



Genetic Algorithm Based on K-means-Clustering Technique for Multi-objective Resource Allocation Problems

Mai A. Farag^{1*}, M. A. El-Shorbagy¹, I. M. El-Desoky¹, A. A. El-Sawy^{1,2}
and A. A. Mousa^{1,3}

¹Department of Basic Engineering Science, Faculty of Engineering, Menoufiya University, Egypt.

²Department of Mathematics, Faculty of Science, Qassim University, Saudi Arabia.

³Department of Mathematics, Faculty of Sciences, Taif University, Saudi Arabia.

Authors' contributions

This work was carried out in collaboration between all authors. Authors MAF, MAES and AAM designed the study and wrote the first draft of the manuscript and managed literature searches. Authors MAES and AAM managed the analyses of the study. Authors IMED, AAES and AAM read and approved the final manuscript.

Article Information

DOI: 10.9734/BJAST/2015/16570

Editor(s):

(1) Qing-Wen Wang, Department of Mathematics, Shanghai University, P.R. China.

Reviewers:

(1) Anonymous, Turkey.

(2) Anonymous, Malaysia.

(3) Ravi Narayan, Computer Science and Engineering Department, Thapar University, India.

Complete Peer review History: <http://www.sciencedomain.org/review-history.php?iid=1069&id=5&aid=8542>

Original Research Article

Received 6th February 2015

Accepted 9th March 2015

Published 20th March 2015

ABSTRACT

This paper presents genetic algorithm based on K-means clustering technique for solving multi-objective resource allocation problem (MORAP). By using k-means clustering technique, population can be divided into a specific number of subpopulations with dynamic size. In this way, different GA operators (crossover and mutation) can be applied to each subpopulation instead of one GA operators applied to the whole population. The purpose of implementing K-means clustering technique is preserving and introducing diversity. Also it enable the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other. Two test problems taken from the literature are used to compare the performance of the proposed approach with the competing algorithms. The results have been demonstrated the superiority of the proposed algorithm and its capability to solve MORAP.

*Corresponding author: E-mail: mai.farag2015@gmail.com;

Keywords: Multi-objective resource allocation problems; genetic algorithm; K-means clustering technique; optimization.

1. INTRODUCTION

Resource allocation is part of resource management. It is used to assign the available resources in an economic way. In project management, resource allocation is the scheduling of activities and the resources required by those activities while taking into consideration both the resource availability and the project time [1].

MORAP is the process of allocating resources among the various projects or business units to meet the expected objectives. Resources may be manpower, dominions, raw materials, capital or anything else in limited supply which can be used to accomplish the goals. The goals may be objectives or targets (i.e., maximizing profits, minimizing costs, or achieving the best possible quality) [2]. MORAP has a variety of applications such as:

- 1- Allocating Marketing Resources [3]: A key responsibility of marketing managers is to decide the allocation of scarce marketing resources, e.g., advertising dollars, selling hours, retail shelf-space or merchandise inventories.
- 2- Stochastic network systems [4]: for getting the optimal performance of the network by maintaining both resource and duration of activities in proper realization.
- 3- Scheduling of Worker Allocation in the Manual Labor environment [5]: for assigning the workers into the jobs to reduce human cost, to shorten production duration and control production overwork.
- 4- Portfolio optimization [6]: for creating efficient portfolios on allocating funds to stocks or bonds to maximize return for a given level of risk, or to minimize risk for a target rate of return.
- 5- Health care resource allocation [7]: Discounting costs and health benefits in cost-effectiveness analysis for the health care resource allocation. And many other application that can be formulated as resource allocation problem [8].

Traditionally, MORAP have been solved using methods in operations research (OR), Integer Programming (ID) [9], branch-and-bound (B & B) [10], and Dynamic Programming (DP) [11]. None of these methods are computationally tractable

for any real-life problem size, thus rendering them impractical [12].

In recent years, there has been a growing interest in studying evolutionary algorithms (EAs) for many real world optimization problems, which are conceptually different from the traditional mathematical programming techniques. These methods are based on certain biological, molecular, and neurological phenomena. The reasons for their popularity are many [13]:

- (i). EAs do not require any derivative information
- (ii). EAs are relatively simple to implement
- (iii). EAs are flexible and have a wide-spread applicability

One of the main advantages of these EAS is finding high quality or near-optimal solutions with reasonable computational times. Recently, several evolutionary algorithms have been developed for MORAP including genetic algorithm (GA) [14], genetic algorithm with fuzzy inference [15], A hybrid simulated annealing approach [16], ant colony optimization (ACO) [2], variable neighborhood search [17], a memetic algorithm based on node-weighted graphs [18], hybrid particle swarm optimization [19], tabu search [20], differential evolutionary algorithm [21], multi-objective differential evolution based on weighted normalized sum (WNS-MODE) approach [22], co-evolutionary hyper-heuristic method [23], combining exhaustive search with evolutionary computation method [24], Pareto multistage decision-based genetic algorithm (P-mdGA) [25], an improved quantum evolutionary algorithm (QEA) [26].

Clustering is a process of division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups [27]. There are many clustering algorithms [28]. The k-means is possibly the most commonly-used clustering algorithm because of its simplicity and accuracy. The k-means finds a locally optimal solution by minimizing a distance measure between each data and its nearest cluster center [29]. Several algorithms have been proposed in the literature for clustering: The Iterative Self-Organizing Data Analysis Technique (ISODATA) [30], Clustering Large Applications based up on Randomized

Search (CLARANS) [31], Parallel-cluster (p-cluster) [32], Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [33] and Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) [34]. The most commonly and widely used clustering is K-Means because of its simplicity and accuracy [29].

In particular, genetic algorithms (GA) produce good results compared to other techniques in solving resource allocation problem [35]. The major disadvantage of GA, though, is that they easily become trapped in the local minima. In this paper, k-means clustering technique is used along with a GA in order to avoid the local minima problem and to benefit from the advantages of dynamic clustering algorithms. By using K-means clustering algorithm, population can be divided into a specific number of subpopulations. Within each cluster, subpopulation has common features. After grouping all individuals into pre-defined number of clusters, instead of taking care of all individuals, population can regard the huge amount of individuals as just the number of groups that has been divided. In this way, different GA operators can apply to subpopulations instead of one GA operator applied to all population.

This paper is organized as follows. In section 2, we describe the mathematical formulation of the MORAP. In sections 3, genetic algorithm is briefly introduced. In section 4, clustering technique is briefly introduced. Our approach is proposed and explained in detail in section 5. Numerical simulation is presented in section 6. Finally, we conclude the paper in section 7.

2. MATHEMATICAL FORMULATION

The general form of the MORAP is as follows [14]:

$$\begin{aligned}
 \text{Max } Z_1(x_1, x_2, \dots, x_n) &= \sum_{k=1}^n z_k^1(x_k) \\
 \text{Max } Z_2(x_1, x_2, \dots, x_n) &= \sum_{k=1}^n z_k^2(x_k) \\
 &\dots\dots\dots \\
 \text{Max } Z_q(x_1, x_2, \dots, x_n) &= \sum_{k=1}^n z_k^q(x_k) \quad (1) \\
 \text{Subject to : } \sum_{k=1}^n g_k(x_k) &\leq S, \\
 g_k(x_k) &\geq 0, \\
 x_k &\geq 0;
 \end{aligned}$$

where x_k define (decision variables), S donates the (resources), $g_k(x_k)$ represents the activities stages and $Z_q(x_1, x_2, \dots, x_n)$ is the q objective function. The process of MORAP seek to find an optimal allocation (decision variables) of limited amount of resource to a number of tasks (activities stages) for optimizing their objective subjected to the given resource constraint [14].

3. GAPROCEDURE

GAs operates on a population of candidate solutions encoded to finite bit string called chromosome. In order to acquire optimality, each chromosome exchanges information by using operators borrowed from natural genetic to produce the improved solution [36,37].

We can describe the steps of basic genetic operators as follow [38]:

- Step 0 (Initialization stage): Randomly generate an initial population.
- Step 1 (Evaluation of non-dominated solutions): Calculate the values of q objective function for each individual.
- Step 2 (Selection operators): Selection operators are used to select the individuals to which the crossover operators will be applied.
- Step 3 (Crossover operators): In this step, crossover operators are applied to each pair selected to generate offspring
- Step 4 (Mutation operators): Apply mutation operation to each offspring generated by the crossover operation.
- Step 5 (Stopping criterion): check the stopping criterion, if it is not reached, then go to Step 1, otherwise, go to the next step.
- Step 6 (DM selection): DM selects the most preferred among the alternatives.

The basic step of GA is illustrated in flowchart as shown in Fig. 1.

4. CLUSTERING ALGORITHM

Clustering is process of Finding groups of objects such that the objects in a group will be similar (or related) to each other and different from (or unrelated to) the objects in other groups [27]. K-means [29] is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a

simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group is done. At this point we need to re-calculate k new centroids as centers of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. The loop has been generated, as a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. A centroid is defined as the point whose coordinates are obtained by computing the average of each of the coordinates (i.e., feature values) of the points of the cluster. Formally, the k -means clustering algorithm follows the following steps (taken from [40]):

- Step 1: Define a number of desired clusters, k .
- Step 2: Choose initial cluster centroid randomly. These represent the “temporary” means of the clusters.
- Step 3: Compute the squared Euclidean distance (sum of square error) from each object to each cluster and each object is assigned to the closest cluster as follows:

$$\text{Sum of Squire Error (SSE)} = \sum_{i=1}^k \sum_{x \in C_i} \text{dist}^2(m_i, x); \quad (2)$$

Where x is a data point in cluster C_i and m_i is the centroid of cluster C .

- Step 4: For each cluster, the new centroid is computed, and each centroid value is now replaced by the respective cluster centroid.
- Step 5: Repeat steps 3 and 4 until no point changes its cluster.

Fig. 2 shows an illustration of K-means algorithm on a 2-dimensional dataset with three clusters.

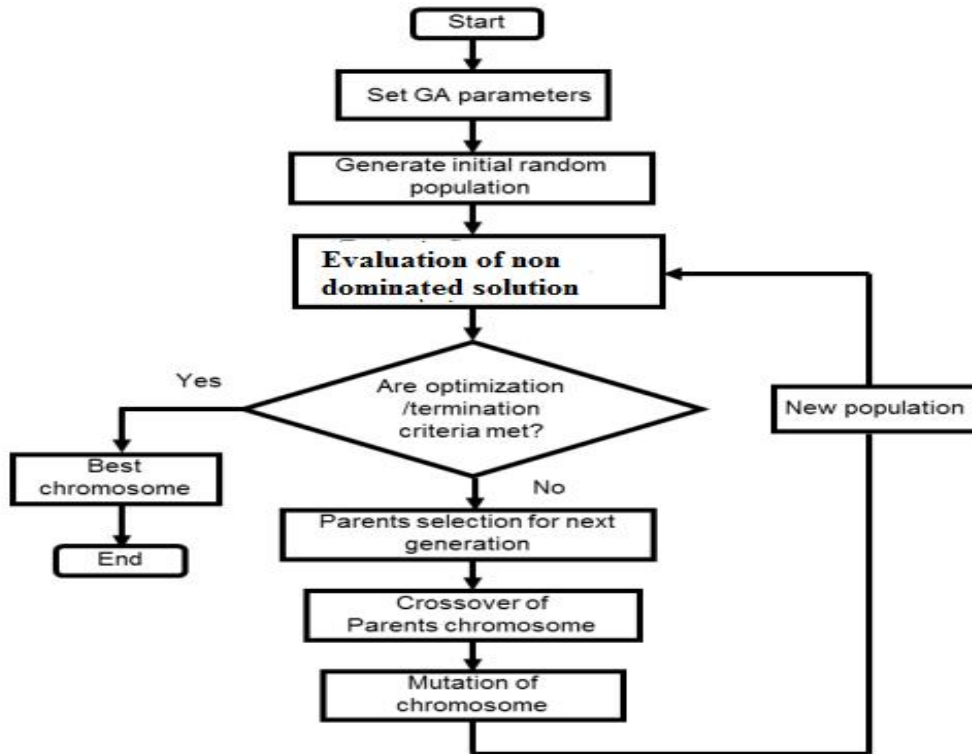


Fig. 1. Main flowchart of GA (Taken from [39])

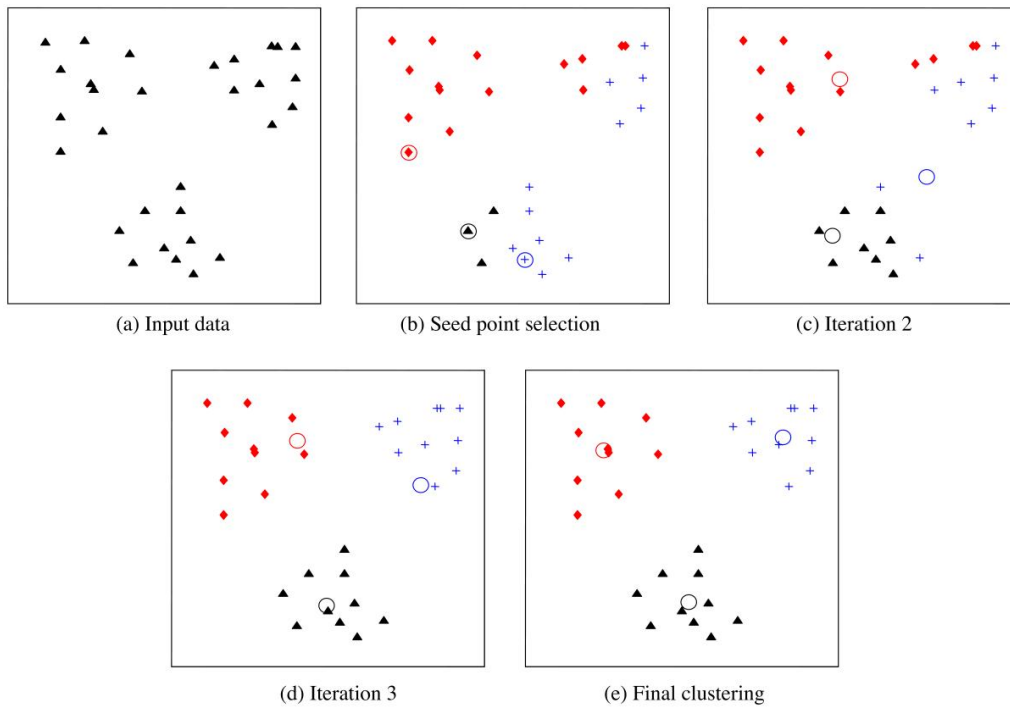


Fig. 2. Illustration of K-means algorithm. (a) Two-dimensional input data with three clusters; (b) three centroid points selected as cluster centers and initial assignment of the data points to clusters; (c) & (d) intermediate iterations updating cluster labels and their centers; (e) final clustering obtained by K-means algorithm at convergence. (Taken from [41])

5. GENETIC ALGORITHM BASED ON K-MEANS-CLUSTERING TECHNIQUE

In this paper, we propose GA based on k-means clustering technique to solve MORAP. Where, population can be divided into a specific number of groups (subpopulations) by clustering technique. Within each cluster, subpopulation has common features. After grouping all individuals into pre-defined number of clusters, instead of taking care of all individuals, population can regard the huge amount of individuals as just the number of groups that has been divided. In this way, different GA operators can be applied to sub population instead of one GAs operator applied to all population. The main steps of the proposed algorithm are described as follows:

5.1 Initialization Stage

In this paper, we reformulate MORAP as network model, where limited supply represented by stages as shown in Fig. 3. We consider the human resource allocation problem with multi-objectives for minimize the total cost and

maximize the total efficiencies of the human allocation decision. It is required to find a path between two nodes source node (S) and terminal node (T) having minimum total cost and maximum efficiency [14]. A path from node S to node T is a sequence of arcs $(S, x_{1m}), (x_{1m}, x_{2m}), \dots, (x_{n-1m}, T)$. A path can be equivalently represented as sequence of nodes $(S, x_{1m}, x_{2m}, \dots, x_{n-1m}, T)$; where $m = 0, 1, 2, \dots, m$ and n is number of stage (district), thus the structure of each individual (chromosome) can express a path, i.e., each chromosome is represented as sequence of nodes.

In this step, the algorithm generates an initial population containing $Npop$ strings. We use a random selection element from available number in each stage. For example, in the MORAP of allocating 10 worker to a certain set of 4 jobs (4 stages), the multi-objective resource allocation considers determining a vector of allocation path in the 11 states and the 4 stages under the minimum costs and the maximum efficiency as shown in Fig. 4. Fig. 5 shows the structure of the chromosome for four stage allocation path.

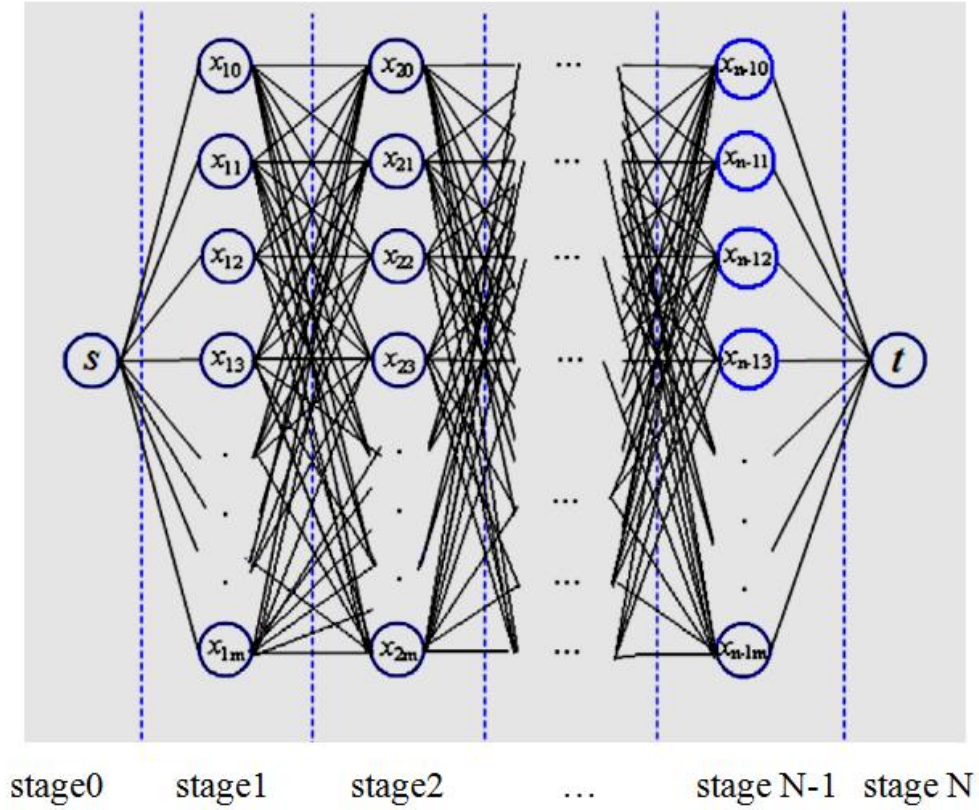


Fig. 3. Representation MORAP as network model

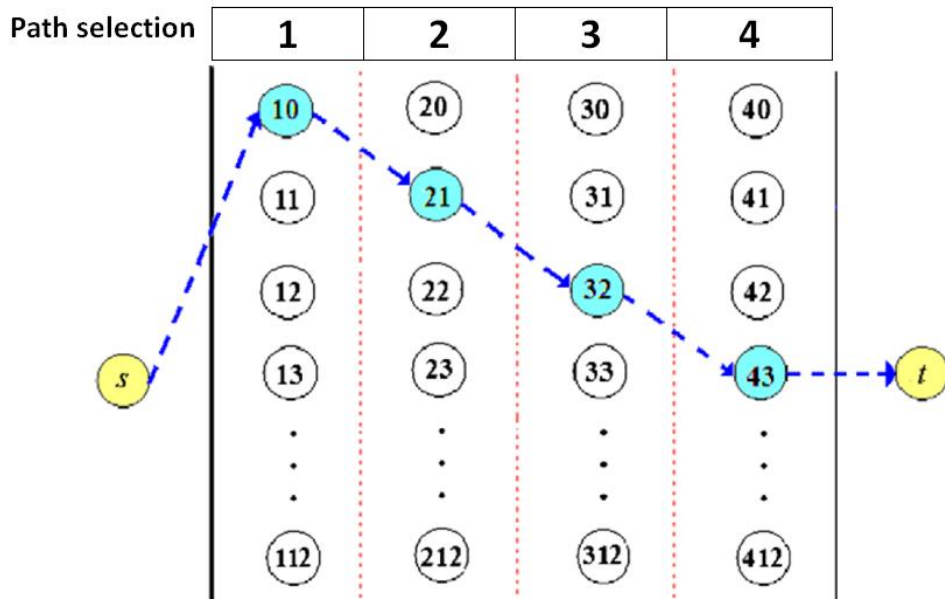


Fig. 4. The figure of optimal path allocation

Stage :	1	2	3	4
Path Selection	3	2	5	0

Fig. 5. The structure of the chromosome for four-stage allocation path

5.2 Rejection of Illegal Individuals

The chromosome structure may produce some illegal individuals, which do not satisfy the feasible region of the problem, for example, if there are N workers needed to allocate to M jobs, the number of worker must not exceed N workers [14]. So before the genetic operation for every time, we should eliminate these illegal individuals which do not satisfy the conditions mentioned above. In this step, the illegal individuals are simply eliminated and replaced by randomly drawn new legal individuals.

5.3 Evaluation of Non-dominated Solutions

In multi-objective optimization problem, there does not exist a single solution that simultaneously optimizes each objective. In that case, the objective functions are said to be discordant and there exists a (possibly infinite) number of solutions. For such solutions, called non-dominated solutions, Pareto optimal solutions, Pareto efficient solutions or noninferior solutions [42]. So it is necessary to find the Pareto optimal solutions by classifying a population according to non-domination.

5.3.1 Definition (Pareto optimal solution)

x^* is said to be a Pareto optimal solution of multi-objective optimization problem if there exists no other feasible x (i.e., $x \in S$) such that, $f_j(x) \leq f_j(x^*)$ for all $j = 1, 2, \dots, q$ and $f_j(x) < f_j(x^*)$ for at least one objective function f_j .

The image of the Pareto set, i.e., the image of all the Pareto solutions, is called Pareto front or Pareto curve or surface. A population can be evaluated according to non-domination criteria. Consider any two solutions x_1 and x_2 for a problem having more than one objective function $q (q > 1)$ objective function values, may have one of these possibilities, one of these solution can be dominates the other or non-dominates the other. The solution x_1 is said to dominate other

solution x_2 if the two following condition are true [43].

- 1- The solution x_1 is no worse (say the operator (\prec) denotes worse and (\succ) denotes better) than x_2 in all objectives, or $f_k(x_1) \prec f_k(x_2)$ for all $k = 1, \dots, q$ objectives.
- 2- The solution x_1 is strictly better than x_2 in at least one objective, or $f_k(x_1) \succ f_k(x_2)$ for at least one $k \in \{1, 2, \dots, q\}$.

Consider a set of population members, each having $q (q > 1)$ objective function values. The following procedure is used to find the non-dominated set of solutions [44].

- Step 0: Begin with $m = 1$.
- Step 1: For all $n = 1, 2, \dots, N_{POP}$ and $m \neq n$, compare solutions x_m and x_n for domination using the previous conditions for all q objectives.
- Step 2: If for any n , x_m is dominated by x_n , mark x_m as 'dominated', and it is inefficient.
- Step 3: If all solutions (that is, when $m = N_{POP}$ is reached) in the set are considered, Go to Step 4, else increment m by one and Go to Step 1.
- Step 4: All solutions that are not marked 'dominated' are non-dominated solutions.

All these non-dominated solutions are assumed to constitute the non-dominated front in the population in a specified generation.

5.4 Selection Stage

The main purpose of the selection stage (parent selection stage) is to determine which individuals are best suited to have children and pass their chromosomes to the next generation according to their fitness. Better the fitness, the bigger chance to be selected to be the parent [39] Selection (reproduction) operator is intended to improve the average quality of the population by

giving the high quality chromosomes a better chance to get copied into the next generation. The selection directs GA search towards promising regions in the search space.

Murata et al. [45] proposed a dynamic-weight approach to obtaining fitness function. In weighted sum approach, multiple objective functions combined into a scalar fitness solution. The weighed sum objective is given as follows:

$$f(x) = w_1 \cdot f_1(x) + \dots + w_i \cdot f_i(x) + \dots + w_q \cdot f_q(x); \quad (3)$$

where x is a string (i.e., individual), $f(x)$ is a combined fitness function, $f_i(x)$ is the i th objective function and w_i is a constant weight for $f_i(x)$, where w is a weighting-vector with $w_i \geq 0$ for all $i = 1, \dots, q$ and $\sum_{i=1}^q w_i = 1$. In general, the value of each weight can determine randomly. For a multi-objective optimization problem with q objective functions, we can assign a random real number to each weight as follows:

$$w_i = \frac{rand_i}{\sum_{j=1}^q rand_j} = 1, \quad i = 1, 2, \dots, q; \quad (4)$$

where $rand_i$ and $rand_j$ are non-negative random real numbers.

In this paper, Binary Tournament selection [46] is used. In the Binary Tournament selection, two individuals are chosen at random and the better of the two individuals is selected and copied in mating pool.

5.5 K-means Clustering Technique

In order to keep diversity and to avoid trapping in local minima, the k-means cluster algorithm was implemented. In this step, the population in mating pool is split into K separated subpopulations with dynamic size, as illustrated in Fig. 6.

5.6 Genetic Algorithms (GAs) Operators

GA uses two operators to generate new solutions from existing ones: crossover and mutation. In this step, different GA operators applied to subpopulations instead of one GA operator applied to the whole population. In a simple GA, there is only one population in each

generation and all the genetic operations are applied on it, but here, population is split into separated subpopulations, so all the genetic operations, including, crossover and mutation on each subpopulation are performed separately to create new generations.

5.6.1 Crossover operator

The goal of crossover is to exchange information between two chromosomes in order to produce two new offspring for the next population [47]. In our study we used common crossover techniques in a GA. A brief explanation of these techniques is given below.

5.6.1.1 One-point crossover

In one point crossover operator that randomly selects a crossover point within a chromosome then interchanges the two parent chromosomes at this point to produce two new offspring [48].

5.6.1.2 Two-point crossover

In two-point crossover operators two different cut-off points were randomly selected. New offspring were obtained by relocating zones between the cut-off points of parent chromosome [48].

5.6.1.3 Uniform crossover

In this method, a random binary string is generated with the same-size of chromosome. Then relative genes under this binary string between parents is exchanged, where parent strings exchange their bit at the position where the corresponding position in random binary string is 1. Otherwise, no exchange of bit is performed [49].

5.6.1.4 Cross crossover

In such crossover, the new offspring are selected from different parts of parent; often the produced generation is very different from their parent. A cut point is selected in the middle point in the two parent, the left side gene of the cut point of the first parent are copied directly to right side of the new first offspring. Then, the remaining portion of offspring array is selected from left side gene of the cut point of the second parent. In order to produce the second offspring the parents are swapped in this process as pervious manner [50].

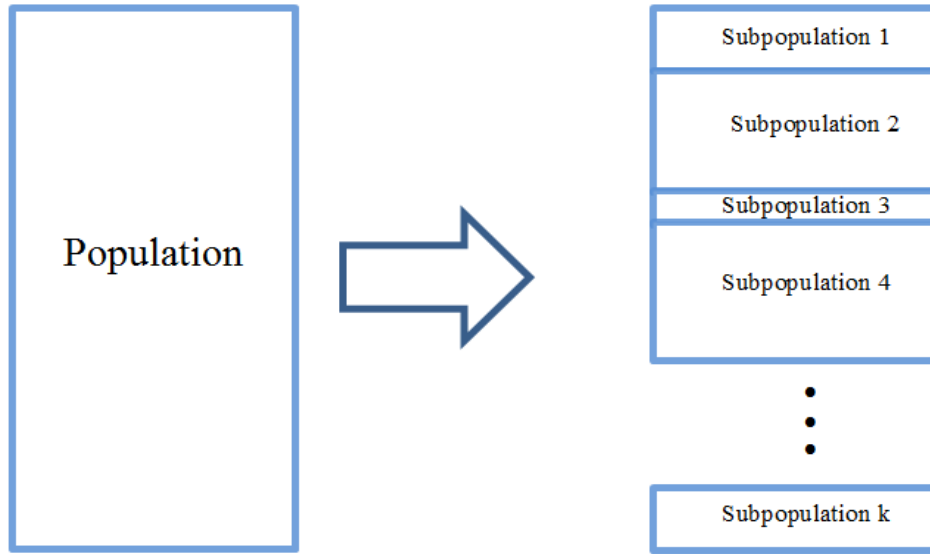


Fig. 6. The population is split into K separated subpopulations with dynamic size

5.6.2 Mutation operator

Mutation operator is one of the GA operators that used to produce new chromosomes or modify some features of them depending on some small probability value. The objective of this operator is to prevent falling of all solutions in population into a local optimum of solved problem. This operator is applied to each offspring in the population with a predetermined probability [47]. In our study we used common mutation techniques in a GA. A brief explanation of these techniques is given below.

5.6.2.1 Twors mutation

In Twors mutation [51], the two genes randomly chosen to exchange of their position, as shown in Fig. 7.

5.6.2.2 Reverse sequence mutation (RSM)

In the reverse sequence mutation operator, we take a sequence S limited by two positions i and j randomly chosen, such that $i < j$. The gene order in this sequence will be reversed by the

same way as what has been covered in the previous operation. Fig. 8 shows the implementation of this mutation operator.

5.6.2.3 One point mutation

In one point mutation, data at a particular point is mutated (a particular gene was randomly selected and then it was replaced with a random state from the available set) [48].

5.6.2.4 Centre inverse mutation (CIM)

The chromosome is divided into two sections. All genes in each section are copied and then inversely placed in the same section of a child [51]. Fig. 9 shows the implementation of this mutation operator.

5.7 Combination Stage

In combination stage, all subpopulations are combined together again to create a new population, as illustrated in Fig. 10.



Fig. 7. Twors mutation



Fig. 8. Reverse sequence mutation



Fig. 9. Centre inverse mutation

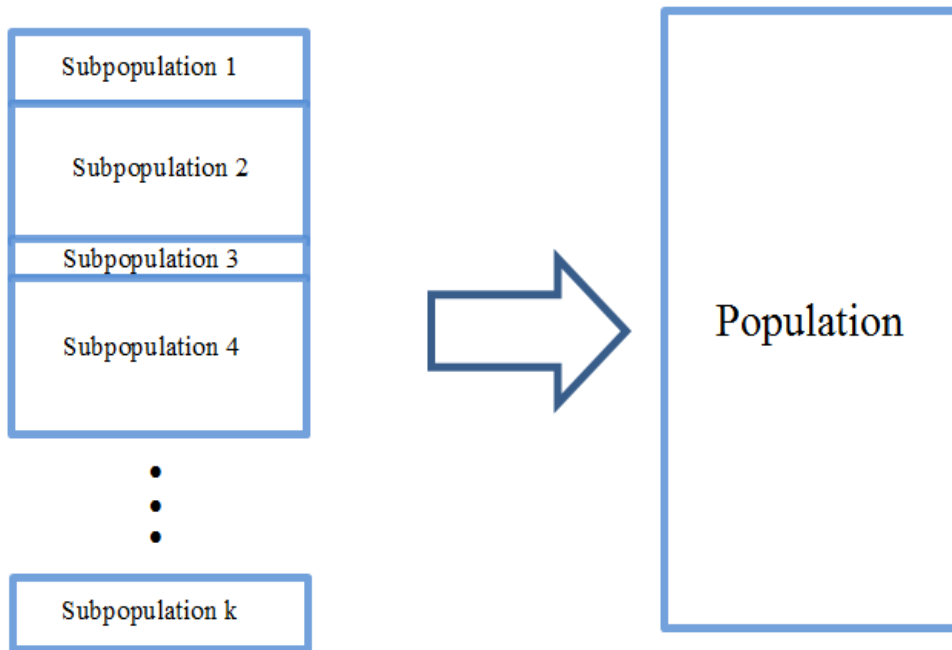


Fig. 10. Commination stage

5.8 Update the Archive of Non-dominated Solution

The algorithm maintains an external archive of non-dominated solutions which gets iteratively updated in the presence of new solutions based on the concept of non-domination. The major function of the archive is to store a historical record of the non-dominated solutions found along the heuristic search process [52]. During the implementation of the algorithm a tentative set of non-dominated solutions is stored in archive which gets iteratively updated every

generation. Algorithm 1 show the procedure (Iterative search procedure) which is used to update the archive of non-dominated solutions [53]. The purpose of the function $p^{(t)}$ is to generate a new search point in each iteration t .

The function $A^{(t)}$ updates the old archive set $A^{(t-1)}$ using the contents of $p^{(t)}$ and the old archive set $A^{(t-1)}$ and determines the update archive set.

Algorithm 1: Iterative search procedure

1. $t \triangleq 0$
 2. $A^{(0)} = 0$
 3. *While terminate* ($A^{(t)}$, t) = *false do*
 4. $t \triangleq t + 1$
 5. $p^{(t)} \triangleq \text{generate}(\)$ {generate new search point}
 6. $A^{(t)} \triangleq \text{update}(A^{(t-1)}, p^{(t)})$ {update archive }
 7. *End while*
 8. *Output* : $A^{(t)}$
-

6. EXPERIMENTAL RESULTS

In this section, the proposed approach has been applied to multi-objective resource allocation test problems taken from the literature [14,38] to verify the performance of the proposed algorithm and illustrate its ability for solving such problems. The test problems have been executed on an Intel core i5, 2.6 GHz processor. The proposed approach is coded using MATLAB programming language. The parameter setting used for all runs are depicted in Table 1.

6.1 Test Problem 1

Test problem 1 (taken from [14]) of allocating 6 workers to a certain set of 4 jobs. Table 2 provides the expected cost and efficiency.

Tables 3 and 4 present the efficient solution obtained by the proposed algorithm at k=4 and k=1, respectively. While Table 5 presents the results obtained by the effective genetic algorithm [14]. Furthermore Fig. 11 shows the

efficient solution by the proposed algorithm at k=4, k=1 and the effective genetic algorithm [14].

It is clear from previous comparison in test problem 1 that the results obtained by the proposed algorithm at k=1, k=4 and the results obtained by effective genetic algorithm is the same values.

Table 1. The proposed algorithm parameters

Parameter	Values	
	Prolem1	Prolem2
Population size	100	500
Crossover rate	0.65	0.95
Mutation rate	0.02	0.02
Iteration	30	200
Number of cluster (k)	1&4	1&4

6.2 Test Problem 2

Test problem 2 (taken from [38]) of allocating 10 workers to a certain set of 4 jobs is solved. Table 6 provides the expected cost and efficiency.

Table 2. The expected cost and efficiency of test problem 1

Number of worker	JOB							
	1		2		3		4	
	Cost	Efficiency	Cost	Efficiency	Cost	Efficiency	Cost	Efficiency
0	70	0	90	0	85	0	130	0
1	60	25	60	20	60	33	115	13
2	50	42	50	38	50	43	100	24
3	40	55	40	54	55	47	100	32
4	40	63	30	65	40	50	90	39
5	45	69	20	73	30	52	80	45
6	50	74	25	80	25	53	80	50

Tables 7 and 8 presents the efficient solution obtained by the proposed algorithm at k=4 and k=1, respectively. While Table 9 present the results obtained by multistage decision-based genetic algorithm [38]. Furthermore Fig. 12 shows the simulation results of the proposed algorithm at different values of k and multistage decision-based genetic algorithm.

Table 3. Efficient solution of the proposed algorithm at k=4

Efficient solution				Overall cost	Overall efficiency
X1	X2	X3	X4		
1	2	1	2	270	120
2	2	1	1	275	126
2	3	1	0	280	129

As shown in Fig. 12 the results obtained by the proposed algorithm with k=4 is dominated to the most of the results obtained by multistage decision-based genetic algorithm. In addition, the result obtained by the proposed algorithm with clustering technique (i.e. k=4) is dominant to the

results obtained by the algorithm without clustering data (i.e. k=1). So the proposed algorithm is more dominant position to be used as an explanation to the problems in hand and different MORAPs.

Table 4. Efficient solution of the proposed algorithm at k=1

Efficient solution				Overall cost	Overall efficiency
X1	X2	X3	X4		
1	2	1	2	270	120
2	2	1	1	275	126
2	3	1	0	280	129

Table 5. Efficient solution of the effective genetic algorithm approach [14]

Efficient solution				Overall cost	Overall efficiency
X1	X2	X3	X4		
1	2	1	2	270	120
2	2	1	1	275	126
2	3	1	0	280	129

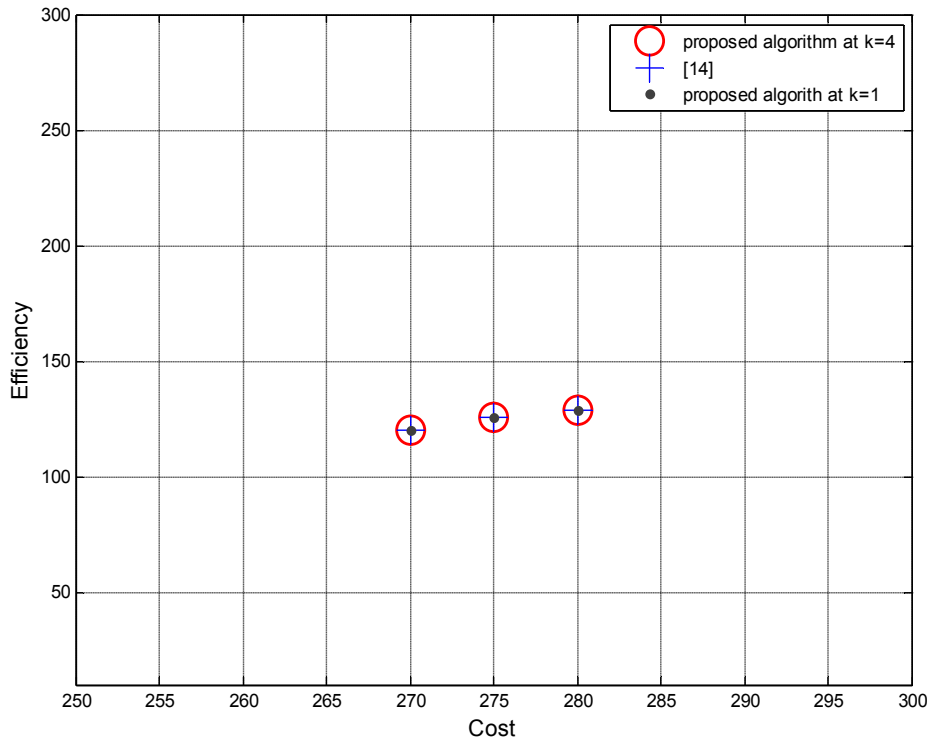


Fig. 11. Efficient solution of test problem 1 obtained by the proposed algorithm at k=4, k=1 and the effective genetic algorithm approach [14]

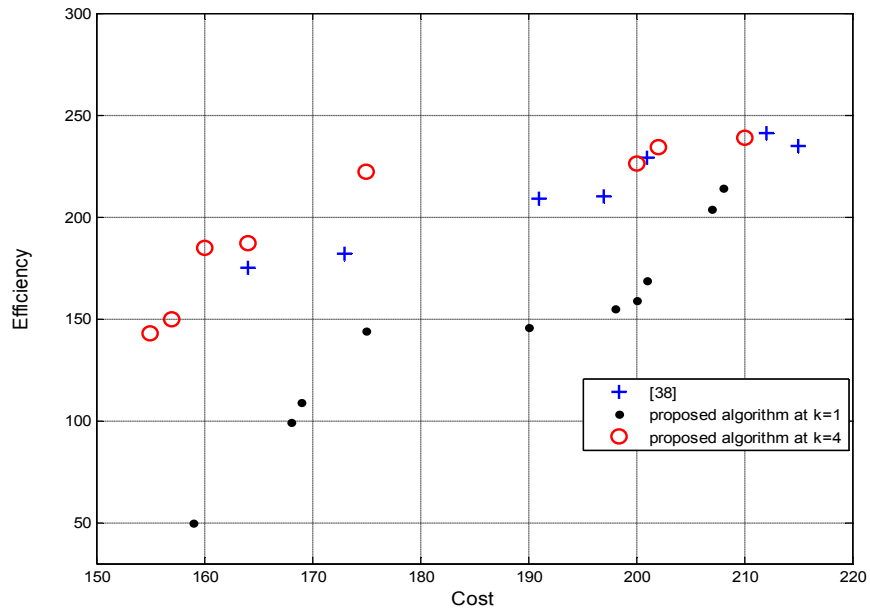


Fig. 12. Simulation results of test problem 2 obtained by the proposed algorithm at k=4, k=1 and multistage decision-based genetic algorithm [38]

Table 6. The expected cost and efficiency of test problem 2

Number of worker	JOB							
	1		2		3		4	
	Cost	Efficiency	Cost	Efficiency	Cost	Efficiency	Cost	Efficiency
0	41	0	45	0	36	0	46	0
1	38	37	54	49	43	45	78	60
2	46	42	36	55	68	49	88	67
3	32	50	55	59	56	57	64	72
4	78	54	87	62	72	64	90	79
5	76	56	82	67	59	77	80	83
6	72	58	90	73	32	88	120	88
7	84	65	132	80	67	92	104	97
8	80	72	97	87	86	100	96	102
9	92	80	21	95	188	105	86	110
10	96	95	134	102	100	110	120	120

Table 7. Efficient solution of the proposed algorithm at k=4

X1	Efficient solution			Overall cost	Overall efficiency
	X2	X3	X4		
2	1	6	1	210	239
3	1	6	0	164	187
1	1	6	1	202	234
3	2	2	3	200	226
3	2	1	0	157	150
3	2	1	3	175	222
2	2	6	0	160	185
0	2	6	0	155	143

Table 8. Efficient solution of the proposed algorithm at k=1

Efficient solution				Overall cost	Overall efficiency
X1	X2	X3	X4		
2	3	1	0	190	146
3	1	1	0	175	144
3	3	0	1	201	169
3	3	0	0	169	109
3	1	1	1	207	204
3	0	0	0	159	50
3	3	1	1	208	214
3	1	0	0	168	99
3	1	0	1	200	159
3	0	1	1	198	155

Table 9. Efficient solution of multistage decision-based genetic algorithm [38]

Efficient solution				Overall cost	Overall efficiency
X1	X2	X3	X4		
3	2	1	4	201	229
0	2	6	2	197	210
3	2	5	0	173	182
3	1	6	0	164	175
1	1	6	2	212	241
1	1	5	3	215	235
0	1	6	3	191	209

7. CONCLUSION

In this paper, GA based on k-means clustering technique is proposed to solve MORAPs. K-means clustering technique divided population of GA into a specific number of subpopulations with dynamic size; where different GA operators can be applied to subpopulations instead of one GA operators applied to the whole population. Furthermore, using GA with K-means clustering preserve and introduce diversity, and enable the algorithm to avoid local minimaby preventing the population of chromosomes from becoming too similar to each other and to benefit from the advantages of both types of algorithms. Two test problems taken from the literature are solved to verify the performance of the proposed algorithm and its capability to solve MORAP. The simulation results prove superiority of the proposed algorithm to those reported in the literature, where it completely better than the other approaches.

ACKNOWLEDGEMENTS

The authors are grateful to the anonymous reviewers for their valuable comments and helpful suggestions which greatly improved the paper's quality.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Ponz-Tienda JL, Yepes V, Pellicer E, Moreno-Flores J. The resource leveling problem with multiple resources using an adaptive genetic algorithm. *Automation in Construction*. 2013; 29:161-172.
2. Chaharsooghi SK, Kermani AHM. An effective ant colony optimization algorithm (ACO) for multi-objective resource allocation problem (MORAP). *Applied Mathematics and Computation*. 2008;200:167-177.
3. Lilien GL, Rangaswamy A, Bruggen GHV, Stark K. DSS effectiveness in marketing resource allocation decisions: Reality vs. Perception. *Information Systems Research*. 2004;15(3):216-235.
4. Wang X, Giannakis GB. Resource Allocation for Wireless Multiuser OFDM Networks. *IEEE*. 2011;57(7):4359-4372.
5. Tan S, Weng W, Fujimura S. Scheduling of Worker Allocation in the Manual Labor Environment with Genetic Algorithm.

- International Multi Conference of Engineers and Computer Scientists (IMECS) I; 2009.
6. Ehrgott M, Klamroth K, Schwehm C. An MCDM approach to portfolio optimization. *European Journal of Operational Research*. 2004;155(3):752–770.
 7. Claxton K, Paulden M, Gravelle H, Brouwer W, Culyer AJ. Discounting and decision making in the economic evaluation of health-care technologies. *Health Economics*. 2011;20(1):2-15.
 8. Ibaraki T, Katoh N. Resource allocation problems: Algorithmic Approaches. MIT Press. Boston(USA); 1988.
 9. Zhao ZJ, Lau HC, Ge SS. Integrated Resource Allocation and Scheduling in a Bidirectional Flowshop with Multimachine and COS Constraints, *Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews*. 2009;39(2):190-200.
 10. Demeulemeester E, Herroelen W. A branch-and-bound procedure for the multiple resource constrained project scheduling problem. *Management Science*. 1992;38(12):1803-1818.
 11. Gavish B, Pirkul H. Algorithms for multi-resource generalized assignment problem. *Management Science*. 1991;37(6):695–713.
 12. Moselhi A, Lorterapong P. Least impact algorithm for resource allocation. *Canadian Journal of Civil Engineering*. 1993;20(2): 180–188.
 13. Deb K. Multi-Objective optimization using evolutionary algorithms. John Wiley & Son. Chichester (UK); 2001.
 14. Osman MS, Abo-Sinna MA, Mousa AA. An effective genetic algorithm approach to multi-objective resource allocation problems. *Applied Mathematics and Computation*. 2005;163:755–68.
 15. Wang KJ, Lin YS. Resource allocation by genetic algorithm with fuzzy inference. *Expert Systems with Applications*. 2007; 33:1025–1035.
 16. Sousa T, Vale Z, Carvalho JP, Pinto T, Morais H. A hybrid simulated annealing approach to handle energy resource management considering an intensive use of electric vehicles. *Energy*. 2014;67:81-96.
 17. Liang YC, Chuang CY. Variable neighborhood search for multi-objective resource allocation problems. *Robotics and Computer-Integrated Manufacturing*. 2013;29:73–78.
 18. Wu J, Chang Z, Yuan L, Hou Y, Gong M. A Memetic algorithm for resource allocation problem based on node-weighted graphs. *IEEE Computational Intelligence magazine*. 2014;9(2):58–69.
 19. Jia Q, Seo Y. An improved particle swarm optimization for the resource-constrained project scheduling problem. *International Journal of Advanced Manufacturing Technology*. 2013;67:2627–2638.
 20. Tounsi M. A Heuristic-based technique for University Resource Allocation Problems. 2006 IEEE GCC conference. 2006;1-6.
 21. Qiao F, Lin P. A Differential evolutionary based algorithm for multiuser OFDMA system adaptive resource allocation. 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08). 2008;1-5.
 22. Yang B, Hu Y, Huang CY. An architecture-based multi-objective optimization approach to testing resource allocation. *IEEE Transactions on Reliability*. 2014;PP(99):1-19.
 23. Fatima S, Bader-El-Den M. Co-evolutionary hyper-heuristic method for auction based scheduling. *IEEE Congress on Evolutionary Computation (CEC2010)*. 2010;1-8.
 24. Zhao SY, Szeto KY. Combining exhaustive search with evolutionary computation via computational resource allocation. *IEEE Congress on Evolutionary Computation (CEC 2007)*. 2007;1878-1881
 25. Murakami K, Gen M, Tasan SO, Oyabu T. A solution of human resource allocation problem in a case of hotel management. 40th International Conference on Computer and Industrial Engineering (CIE). 2010;1-6.
 26. Yan HC, Liu HL, Wang Q. Dynamic resource allocation of TD-LTE system based on improved quantum evolutionary algorithm. 9th International Conference on Computational Intelligence and Security (CIS). 2013;26-30.
 27. Verma M, Srivastava M, Chack N, Diswar AK, Gupta N. A Comparative study of various clustering algorithms in data mining. *International Journal of*

- Engineering Reserch and Applications (IJERA). 2012;2(3):1379-1384.
28. Aggarwal CC, Reddy CK. DATA CLUSTERING Algorithms and Applications . CRC Press Taylor & Francis Group. New York (USA); 2014.
 29. MacQueen JB. Some methods for classification and analysis of multivariate observations. Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press. 1967:281-297.
 30. Arai K, Bu XQ. ISODATA clustering with parameter (threshold for merge and split) estimation based on GA: Genetic Algorithm. Reports of the Faculty of Science and Engineering, Saga University. 2007;36(1):17-23.
 31. Ng RT, Han J. Clarans. A method for clustering objects for spatial data mining. IEEE Transactions Knowledge Data Engineering. 2002;14(5):1003-1016.
 32. Judd D, McKinley PK, Jain AK. Large-scale parallel data clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1998;20(8):871-876.
 33. Edlaa DR, Janaa PK. A prototype-based modified DBSCAN for gene clustering. Procedia Technology. 2012;6:485–492.
 34. Rani Y, Manju, Rohil H. Comparative analysis of BIRCH and CURE hierarchical clustering algorithm using WEKA 3.6.9. The SIJ Transactions on Computer Science Engineering & Its Applications (CSEA). 2014;2(1):25-29.
 35. Abu-Srhan A, Al Daoud E. A hybrid algorithm using a genetic algorithm and cuckoo search algorithm to solve the traveling salesman problem and its application to multiple sequence alignment. International Journal of Advanced Science and Technology. 2013;61:29-38.
 36. Goldberg DE. Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley Publishing Company .Boston (USA); 1989.
 37. Michalewiz Z. Genetic Algorithms+Data Structures¼Evolution Programs, 3rd ed. Springer. Verlag; 1996.
 38. Lin CM, Gen M. Multi-objective resource allocation problem by multistage decision-based hybrid genetic algorithm. Applied Mathematics and Computation. 2007;187: 574-583.
 39. Mohd RN, Geraghty J. Genetic Algorithm Performance with Different Selection Strategies in Solving TSP. Proceedings of the World Congress on Engineering II; 2011.
 40. Sakthi M, Thanamani AS. An effective determination of initial centroids in K-means clustering using kernel PCA. International Journal of Computer Science and Information Technologies. 2011;2(3):955-959.
 41. Jain A. Data clustering: 50 years beyond k-means. Pattern Recognition Letters. 2010; 31(8):651–666.
 42. Miettinen K. nonlinear multi-objective optimization. Kluwer Academic Publishers. Boston (USA); 1999.
 43. Steuer RE. Multiple criteria optimization: theory, computation, and application. John Wiley & Sons. New York (USA); 1986.
 44. Srinivas N, Deb K. Multi-objective optimization using non-dominated sorting in genetic algorithms. Evolutionary Computation. 1999;2(3):221–248.
 45. Murata T, Ishibuchi H, Tanaka H. Multiobjective genetic algorithm and its application to flow shop Scheduling. Computers and Industrial Engineering. 1996;30(4):957–968.
 46. Sivaraj R, Ravichandran T. A review of selection methods in genetic algorithm. International Journal of Engineering Science and Technology. 2011;3(5):3792-3797.
 47. Tzeng GH, Huang JJ. Fuzzy multiple objective decision making. Chapman and Hall / CRC Press; 2013.
 48. Kaya M. The effects of two new crossover operators on genetic algorithm performance. Applied Soft Computing. 2011;11:881–890.
 49. Gen M, Cheng R. Genetic algorithms and engineering optimization. John Wiley & Sons. New York (USA); 2000.
 50. Faraji R, Naji HR. An efficient crossover architecture for hardware parallel implementation of genetic algorithm. Neurocomputing. 2014;128:316–327.
 51. Abdoun O, Abouchabaka J, Tajani C. Analyzing the performance of mutation operators to solve the travelling salesman problem. International Journal of Emerging Sciences. 2012;2(1):61-77.

52. Goh CK, Ong YS, Tan KC. Multi-Objective Memetic Algorithms. Springer, Berlin (Germany); 2009.
53. [Laumanns M, Thiele L, Deb K, Zitzler E. archiving with guaranteed convergence and diversity in multi-objective optimization. In GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers, New York (USA). 2002;439-447.

© 2015 Farag et al.; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:

The peer review history for this paper can be accessed here:
<http://www.sciencedomain.org/review-history.php?iid=1069&id=5&aid=8542>