are the best compare to the original one to achieved and find the optimal solutions.

In [14] (2019), Introduce an enhanced scatter search algorithm to overcome the difficulty of calculating the parameter. Improved SS relies on the hybridization of quasi-opposition-based learning in the developed scatter search (QOBLESS) strategy. A large-scale Chinese Hamster Ovarian (CHO) cell metabolic model is used to check the algorithm. In [15, 16](2020), The proposed improved Scatter Search (SS) algorithm is designed to address the issue of corridor allocation. Many of the development models and mechanisms are implemented to the SS dependent on the specific features of the problem, like the adoption of a dynamic reference set update method, simulated annealing operation, and an improved subset generation method. The findings reveal that the consistency of the solution is higher than the initials SS.

## 3. Information backgrounds and methodology

In this section will dicuss and explain the information background about the basic algorithms which is scatter search algorithm and meerkat clan algorithm that has been used to improved a new prposed algorithim SS-MCA to solve the NP-hard problems.

### 3.1 Basic scatter search algorithm

Scatter search (SS) is a good technique to find metaheuristic solutions, the most successful of it which optimization of single-objective for several of optimization problems, in addition to used SS in multi-objective optimization, generally, SS deal in tow filed with continuous, nonlinear problems[17, 18].

The basic strategy of SS is a systematic combination of the population solutions named reference set. This set updated to include each of better and disperse solutions, and the resulting has been improved by the method of local search, repeated these steps until met the terminate condition. The aim of this technique is to find better solutions [19, 20]. The scatter search is used mechanisms which not restricted to consolidated design permit the vast of strategic capabilities that may ensure effective performance in specific implementation[21].

The Scatter Search contains five major components summarized as[22] :

1. A Diversification Generation Method: build a large population (p) of different experiential solutions in the search space. The main aim is to Generate and build solutions different from each other to ensure diversity.
2. An Improvement Method: this part process enhancing the experiential solutions to one or better solutions. Improvement heuristic has been applied when the solution is already possible.
3. A Reference Set Update Method: this component related to generate and update the reference set which is initialed from various solution set P, the solutions have been two subsets, one for high-quality solutions named RefSet1 and the other for diverse solutions named RefSet2.
4. A Subset Generation Method: this method determines the mechanism for selected subsets of solutions from the reference set in order to be combined.
5. Solution Combination Method: this process transform solution in a given subset to one or more new combined solutions. This method applied the combination with considerable taking the good characteristic of solutions in every subset without based on randomization.

The combination results are handle by Improvement Method in order to produce an enhanced solution. Then, the Reference Set Update Method takes the protected solution to update the reference set following both intensification and diversification criteria [23-25].
These steps are continued while RefSet is modified. The search cheek if the condition for terminating determined by the user is met, then the algorithm submits the final better solutions. Otherwise, the new population have been built, and add to the initial population of the RefSet. The following steps show the basic SS algorithm [26, 27].

**Scatter Search Algorithm**
**Input:** Random population selection;
**Begin**
**step1**: Use the Diversification Generation Method to generate a population of solutions;
**step 2**: Improve solutions generated by Improvement Method;
**step 3**: Apply Reference Set Update Method (RefSet1 for Best solutions and RefSet2 for Diversity solutions);
**step 4**: While (the termination condition is not met) Do
Subset Generation Method;
Solution Combination Method (Combine each subset into a new solution);
Apply Improvement Method to improve and update each combined solution;
Reference Set Update Method to update the reference set depend on the merge of the of the current reference set and the new combined solutions;
**step 5**: End while
**End**
**Output**: best solutions.

### 3.2 Basic meerkat clan algorithm

Results Meerkat is a social creature living in states of 5 to 30 people. To be friendly animals, they Collaborate in Duties for latrine and parental care. Every crowd has an alpha male leader and an alpha female conquer. There is a domain for crawling crowd which they transfer here and there if feeding is uncommon or if it is restricted by a more grounded horde. The critical factors used in the MCA algorithm derived from the behavior of Meerkat. These factors are n refer to the size of a clan, m is the size of foraging group, c for the size of care group, the Fris for worst foraging and Cr for care ratio, and K for neighbors number[28].

The parameters for the algorithm have been generated by the first algorithm, and n identifies the solution set of the clan. the size of the foraging group is indicated by M and c for the care group. In addition, the algorithm determines the worst foraging and care rate. Lastly, generate the number of K neighbors [28].
In the beginning, randomly initialization for solutions group by the algorithm called the size n clan. The developed clan is estimated by computing the fitness function. Choose the best solution which is called sentry through the generated clan. Two groups where The remaining clan is divided into size m for the foraging group (which is m < n) and size c for a care group of (n-m-1)[28].
Recently, in the foraging group, every solution delivered to the neighbor produces a sub-algorithm shown in the MCA algorithm, return the optimal answer from the neighbor generated. Receive K, Sentry, and Foraging in neighbor generate. The neighbor generates produces K neighbors from foraging and calculates the fitness function for it. If all the neighbors produced are the worst from foraging, then the Sentry K neighbor will be produced. The best solution is picked from the K neighbors and returned to the major algorithm. If the foraging transmitted to the neighbor produces the worst of the best neighbor, subsequently delete it [28-30].
The separation of the worst solution in the foraging group is done by the algorithm based on the worst foraging ratio and changes it with the best solution in the care group. Remove the worst solution in the care group based on the ratio of worst care and replace it with a new random solution [28, 29].
The optimal solution value in the foraging group is chosen and contrasted with Sentry, whether it is possible to replace Sentry with the best solution in the foraging group. While the condition of termination takes place, the steps are mentioned shall be repeated. Finally, the best solution is the Sentry.[28, 29].

```
Meerkat Clan Algorithm
Input: Parameters of MCA; Output: best solutions.
Begin
Initialize random of solutions n         // n is size of mob between (30-50)
Calculate the solutions fitness
Sentry = best solution
Diviide the solutions into 2 groups (foraging group and care group)
While terminated condition not met Do
For i=1 to m                              // m refere to  size of foraging   where m < n
                                          // c refere to size of care          where n-m-1
Call neighbor_generat (k, Sentry, foriaging(i), best_one) // k neighbour solution   foriaging(i)= best one from k neighbor
end for
Swap in foraging group the worst Fr solution by best ones' solution in care group;
                                                        // Fr worst foraging rate
Drop the worst Cr solution from care group and produce ones' solution randomly;
                                                        //   Cr worst care rate
determine the best one solution of foraging name it best_forg
If best_forg <= Sentriy then Sentriy = best_forg
end while
End
```

.

## 4. Method preposed imperoved scatter search algorithm based on meerkat clan algorithm (SS-MCA)

Swarm intelligence based meerkat clan algorithm used to improve the SS algorithm (SS-MCA). Meerkat clan algorithm inspired by the behavior of meerkat clan social. Meerkat clan algorithm has more than good features as a meta-heuristic algorithm, such as good solution diversity, well neighbor's generation, its ability in solving most NP-hard problems, and finding the nearest global optimum solution in a reasonable time.
SS algorithm including several steps, there is more than one place to improve it. The SS algorithm. Whilst

working, one of the best places to improve the SS algorithm via the meerkat clan algorithm is an update of the Reference Set and Improvement Method.

Time factor forms the big challenge in the improvement of the SS algorithm. Therefore, any improving must be taken into consideration in case the performance will increase. If the Improvement method is implemented to all populations instead of to every other new solution produced by the Combination Method because that will take lots of time, it will actually impact on one of the metaheurstic algorithim SS algorithm, the main goal of which is a reasonable time to find the optimal solution.

Initially, the foraging group equal to RefSet1 and the care group equal to RefSet2. Of course, in the foraging group, Sentry represents the best solution. The substantial stages of the meerkat clan algorithm have been applied in the main loop of the SS algorithm. By enhancing foraging and care groups, the Reference Set has been improved. In this step, the neighbour's generation strategy plays a big role to enhance foraging and care groups. Through these steps, the sentry has been updated.

Then replacing the worst solutions in the foraging group by good ones in the care groups and the worst solutions in the care group will be replaced by random others. The neighbor's generation strategy depends on 3-opt. Operation in most cases, this operation applies to a current solution, and repeatedly within k neighbors on the same single solution. The bold steps represent the proposed changes to the original SS algorithm.

```
Basic Scatter Search-Meerkat Clan (SS-MCA)
Begin
Initialize the population Pop using a Diversification Generation Method.
Apply the Improvement Method to the population.
Reference Set Update Method (Good solutions for RefSet1 and Diversity solutions for RefSet2)
Foraging group = RefSet1
Care group = RefSet2
While (itr < MaxItr) do
    While (Reference set is changed) do
        Sentry = best solution in Foraging group
        For i=1 to each one in foraging group
            Call neighbors_generate (k, Sentry, foraging(i), best_one)
            foraging(i)= best one from k neighbor
        end for
        Sentry = best solution in Foraging group
        Swap in foraging group the worst Fr solution by best ones' solution in care group;
        Drop the worst Cr solution from care group and produce ones' solution randomly;
        RefSet1=Updated Foraging group;
        RefSet2=Updated Care group;
        Subset Generation Method
        While (subset-counter < > 0) do
            Solution Combination Method.
            Improvement Method.
            Reference Set Update Method;
        End while
    End while
End while
```

## 5. Result and discussion

Two NP-Hard problems have been selected to verify the performance of the modification algorithm, Travelling Salesman Problem (TSP), and Flexible Job Shop Scheduling Problem (FJSSP), because these two problems have several applications in the real worlds. Table 1 shows the important parameter's range values of SS-MCA which are used in TSP & FJSSP solving.

Table 1. SS-MCA parameter range values

| Parameter | Value |
|---|---|
| Population Size | 80 - 100 |
| RefSet1 = Foraging group | 30 |
| RefSet2 = Care group | 20 |
| K : No. of Generated Neighbors | 3 - 5 |
| Max. Iteration | 100 - 130 |
| Fr : Worst Foraging Ratio | 0.2 |
| Cr : Worst Care Ratio | 0.25 |

The proposed algorithm was coded in MATLAB R2015b and applied on Intel Core i7 2.70 GHz personal computer with 8GB RAM.

**5.1 Travelling salesman problem (TSP)**

Benchmark TSPLIB [31] (available in, http://www.iwr.uniheidelberg.de/iwr/ comopt/software/TSPLIB95/) represents the standard dataset of TSP have been selected to verify the performance of SS-MCA. The experimental results compare with both MCA [28]and the SS algorithm.

The experimental results of TSP have relied on 3 factors, nearest optimal solution (NOPT), estimated running time and Error Ration (ER), where

$$ER = \frac{NOPT - Optimal}{Optimal} \tag{1}$$

Table 2 shows the results of NOPT for SS, MCA, and SS-MCA for benchmark TSPLIB, Table 3 shows the estimated time, and Table 4 shows the ER of SS-MCA comparison with other modified SS algorithms and Table 5 shows the ER of SS-MCA comparison with different algorithms. All these experiments where the parameters as in Table 1.

Table 2. Results of NOPT for proposed SS-MCA with both original SS & MCA (For 10 Runs)

| Instances | Comparisons of SS-MCA with SS & MCA for optimality | | | |
|---|---|---|---|---|
| | Optimal in TSPLIB [31] | NOPT in SS | NOPT in MCA [28] | NOPT in SS-MCA |
| A280 | 2579 | 28589 | 3094 | 3043 |
| Att48 | 10628 | 87053 | 22743 | 18599 |
| Berlin52 | 7542 | 18431 | 9201 | 8899 |
| Bier127 | 118282 | 508461 | 185702 | 162046 |
| Dantzig42 | 699 | 1836 | 803 | 768 |
| Eil51 | 426 | 1054 | 477 | 464 |
| Eil76 | 538 | 1232 | 645 | 624 |
| Fri26 | 937 | 1414 | 1105 | 1030 |
| KroA100 | 21282 | 116553 | 37456 | 28517 |
| KroB200 | 29437 | 274029 | 80951 | 62700 |
| Lin105 | 14379 | 86938 | 25307 | 17829 |
| Pr76 | 108159 | 413947 | 170891 | 141688 |

Table 3. Estimated time in (Sec.) for proposed SS-MCA with both original SS & MCA

| Instances | Average of Estimated Time for SS, MCA & SS-MCA in (Sec.) | | |
|---|---|---|---|
| | SS | MCA | SS-MCA |
| A280 | 2.362 | 2.998 | 4.187 |
| Att48 | 2.132 | 2.832 | 3.302 |
| Berlin52 | 1.364 | 2.094 | 2.998 |
| Bier127 | 1.884 | 2.745 | 3.485 |
| Dantzig42 | 1.881 | 2.247 | 2.897 |
| Eil51 | 2.126 | 2.895 | 3.521 |
| Eil76 | 2.604 | 3.105 | 3.985 |
| Fri26 | 1.341 | 1.484 | 2.114 |
| KroA100 | 3.954 | 4.457 | 5.153 |
| KroB200 | 4.562 | 5.247 | 6.927 |
| Lin105 | 4.157 | 4.987 | 5.783 |
| Pr76 | 3.177 | 3.998 | 5.052 |

With regard to solving TSP, our proposed SS-MCA gives a good result compared with the original SS by 64.7% and gives the best results than the MCA by 12.17% (these percentage as an average). On the other side, the time of SS-MCA is larger than both original SS and MCA by 1.62 sec. & 1.28 sec as an average, respectively.

Table 4. Results of error ratio for proposed SS-MCA with some modified SS algorithms (For 10 Runs)

| Instances | Optimal in TSPLIB [24] | ER for NOPT of Different Modified SS Algorithms | | | | |
|---|---|---|---|---|---|---|
| | | SS | SS-MCA | SS-Bees [10] | HSS [32] | SS-CS [13] |
| A280 | 2579 | 10.08 | 0.17 | 8.95 | 4.82 | 6.65 |
| Att48 | 10628 | 7.19 | 0.75 | 6.4 | 5.86 | 6.51 |
| Berlin52 | 7542 | 1.44 | 0.17 | 1.25 | 0.42 | 0.59 |
| Bier127 | 118282 | 3.29 | 0.36 | 2.78 | 1.48 | 2.38 |
| Dantzig42 | 699 | 1.62 | 0.09 | 1.15 | 0.45 | 0.7 |
| Eil51 | 426 | 1.47 | 0.08 | 1.07 | 1.09 | 0.88 |
| Eil76 | 538 | 1.28 | 0.15 | 3.09 | 2.82 | 2.35 |
| Fri26 | 937 | 0.5 | 0.09 | 0.31 | 0.04 | 0.18 |
| KroA100 | 21282 | 4.47 | 0.33 | 3.78 | 3.42 | 3.61 |
| KroB200 | 29437 | 8.3 | 1.12 | 6.86 | 5.42 | 4.85 |
| Lin105 | 14379 | 5.04 | 0.23 | 4.39 | 3.89 | 3.93 |
| Pr76 | 108159 | 2.82 | 0.3 | 2.26 | 1.31 | 1.78 |
| Average ER | | 3.95 | 0.32 | 3.52 | 2.58 | 2.86 |

Several experiments were conducted using different modified SS algorithms, by error ration computation, we are found that the proposed SS-MCA gives the best error ratio compare with the other 4 modified SS algorithms. Also, a various comparison using different algorithms (Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particles Swarm Optimization (PSO), Artificial Bees Colony (ABC)[28], Camel Herd Algorithm (CHA)[33], Crow Search Algorithm based on Order Crossover operation (CSA-OC) [34] and 2 important methods in [35, 36], by error ration computation, we are found that the proposed SS-MCA gives good error ratio compare with other algorithms, but it is not the best one, by way of TSP results.

Table 5. Results of error ratio for proposed SS-MCA with different algorithms (For 10 Runs)

| Instances | ER for NOPT of Different Algorithms with Our Proposed | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | SS-MCA | GA [28] | ACO [28] | PSO [28] | CHA [33] | CSA-OC [34] | ABC [28] | [35] | [36] |
| A280 | 0.17 | 1.02 | 0.41 | 0.31 | 0.21 | 0.18 | 0.48 | 0.05 | 0.1 |
| Att48 | 0.75 | 1.23 | 0.72 | 0.76 | 2.15 | 0.11 | 0.31 | 0 | 0 |
| Berlin52 | 0.17 | 1.62 | 0 | 1.03 | 0.05 | 0.17 | 0.27 | 0 | 0 |
| Bier127 | 0.36 | 3.04 | 2.12 | 1.25 | 0.31 | 0.39 | 1.25 | 0.064 | 0 |
| Dantzig42 | 0.09 | 1.42 | 0.68 | 0.49 | 0.28 | 0.11 | 1.17 | 0 | 0 |
| Eil51 | 0.08 | 1.71 | 0.43 | 0.2 | 0.13 | 0.11 | 0 | 0 | 0 |
| Eil76 | 0.15 | 2.03 | 0 | 0 | 0.27 | 0.23 | 0 | 0.107 | 0 |
| Fri26 | 0.09 | 1.12 | 0.63 | 0.52 | 0.17 | 0.07 | 0.78 | 0 | 0 |
| KroA100 | 0.33 | 1.92 | 1.25 | 1.37 | 0.49 | 0.33 | 1.08 | 0 | 0 |
| KroB200 | 1.12 | 4.37 | 2.89 | 1.92 | 1.05 | 1.37 | 1.27 | 0.52 | 0.02 |
| Lin105 | 0.23 | 2.67 | 1.08 | 0.93 | 0.87 | 0.51 | 0.36 | 0 | 0 |
| Pr76 | 0.3 | 3.01 | 1.78 | 1.23 | 0.26 | 0.27 | 0.89 | 0 | 0 |
| Average Error | 0.32 | 2.09 | 0.99 | 0.83 | 0.52 | 0.47 | 0.65 | 0.06 | 0.01 |

**5.2 Flexible job shop scheduling problem (FJSSP)**

FJSSP problem is to allocate each process to a machine and to arrange machine operations so which the maximum termination time (makespan) among all processes is reduced. The MCA utilizes equation two as a fitness function to identify the optimum solution. The crucial factor in measuring job scheduling quality is Makespan.

$$Minimize\ C_{max} = Max(C_j), j = 1, 2, \dots \dots n \tag{2}$$

The proposed algorithm evaluated with specific samples of FJSS datasets (HUdata) with a problem group (129)[37]. Such problems are three by[38] and 40 by [39, 40] (la01 to la40). i[37]generated three categories: the first category data is E, the second category data is R, and last data is V. The first category flexibility (1.15), whereas average flexibility 2 for the second category and third category is m/2 (range between 2.50 and 7.50) where the number of the machine is equal.

The quality experimental results depend on the error ratio value compare with the lower bound (LB) of each dataset sample.

The performance of SS-MCA measures when using the same data by comparing the empiricism result of the proposed algorithm with basic Scatter Search (SS), Meerkat Clan Algorithm (MCA) [29], Cuckoo search algorithm[41], Artificial Fish Swarm Algorithm (AFSA) [41] and Camel Herd Algorithm (CHA)[42].

The result of the suggested algorithm compared with the results of the other algorithms presents in table 6. The first field represents the name of the instances dataset whereas the second field represents the lower bound for the instances and the third field represents the result of the suggested SS-MCA. The field showing the result of the suggested algorithms includes the best results from a sequence of Implementations with various parameter values. The results of SS, MCA, CHA, CS, and AFSA are shown in the other fields. The results show that the SS-MCA is better than the original SS, MCA, CHA, and CS, but the AFSA is still the better.

Table 6. Comparison of the best error ratio for 10 runs (In Percentage)

| Instances | LB | SS-MCA | SS | MCA [29] | CHA [42] | CS [41] | AFSA [41] |
|---|---|---|---|---|---|---|---|
| edata_mt06 | 55 | 0 | 0 | 0 | 0 | 0.01 | 0 |
| edata_mt10 | 871 | 0.01 | 0.39 | 0.01 | 0.09 | 0.36 | 0.13 |
| edata_la1 | 609 | 0.14 | 0.47 | 0.28 | 0.45 | 0.19 | 0.01 |
| edata_la2 | 655 | 0.07 | 0.32 | 0.1 | 0.25 | 0.19 | 0.05 |
| edata_la3 | 550 | 0.17 | 0.31 | 0.28 | 0.33 | 0.21 | 0.05 |
| edata_la4 | 568 | 0.11 | 0.48 | 0.23 | 0.46 | 0.24 | 0.06 |
| edata_la5 | 503 | 0.21 | 0.36 | 0.3 | 0.35 | 0.2 | 0.02 |
| edata_la6 | 855 | 0.14 | 0.58 | 0.43 | 0.54 | 0.14 | 0 |
| edata_la7 | 762 | 0.37 | 0.54 | 0.49 | 0.64 | 0.25 | 0.06 |
| edata_la8 | 845 | 0.24 | 0.42 | 0.33 | 0.48 | 0.18 | 0.02 |
| rdata_mt06 | 47 | 0.08 | 0.2 | 0.08 | 0 | 0.17 | 0.02 |
| rdata_mt10 | 679 | 0.17 | 0.52 | 0.26 | 0.19 | 0.57 | 0.24 |
| rdata_la1 | 570 | 0.23 | 0.39 | 0.38 | 0.16 | 0.26 | 0.06 |
| rdata_la2 | 529 | 0.21 | 0.51 | 0.45 | 0.19 | 0.28 | 0.06 |
| rdata_la3 | 477 | 0.33 | 0.53 | 0.48 | 0.23 | 0.3 | 0.06 |
| rdata_la4 | 502 | 0.33 | 0.41 | 0.43 | 0.24 | 0.28 | 0.07 |
| rdata_la5 | 457 | 0.29 | 0.6 | 0.45 | 0.24 | 0.26 | 0.05 |
| rdata_la6 | 799 | 0.25 | 0.54 | 0.56 | 0.3 | 0.21 | 0.03 |
| rdata_la7 | 749 | 0.37 | 0.48 | 0.49 | 0.34 | 0.22 | 0.05 |
| rdata_la8 | 765 | 0.41 | 0.63 | 0.57 | 0.38 | 0.22 | 0.04 |
| vdata_mt06 | 47 | 0.04 | 0.07 | 0.04 | 0 | 0.17 | 0 |
| vdata_mt10 | 655 | 0.18 | 0.41 | 0.3 | 0.09 | 0.52 | 0.18 |
| vdata_la1 | 570 | 0.24 | 0.39 | 0.42 | 0.17 | 0.27 | 0.05 |
| vdata_la2 | 529 | 0.37 | 0.32 | 0.45 | 0.05 | 0.27 | 0.09 |
| vdata_la3 | 477 | 0.33 | 0.36 | 0.48 | 0.2 | 0.31 | 0.06 |
| vdata_la4 | 502 | 0.28 | 0.47 | 0.4 | 0.27 | 0.29 | 0.06 |
| vdata_la5 | 457 | 0.32 | 0.44 | 0.41 | 0.16 | 0.28 | 0.05 |
| vdata_la6 | 799 | 0.38 | 0.49 | 0.51 | 0.3 | 0.22 | 0.03 |
| vdata_la7 | 749 | 0.32 | 0.39 | 0.5 | 0.29 | 0.25 | 0.04 |
| vdata_la8 | 765 | 0.39 | 0.65 | 0.54 | 0.18 | 0.24 | 0.02 |
| Error Ratio | | 0.23 | 0.42 | 0.35 | 0.25 | 0.25 | 0.05 |

.

## 6. Conclusion

The proposed SS-MCA is of modified metaheuristic algorithm, substantially, it is an SS algorithm including important steps from swarm intelligence based MCA. These steps provide more diversity and explore solutions to the SS algorithm. Therefore, the proposed SS-MCA gives more than good solutions compare with both the SS algorithm and MCA. Two NP-hard problems (TSP and FJSSP) have been solved using the proposed algorithm; the experimental results show that the SS-MCA obtains good results compare with original SS and MCA; also the SS-MCA's results are better than other modified scatter search algorithms. Sometimes, the results

of SS-MCA are better than several important metaheuristic algorithms. As future work, it is better to apply the Order Crossover (OC) operation on the SS-MCA to enhance the performance.
paper.

## 7. References


[1] R. K. Sahoo, D. P. Mohapatra, M. R. J. I. J. o. C. S. Patra, and Engineering, "A firefly algorithm based approach for automated generation and optimization of test cases," vol. 4, no. 8, pp. 54-58, 2016.

[2] X.-S. Yang, "Metaheuristic optimization: algorithm analysis and open problems," in *International Symposium on Experimental Algorithms*, 2011, pp. 21-32: Springer.

[3] X. Wang and J. J. P. E. Tian, "Dynamic programming for NP-hard problems," vol. 15, pp. 3396-3400, 2011.

[4] T. Narwadi and Subiyanto, "An application of traveling salesman problem using the improved genetic algorithm on android google maps," in *AIP Conference Proceedings*, 2017, vol. 1818, no. 1, p. 020035: AIP Publishing LLC.

[5] J. J. L. Žerovnik and S. Transport, "Heuristics for NP-hard optimization problems-simpler is better!?," vol. 6, no. 1, pp. 1-10, 2015.

[6] B. Mohammed, R. Chisab, and H. Alrikabi, "Efficient RTS and CTS Mechanism Which Save Time and System Resources," 2020.

[7] R. Martí, Corberán, Á., and Peiró, J. , *Handbook of heuristics.* Springer,Cham, 2015

[8] F. C. G. López, M. G. Torres, J. A. M. Pérez, and J. M. M. Vega, "Scatter search for the feature selection problem," in *Conference on Technology Transfer*, 2003, pp. 517-525: Springer.

[9] N. Mansour, I. Ghalayini, S. Rizk, and M. El-Sibai, "Evolutionary algorithm for predicting all-atom protein structure," 2011.

[10] A. M. Sagheer, A. T. Sadiq, and M. S. Ibrahim, "Improvement of scatter search using Bees Algorithm," in *2012 6th International Conference on Signal Processing and Communication Systems*, 2012, pp. 1-7: IEEE.

[11] T. Zhang, W. A. Chaovalitwongse, Y. J. C. Zhang, and O. Research, "Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries," vol. 39, no. 10, pp. 2277-2290, 2012.

[12] A. T. Sadiq, A. M. Sagheer, and M. S. J. I. J. o. R.-b. I. S. Ibrahim, "Improved scatter search for 4–colour mapping problem," vol. 4, no. 4, pp. 221-226, 2012.

[13] A. T. S. J. I. J. o. A. R. i. A. I. Al-Obaidi, "Improved scatter search using cuckoo search," vol. 2, no. 2, pp. 61-67, 2013.

[14] M. A. Remli, M. S. Mohamad, S. Deris, R. Sinnott, and S. J. C. P. Napis, "An Improved Scatter Search Algorithm for Parameter Estimation in Large-Scale Kinetic Models of Biochemical Systems," vol. 16, no. 5, pp. 427-438, 2019.

[15] Z. Zhang, L. Mao, C. Guan, L. Zhu, and Y. J. S. C. Wang, "An improved scatter search algorithm for the corridor allocation problem considering corridor width," vol. 24, no. 1, pp. 461-481, 2020.

[16] N. Hussien, I. Ajlan, M. M. Firdhous, and H. Alrikabi, "Smart Shopping System with RFID Technology Based on Internet of Things," 2020.

[17] R. Baños, C. Gil, J. Reca, J. J. C. O. Martínez, and Applications, "Implementation of scatter search for multi-objective optimization: a comparative study," vol. 42, no. 3, pp. 421-441, 2009.

[18] H. T. S. Al-Rikabi, *Enhancement of the MIMO-OFDM Technologies*. California State University, Fullerton, 2013.

[19] H. R. Bdulshaheed, Z. T. Yaseen, and I. I. Al-barazanchi, “New approach for Big Data Analysis using Clustering Algorithms in Information,” Jour Adv Res. Dyn. Control Syst., vol. 2, no. 4, pp. 1194–1197, 2019.

[20] I. A. Aljazaery, H. T. S. Alrikabi, and M. R. J. i. Aziz, "Combination of Hiding and Encryption for Data Security," vol. 14, no. 9, p. 35, 2020.

[21] J. A. Egea, E. Vazquez, J. R. Banga, and R. J. J. o. G. O. Martí, "Improved scatter search for the global optimization of computationally expensive dynamic models," vol. 43, no. 2-3, pp. 175-190, 2009.

[22] R. Martí, M. Laguna, and F. J. e. J. o. o. R. Glover, "Principles of scatter search," vol. 169, no. 2, pp. 359-372, 2006.
[23] H. T. Y. J. E. J. o. O. R. Yoshizaki, "Scatter search for a real-life heterogeneous fleet vehicle routing problem with time windows and split deliveries in Brazil," vol. 199, no. 3, pp. 750-758, 2009.
[24] M. A. a. Roa'a, I. A. Aljazaery, S. K. Al_Dulaimi, H. T. S. J. B. o. E. E. Alrikabi, and Informatics, "Generation of High Dynamic Range for Enhancing the Panorama Environment," vol. 10, no. 1, 2020.
[25] A. Alaidi, I. Aljazaery, H. Alrikabi, I. Mahmood, and F. Abed, "Design and Implementation of a Smart Traffic Light Management System Controlled Wirelessly by Arduino," 2020.
[26] B. Durakovic and H. Basic, "Continuous Quality Improvement in Textile Processing by Statistical Process Control Tools: A Case Study of Medium-Sized Company," Periodicals of Engineering and Natural Sciences, vol. 1, no. 1, pp. 39-46, 2013
[27] N. S. Alseelawi, E. K. Adnan, H. T. Hazim, H. Alrikabi, and K. Nasser, "Design and Implementation of an E-learning Platform Using N-Tier Architecture," 2020.
[28] A. T. S. Al-Obaidi, H. S. Abdullah, Z. O. J. I. J. o. E. E. Ahmed, and C. Science, "Meerkat clan algorithm: A new swarm intelligence algorithm," vol. 10, no. 1, pp. 354-360, 2018.
[29] A. T. Sadiq, H. S. Abdullah, and Z. O. J. I. J. o. S. Ahmed, "Solving Flexible Job Shop Scheduling Problem Using Meerkat Clan Algorithm," pp. 754-761, 2018.
[30] O. H. Yahya, H. Alrikabi, I. A. J. I. J. o. O. Aljazaery, and B. Engineering, "Reducing the Data Rate in Internet of Things Applications by Using Wireless Sensor Network," vol. 16, no. 03, pp. 107-116, 2020.
[31] C. Perera, M. Barhamgi, A. K. Bandara, M. Ajmal, B. Price, and B. J. I. S. Nuseibeh, "Designing privacy-aware internet of things applications," vol. 512, pp. 238-257, 2020.
[32] A. T. S. J. I. J. o. S. Al-Obaidi, "Harmony-Scatter Search to Solve Travelling Salesman Problem," vol. 54, no. 4, pp. 965-974, 2013.
[33] Z. O. Ahmed, A. T. Sadiq, and H. S. Abdullah, "Solving the Traveling Salesman's Problem Using Camels Herd Algorithm," in *2019 2nd Scientific Conference of Computer Sciences (SCCS)*, 2019, pp. 1-5: IEEE.
[34] S. A. Hussein, A. T. S. J. J. o. C. Al-Obaidi, and T. Nanoscience, "An Improved Crow Search Algorithm for Solving the Traveling Salesman Problem," vol. 16, no. 3, pp. 978-981, 2019.
[35] J. a. R. Gottlieb, Günther R., " “Evolutionary Computation in Combinatorial Optimization,”" pp. 44-46., 2006. Springer-Verlag
[36] G. Gutin and A. P. Punnen, *The traveling salesman problem and its variations*. Springer Science & Business Media, 2006.
[37] M. J. v. Staaden, "Suricata suricatta," *Mammalian Species,* pp. 1--8, 1994.
[38] J. Hurink, B. Jurisch, and M. J. O.-R.-S. Thole, "Tabu search for the job-shop scheduling problem with multi-purpose machines," vol. 15, no. 4, pp. 205-215, 1994.
[39] F. H. a. T. G.L., ""Probabilistic learning combinations of local job shop scheduling rules"," *In Industrial Scheduling,J.F. Muth and G.L. Thompson (Eds),* pp. 225–251, 1993. Englewood Cliffs, NJ: Prentice-Hall.
[40] A. Alaidi, O. Yahya, and H. Alrikabi, "Using Modern Education Technique in Wasit University," 2020.
[41] Ahmed T. Sadiq Al-Obaidi and Samer Alaa Hussein, ""An Improved Artificial Fish Swarm Algorithm to Solve Flexible Job Shop Scheduling"," *New Trends in Information & Communications Technology Applications (NTICT).* 2017.
[42] H. S. A. A. T. Sadiq Al-Obaidi, and zied O. Ahmed, " “Camel Herds Algorithm: a New Swarm Intelligent Algorithm to Solve Optimization Problems”,," *IJPCC,,* vol. vol. 3, no. 1, May 2017.

**Acknowledgments**

This research was supported by a research scholarship from the Faculty of Computer Science, Technology University and Faculty Education, Al-Mustansiriya University. Special thanks to Prof.Dr. Ahmed Tariq Sadeq