International Journal of Applied Evolutionary Computation

Volume 9 • Issue 1 • January-March 2018 • ISSN: 1942-3594 • eISSN: 1942-3608
An official publication of the Information Resources Management Association

IGI PUBLISHING
AN IMPRINT OF IGI GLOBAL
WWW.IGI-GLOBAL.COM

Call for Articles

International Journal of Applied Evolutionary Computation

Volume 9 • Issue 1 • January-March 2018 • ISSN: 1942-3594 • eISSN: 1942-3608
An official publication of the Information Resources Management Association

MISSION

The **International Journal of Applied Evolutionary Computation (IJAEC)** presents comprehensive interdisciplinary and refereed research on the most emerging and breakthrough areas of Evolutionary Computation (EC) and applications. The journal expands existing research findings (theoretical innovations and modeling applications) to provide the highest quality original concepts, hybrid applications, innovative methodologies, and development trends studies for all audiences. The primary mission of IJAEC is to publish original/state-of-the-art/highest quality research articles in application and convergence of the areas of evolutionary computation and other similar techniques to address real world complexities. Evolutionary computation theory, practice and applications are the essential topics being covered. The journal succeeds in expanding the fields and depths of the most principal and critical concepts forming a more matured research society. The journal is targeted at both academic researchers and practicing EC professionals.

COVERAGE AND MAJOR TOPICS

The topics of interest in this journal include, but are not limited to:
Bio-inspired systems • Chaos Theory • Cloud theory • Data Mining • Data visualization • Decision Support Systems • Evolutionary algorithms and programming • Evolutionary computing • Financial modeling • Fuzzy Computing • Grid Computing • Hybrid Methods • Immunological computing • Interactive computational models • Manufacturing Systems • Natural Computing • Neural Computing • Optimization • Power and energy • Probabilistic computing • Signal or image processing • Support vector regression forecasting • Time series prediction

ALL INQUIRIES REGARDING IJAEC SHOULD BE DIRECTED TO THE ATTENTION OF:

Wei-Chiang Samuelson Hong, Editor-in-Chief • IJAEC@igi-global.com

ALL MANUSCRIPT SUBMISSIONS TO IJAEC SHOULD BE SENT THROUGH THE ONLINE SUBMISSION SYSTEM:

http://www.igi-global.com/authorseditors/titlesubmission/newproject.aspx

IDEAS FOR SPECIAL THEME ISSUES MAY BE SUBMITTED TO THE EDITOR(S)-IN-CHIEF

PLEASE RECOMMEND THIS PUBLICATION TO YOUR LIBRARIAN

For a convenient easy-to-use library recommendation form, please visit:
http://www.igi-global.com/IJAEC

Table of Contents

International Journal of Applied Evolutionary Computation

Volume 9 • Issue 1 • January-March-2018 • ISSN: 1942-3594 • eISSN: 1942-3608

An official publication of the Information Resources Management Association

Research Articles

The *International Journal of Applied Evolutionary Computation* is indexed or listed in the following: ACM Digital Library; Bacon's Media Directory; Cabell's Directories; DBLP; Google Scholar; JournalTOCs; MediaFinder; The Standard Periodical Directory; Ulrich's Periodicals Directory

International Journal of Applied Evolutionary Computation
Volume 9 • Issue 1 • January-March 2018

Efficient Golden-Ball Algorithm Based Clustering to solve the Multi-Depot VRP With Time Windows

Lahcene Guezouli, University of Batna 2, Batna, Algeria
Mohamed Bensakhria, University of Batna 2, Batna, Algeria
Samir Abdelhamid, University of Batna 2, Batna, Algeria

ABSTRACT

In this article, the authors propose a decision support system which aims to optimize the classical Capacitated Vehicle Routing Problem by considering the existence of multiple available depots and a time window which must not be violated, that they call the Multi-Depot Vehicle Routing Problem with Time Window (MDVRPTW), and with respecting a set of criteria including: schedules requests from clients, the capacity of vehicles. The authors solve this problem by proposing a recently published technique based on soccer concepts, called Golden Ball (GB), with different solution representation from the original one, this technique was designed to solve combinatorial optimization problems, and by embedding a clustering algorithm. Computational results have shown that the approach produces acceptable quality solutions compared to the best previous results in similar problem in terms of generated solutions and processing time. Experimental results prove that the proposed Golden Ball algorithm is efficient and effective to solve the MDVRPTW problem.

KEYWORDS

Clustering, Golden Ball Algorithm, Multi-Depot Vehicle Routing Problem, Routing, Scheduling

1. INTRODUCTION

Within the wide scope of logistics management, transportation plays a central role and is a crucial activity in the delivery of goods and services. The transport problem is one of the mainly essential combinatorial optimization problems that have taken the interest of several researchers. Huge research efforts have been devoted to the study of logistic problems and thousands of papers have been written on many variants of this problem such as Traveling Salesman Problem (TSP) (Toth, 2001), Vehicle Routing Problem (VRP) and supply chain management (SCM) (Pisinger, 2007).

The Vehicle Routing Problem (VRP) has been one of the central topics in optimization since Dantzig proposed the problem in 1959(Dantzig, 1959). A simple general model of VRP can be described as follows: a set of service vehicles need to visit all customers in a geographical region with the minimum cost. The VRP is also named Single-depot VRP (SDVRP). In cases with more than one depot, VRPs are known as multi-depot VRPs (MDVRP) (Filipec, 2000), (Mirabi, 2014). Single-depot VRPs are not suitable for practical situations though they have attracted researchers in a wide sense.

The Multi-Depot Vehicle Routing Problem (MDVRP) is a generalization of SDVRP in which the vehicles start from multiple depots and return to their original depots at the end of their assigned tours. As there are a large number of depots, it is a difficult task for decision makers to determine

DOI: 10.4018/IJAEC.2018010101

which customers are served by which depots without exceeding the capacity constraints, especially when we use heterogeneous vehicles. Hence grouping is performed to cluster customers based on distance between the customers and the depots, prior to the routing and scheduling phases.

The MDVRP in known to be NP-hard problem (Lenstra, 1981), and also one of problems which are applicable to the real world. For these reasons, in the literature, many different techniques designed to be applied to these problems can be found.

Nowadays, the development of new meta-heuristic algorithms for the MDVRP is still active such as neighborhood simulated annealing (Kirkpatrick, 1983), genetic algorithm (GA) (Goldberg, 1989), ant colony optimization (Dorigo, 2005), and particle swarm optimization (Kennedy, 2011).

The objective of this paper is to present a new meta-heuristic based on soccer concepts for solving routing problems. This new technique is a multiple population based algorithm, and we called Golden Ball (GB). It divides the different solutions of the problem in different teams, which improve independently and cooperatively and face each other in a competition. This competition will be crucial to decide the transfer of solutions between teams and to decide the model of training of each team.

The Golden ball algorithm is a relatively new and efficient meta-heuristic algorithm which can be used to solve the MDVRP. However, in the literature we have found that they only prove its success with only two simple variants of the vehicle routing problem: The CVRP, which was introduced in 2013 by (Osaba, 2014), and improved by (Ruttanateerawichien, 2014). And (Osaba, 2013) who applied the GB algorithm to solve the Vehicle Routing Problem with Backhauls (VRPB). This is the reason that motivates the work presented in this paper. Thus, the main objective is to prove that the Golden Ball algorithm is a promising meta-heuristic to solve the MDVRPTW.

The remainder of this paper is structured as follows: the next section presents the existing works related to MDVRPTW solutions by various heuristic methods, Section 3 describes the MDVRPTW with an example and the mathematical model. The step by step procedure of implementing MDVRPTW using Genetic Algorithm and clustering concept is explained in Sections 4-12. The computational results for the benchmark instances are analyzed in Section 13 and 14. Section 15 draws the conclusion and future scope of the application.

2. LITERATURE SURVEY

This section briefs the existing work related to MDVRPTW solutions by various heuristic methods. The available literature on MDVRPs is quite limited compared with the extensive literature on simple VRPs and their variants.

The Multi Depot Vehicle Routing Problem with Time Windows (MDVRPTW) (Surekha, 2011) is considered for a practical description of transportation planning. The MDVRPTW describes the problem to deliver uniform goods to a set of customers from a set of depots with heterogeneous capacities vehicles. The delivery has to be done within a customer-specified time window and the vehicles need to return to the same depot where they have started. Each customer has to be delivered once.

The MDVRPTW is an extension of the fundamental route planning problem the so-called Vehicle Routing Problem (VRP) from (Dantzig, 1959). As the VRP is NP-hard the MDVRPTW with more restrictions also belongs to the set of NP-hard problems (Polacek, 2004). Therefore, it is recommended to use a heuristic approach solving the MDVRPTW. The definition of time windows in this case refers to the definition of so-called soft time windows from (Chiang, 2004). In comparison to the definition of hard time windows a route is still feasible although the time window restriction is violated. The resulting delay at one or more customers is measured by a penalty.

Various meta-heuristics have been proposed for MDVRPTW problems. These approaches seek approximate solutions in polynomial time instead of exact solutions which would be at intolerably high cost. Early heuristics were based on simple construction techniques incorporating improvement procedures, like (Tillman, 1969) who used the Clarke and Wright savings method. (Renaud, 1996) who introduced a tabu search with diversification and intensification, and (Thangiah, 1998) used a genetic clustering approach.

This paper is focused on one recently proposed meta-heuristic called Golden Ball (GB).This technique is a multiple-population based meta-heuristic, and it is based on soccer concepts. A preliminary version of the GB and some basic results were firstly introduced in 2013 by (Osaba, 2013). Furthermore, the final version of the GB and its practical use for solving complex problems have been presented in (Osaba, 2014) by the same authors.

The GB algorithm began from the simulation of different concepts related to soccer for the search process (Osaba, 2014). It represents a solution of a problem as a soccer player, and a population is represented by the soccer teams. In each iteration, all players have the chance to improve their solution quality by training procedures. The players can switch from their team to another by transfer procedure.

Since the GB algorithm was firstly proposed in 2014, it has been applied to the travelling salesman problem (Osaba, 2014), the capacitated vehicle routing problem (Osaba, 2014), (Ruttanateerawichien, 2016), asymmetric travelling salesman problem (Osaba, 2013), vehicle routing problem with backhauls (Osaba, 2013), n-Queen problem (Osaba, 2013), one-dimensional bin-packing problem (Osaba, 2013).

In (Osaba, 2014) proposed an original version of GB algorithm to CVRP. In their work, all players are generated randomly. After that, these players are randomly divided among the different teams. In the training process, they used 2-opt and vertex insertion as intra-route improvement algorithm, and swapping routes and vertex insertion routes as inter-route improvement algorithm. Furthermore, they used golden help function in order to improve more solution quality of players.

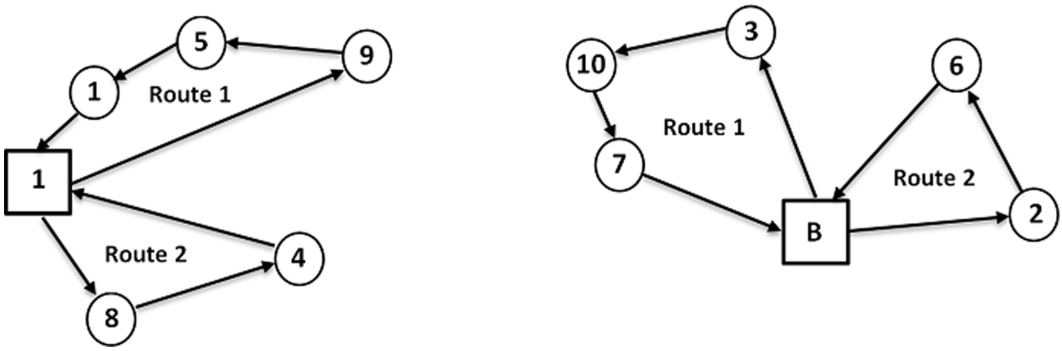
(Ruttanateerawichien, 2014) introduced an improved version of GB algorithm to CVRP. The solution representation in their work is similar to the original GB algorithm that one CVRP solution is represented by one soccer player. In their work, all players in each team are generated randomly, except one player who is generated by using and improved Clarke and Wright savings algorithm (Jeon, 2007). In the training process, they used intra-move and inter-move operators composed of shift and swap moved to improve solution quality of the players.

In line with this, it was noticed that the GB algorithm has only been used to solve two variants of the vehicle routing problem (CVRP, VRPB). In this way, the main motivation of this work is to prove that the GB algorithm can also compete with other famous and widely used variants like the MDVRPTW. To perform this task, in this study, we develop a solution for the MDVRPTW based on GB and clustering concept with best cost route crossover. The obtained results are compared with the ones obtained by two different and frequently used meta-heuristics: the Evolutionary Simulated Annealing (ESA) (Kirkpatrick, 1983) and the Tabu Search (TS) (Renaud, 1996). In this first study, the well-known TSP is used for the comparison.

3. MD-VRPTW MODELING

In the modeling phase, we assume that the customer size, location, the number and location of all potential depots are known. The vehicle type and size are also given. Each depot is large enough to store all the products ordered by the customers. Each vehicle starts and finishes at the same depot. The location and demand of each customer is also known in advance. Each customer is visited by a vehicle exactly once. The customers are served in their respective time windows and each vehicle must follow a chronological distribution during its trip. Figure 1 shows an example of the MDVRPTW with 2 depots and 10 customers. Since there are additional depots for storing the products, the decision makers have to determine depots through which the customers are served (Ho, 2008). The decision-making stages are classified into clustering, routing, scheduling and optimization as shown in Figure 2. In clustering, customers are grouped based on distance between customers and depots. In the example, customers 1,5,9,4,8 are assigned to depot A, while customers 7,10,3,6,2 are assigned to depot B.

Figure 1. Example of an MDVRP with 2 depots and 10 customers



In this study, we use the MDVRP model adopted by (Renaud, 1996). Let G = (J, I) be a direct graph where $J = \{j_1, \ldots, j_n\}$ representing the set of customers, $I= \{i_{n+1}, \ldots, i_{n+p}\}$ the set of depots, and $K = \{1, \ldots, m\}$: the set of available vehicles.

The set of arcs A denotes all possible connections between the nodes (including nodes denoting depots). We define a cost matrix $X = (x_{ij}^k)$ on A corresponding to travel times (x_{ij}^k represent the cost of the arc (i,j) taken by the vehicle k).

Our MDVRPTW can be stated as follows:

- N: Number of vehicles.
- C_{ij}: Euclidean distance between the customer i and the node j.
- V_i: Maximum through-put at depot i.
- t_i: The service time of the customer j_i.
- t_{ij}: The transport time from the customer j_i to the node j_j.
- d_i: Demand of customer i.
- Q_k: The capacity of the vehicle(route) k.
- $[a_i, b_i]$: The time window associated to the customer j_i with a_i representing the arrival time at the earliest to the customer j_i and b_i representing the end time of service at the latest for the customer j_i.
- M: Great value.

The decision variables used in this formulation are:

- $y_i^k = \begin{cases} 1 & \text{if the node } j_i \text{ is visited by the vehicle k} \\ 0 & \text{otherwise} \end{cases}$
- $x_{ij}^k = \begin{cases} 1 & \text{if i immediately preceeds j on route k} \\ 0 & \text{otherwise} \end{cases}$
- $z_{ij} = \begin{cases} 1 & \text{if customer j is allotted to depot i} \\ 0 & \text{otherwise} \end{cases}$
- U_{lk}: auxiliary variable for sub-tour elimination constraints in route k.

Figure 2. Decision making in MDVRPTW

- S_i^k: The temporal decision variable representing the arrival time of the vehicle k to the customer j_i.

With the aim to simplify the solution of our problem, we set a main goal based on the cost of the tour and a set of constraints to be respected. We recapitulate the mathematical formulation of our problem as follows:

- The objective function is to minimize the total cost of the tour. (C1),

$$Min\left(\sum_{i\in\left(i\cup j\right)}\sum_{j\in\left(i\cup j\right)}\sum_{k\in K}c_{ij}x_{ij}^{k}\right) \tag{C1}$$

Subject to:

- Each customer has to be assigned a single route according to Equation (C2),

$$\sum_{keK}\sum_{i\in(I\cup J}x_{ij}^{k}=1,\ (\mathrm{j}\in\mathrm{J}) \tag{C2}$$

- The constraint (C3) expresses the fact that the vehicle capacity is limited,

$$\sum_{j\in J^{d_j}}\sum_{j\in\left(I\cup J\right)}X_{ij}^{k}\ .\leq \mathrm{Q_k},\ (\mathrm{k}\in\mathrm{K}) \tag{C3}$$

- Equation (C4) gives the new sub-tour elimination constraint set as,

$$\mathrm{U_{lk}} - \mathrm{U_{jk}} + \mathrm{N}\,X_{ij}^{k}\ \leq \mathrm{N}\text{-}1,\ (\mathrm{l,j}\in\mathrm{J}),\ (\mathrm{k}\in\mathrm{K}) \tag{C4}$$

- Each route can be served at most once according to Equation (C5),

$$\sum_{ieP}X_{ij}^{k}\ .\leq 1,\ (\mathrm{k}\in\mathrm{K}) \tag{C5}$$

- The capacity constraints for the depots are given in Equation (C6) as,

$$\sum_{j\in j}d_{i}z_{ij}\leq V_{i}\ ,\ (\mathrm{i}\in\mathrm{I}) \tag{C6}$$

- Constraints in Equation (C7) specify that a customer can be assigned to a depot only if there is a route from that depot going through that customer,

$$\text{-}\mathrm{Z_{ij}} + \sum_{u\in\left(I\cup J\right)}\left(x_{iu}^{k}X_{uj}^{k}\leq 1\right).,(\mathrm{i}\in I),\ (\mathrm{j}\in J),\ (\mathrm{k}\in K) \tag{C7}$$

- The constraint (C8) ensures that nodes must be served in their respective time windows.

$$a_i y_i^k \le S_i^k \le b_i y_i^k,\ (i \in I),\ (k \in K) \tag{C8}$$

- The constraint (C9) used to establish a chronological distribution to vehicles during their routes.

$$S_i^k + t_i + t_{ij} - M(1\text{-}x_{ij}^k) \le b_j (i \in J),\ (k \in K) \tag{C9}$$

- The binary requirements on the decision variables are given by Equations (C10), (C11) and (C12)

$$x_{ij}^k \in \{0,1\},\ (i,j \in \left(I \cup J\right).;\ i \neq j),\ (k \in K) \tag{C10}$$

$$y_i^k \in \{0,1\},\ (i \in \left(I \cup J\right).\ (k \in K) \tag{C11}$$

$$Z_{ij} \in \{0,1\},\ (i,j \in \left(I \cup J\right) \tag{C12}$$

- The positive values of the auxiliary variable is defined in Equation (C13) as

$$Q_{k,}\ C_{ij},\ U_{lk},\ M \ge 0 \tag{C13}$$

4. PROPOSED GOLDEN-BALL META-HEURISTIC

As we said in the introduction of this work, our new meta-heuristic called Golden Ball is a multiple population based algorithm which takes some concepts related to soccer for the search process. Most of the existing applications of Golden Ball algorithm to VRP use the solution representation by generating the player as VRP solution. In this section, we propose a new scheme based on the application of GB algorithm with different solution representation from the original one and by embedding a clustering algorithm in the initialization phase to effectively solve the MDVRPTW.

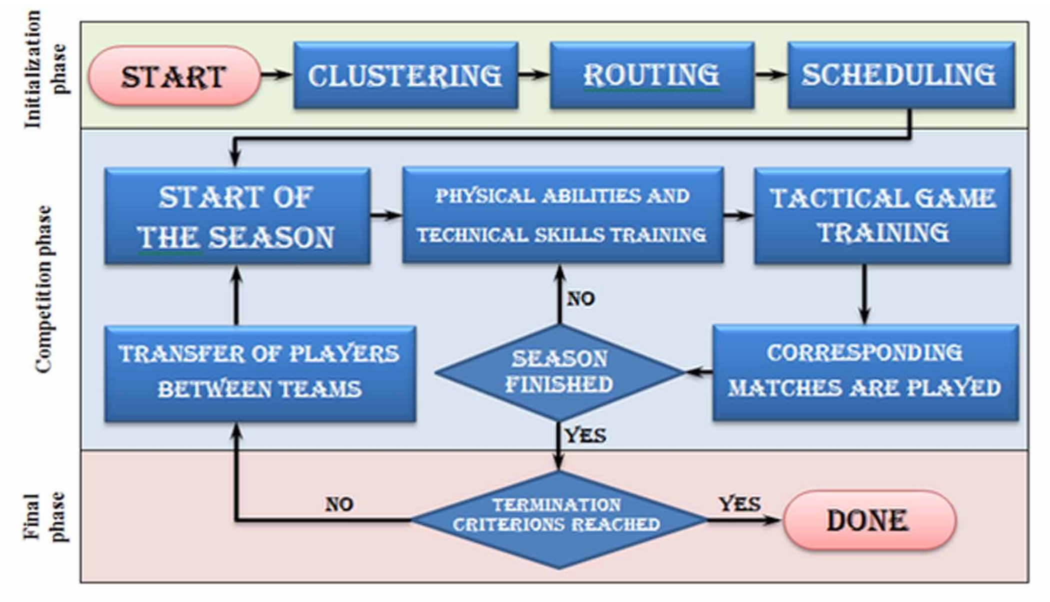
In this new solution representation, the population of the proposed GB algorithm is represented by the soccer teams that each team represents one MDVRPTW solution, each team is composed of a set of players that each player represents a cluster (depot + associated nodes).

The technique starts with the initialization phase, where the whole population of solutions (called teams) is created. Then, these teams are divided among the different subpopulations.

Once this initial phase has been completed, the competition phase begins. This second phase is divided in seasons. Each season is composed of weeks, in which the teams train independently and face each other creating a league competition. At the end of every season, a transfer procedure happens, in which the players can switch teams.

The competition phase is repeated until the termination criterion is met. The entire procedure of the technique can be seen in Figure 1. Now, the different steps that form the proposed technique are explained in detail.

Figure 3. Flowchart of the proposed GB algorithm



5. INITIALIZATION PHASE

After the creation of the initial population P which is considered as a first step of the execution process. We proceed to the optimization phase where all subpopulations P_i are divided fairly in the *TN* different teams *ti*. Once the players (clusters) are divided between the different teams (solutions), they are represented by the variable *pij*, which means the player number *j* of the team *i*. The total set of teams *T* forms the league. All these concepts may be represented mathematically as follows:

P: {p1, p2, p3, p4, …, pPT*TN }
T: {t1, t2, t_3, t_4, …, tT_N }
Team t1: {p11, p12, p13, p14, …, p1PT }
Team t2: {$p2_1$, $p2_2$, $p2_3$, $p2_4$, …, p2PT }
⋮
Team tTN: {pTN1, pTN2, pTN3, pTN4, …, pTN $_{PT}$ }
PT: Number of players per team,
T*N* = Total number of teams of the system.

The feasible solution for the optimization process is generated in three basic steps: Clustering, Routing and Scheduling.

6. CLUSTERING (GROUPING)

Initially each customer is assigned to the nearest depot in terms of Euclidean distance, for *n* depots in the MDVRPTW, the solution consists of *n* clusters and the customers are assigned to each of this n clusters. In Figure 1, there are two depots A and B, each customer C_i has to be assigned to a single depot exactly. This process of grouping is done based on the distance computation according to the following rule:

- If $D(C_i,A) < D(C_i,B)$, then customer C_i is assigned to depot A
- If $D(C_i,A) > D(C_i,B)$, then customer C_i is assigned to depot B
- If $D(C_i,A) = D(C_i,B)$, then customer C_i is assigned to a depot chosen arbitrarily between A and B
  - $D(C_i,P)$: represents the distance between customer C_i and depot k. such as:

$$D(C_i,P)= \sqrt{\left(X_{Ci}-X_K\right)^2+\left(Y_{Ci}-Y_K\right)^2}$$

7. ROUTING

A MDVRPTW solution must specify the number of routes (that is, vehicles), and the delivery order within each route. The customers in the same cluster are assigned to several routes using Clarke and Wright Saving *(Clarke, 1964)* method. The routing process is based on the distance travelled by the vehicles for serving the customers. A saving matrix $S(C_i, C_j)$ is constructed for every two customers i and j in the same cluster. Such as,

$$S(C_i, C_j) = D(P,C_i)+ D(P,C_j)- D(C_i,C_j)$$

8. SCHEDULING

In this stage, a vehicle must depart from one depot and starting from the first customer, the delivery sequence is chosen such that the next customer is as close as to the previous customer. The routing procedure takes into consideration that the vehicle capacity and route length constraints are not violated before adding a customer to the current route. This process is continued until each customer has been assigned to exactly one route. At the end of the scheduling phase, a feasible solution of the MDVRPTW example problem (Figure 4) is constructed as shown in Figure 5.

Moreover, every player p_{ij} has its own quality, which is represented by the variable p_{ij}. This variable is represented by a real number, which is determined by a cost function (p_{ij}).

In this paper, our main objective is to minimize the function cost (p_{ij}) corresponding to the transportation cost (the traveled distance), for example, the lower the q_{ij} is, the better the player is.

Furthermore, each team has a strength value associated with *TQi*. This value is crucial for the matches between teams. It is logical to think that the better the players are, the stronger a team is. Thereby, if one team is strong, it can win more matches and it can be better positioned in the classification of the league. In this way, the strength value of a team *ti* is equal to the average of the q_{ij} of the players of that team. *TQi* can be expressed by the following formula:

$$TQi = \frac{\sum_{J=1}^{PT} q_{ij}}{PT}$$

Once the initialization phase is completed, the competition phase begins. This phase is repeated iteratively until the ending criterion is met.

8.1. Competition Phase

In this phase (also called central step of the meta-heuristic), the teams train independently or cooperatively, and they improve their power (*TQi*) little by little. Meanwhile, the teams face each

Figure 4. Example of soccer team

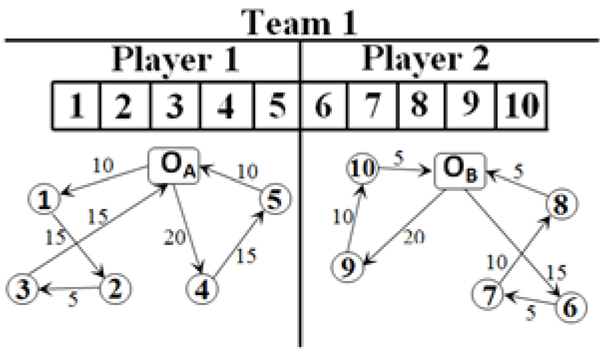
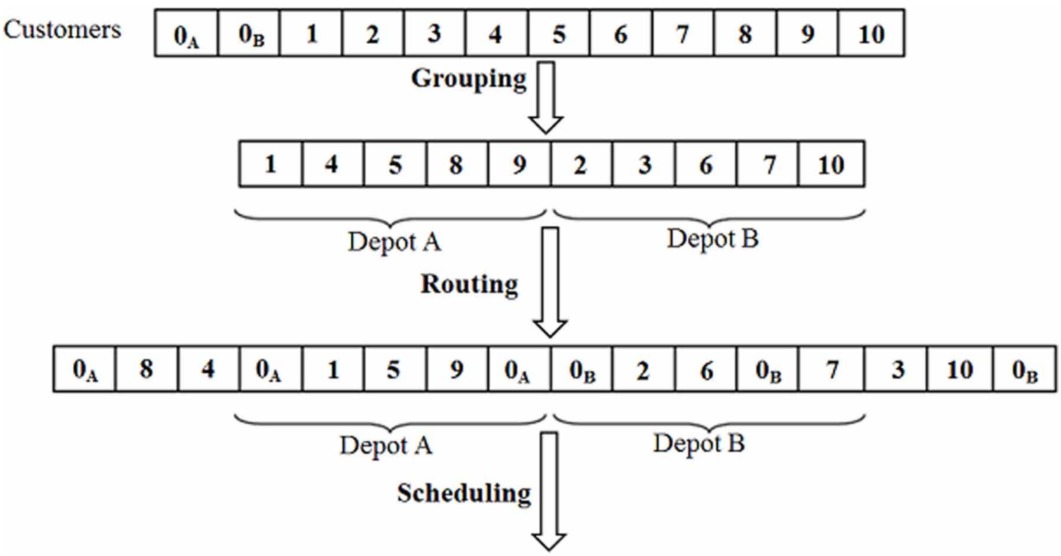


Figure 5. Solution representation



other creating a league competition that helps to decide the transfer of players from different teams. The competition stage is divided into seasons (*Si*). In each season, every team faces each other twice. Finally, a S_i has as many training sessions as matches in the season.

9. PHYSICAL ABILITIES AND TECHNICAL SKILLS

As in real life, the physical abilities and technical skills training are the processes that make players improve their quality. These processes will target to provide every player with the opportunity to develop themselves into the best player they can possibly be. The example components of physical abilities are: jogging, running and sprinting, and technical skills are: dribbling, passing, heading and shooting (Ruttanateerawichien, 2016).

In the proposed GB algorithm, these components are represented by intra-route moves() composed of shift and swap moves that an example is shown in Figure 3. The shift move removes customer 3 from the first cluster (player p1) and insert them in the second place of the same cluster. The swap move selects customers 6 and 7 from the second cluster (player p2) and exchange them. This procedure obtains new cluster which is accepted only if the quality of new player (cluster) is better than the quality of existing player (cluster) (see Figure 6).

10. TACTICAL GAME TRAINING

On the other hand, all players have to learn, on a more individual base, the roles and tasks in order to perform their skills under guidance and then applying their skills in a suitable situation. The example components of tactical game are: defending and attacking exercises with 1-1, 2-2, and 3-3 players involving attackers and defenders (Ruttanateerawichien, 2016).

In the proposed GB algorithm, these components are represented by inter-route moves (Groër, 2010) composed of shift and swap moves that an example is shown in Figure 7. In Figure 7(a)-7(b), the shift moves remove customer 9 from the first cluster (player p1) and insert them in the first place of the second cluster (player p2). In Figure 7(c)-7(d), the swap moves select customer 9 from the first cluster (player p1) and customer 5 from the second cluster (player p2), and exchange them. This procedure obtains new solution which is accepted only if quality of new solution is better than the existing one.

11. MATCHES BETWEEN TEAMS

In GB, as in the real world, all teams have to play matches against each other in order to create a league competition. In each match team t_i faces t_j and the team with the highest quality (the lowest TQ_i) wins the match. Furthermore, the team that wins the match obtains 3 points and the loser obtains 0 points. If both teams obtain the same number of goals, each one receives one point.

These points are used to perform a team classification, sorting the teams by the points obtained in a whole season.

12. PLAYER TRANSFERS BETWEEN TEAMS

The transfer is a process in which the teams exchange players between them; the players are moved between the teams in every season. Teams which are in the top half of the league have opportunities to be reinforced with the best players. While the teams in the lower half will have to settle with the acquisition of the less good players. These interchanges of players help the search process of the meta-heuristic. They allow the different treatment of the solutions during the execution, avoiding falling easily into local optima and increasing the searching capability of the technique.

Figure 6. Example of physical abilities and technical skills training

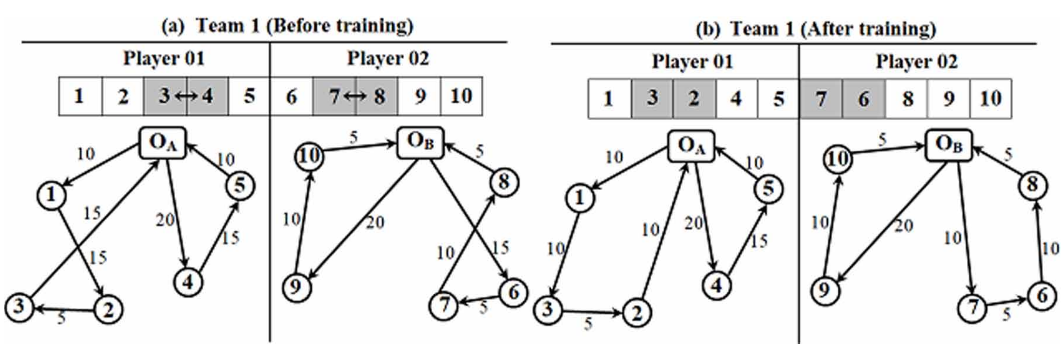
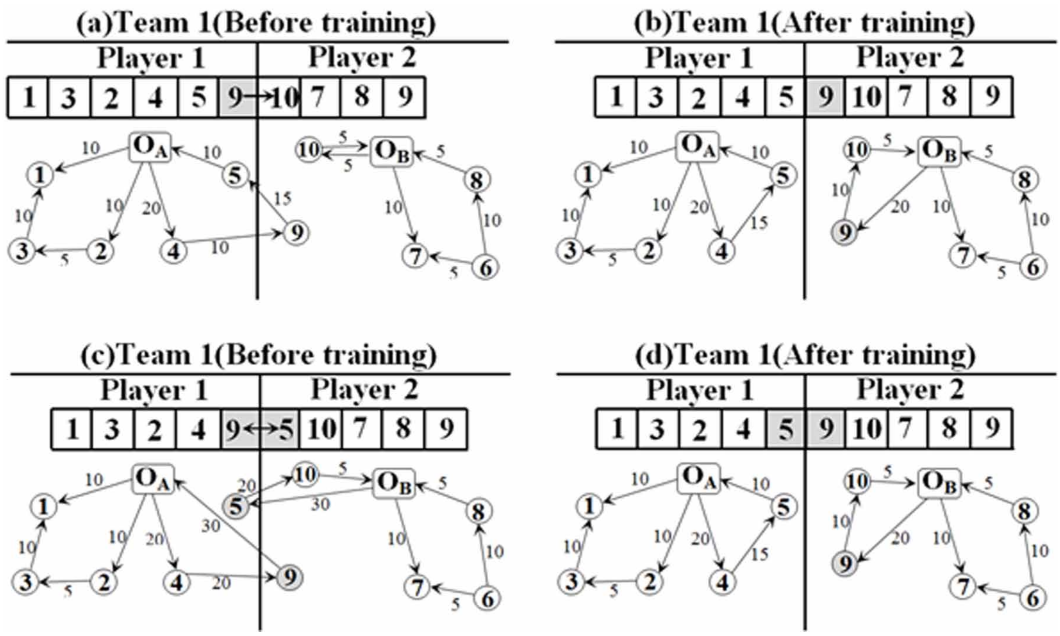


Figure 7. Example of tactical game training



In the proposed GB algorithm, team t_i and team t_j are chosen to move their players that an example is shown in Figure 8. The player p_2 from team t_1 is replaced by the player p_1 from team t_2. This procedure obtains new solution which is accepted only if quality of new solution (new team $t^{'}$) is better than the existing solutions (team t_1 and team t_2).

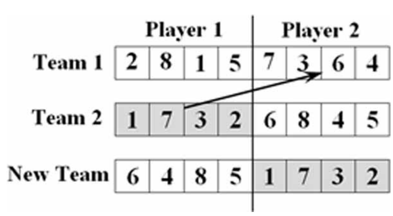
12.1. Termination Criterion

The termination criterion is a critical factor in the development of a meta-heuristic. A bad choice can lead to a considerable waste of time and does not allow to the search to examine a wide area of the solution space. In this way, the termination criterion of the GB is composed of three clauses:

$$\sum_{i=1}^{TN} TQ'_i \geq \sum_{i=1}^{TN} TQ_i \tag{1}$$

$$BestSol' \geq BestSol \tag{2}$$

Figure 8. Example of player transferring



$$MaxIter \geq MI \tag{3}$$

In the proposed GB algorithm, the execution finishes when (1) the sum of the strengths *TQi* of all the teams has not been improved compared to the previous season, (2) there is no improvement in the best solution found (Bestsol') in relation to the previous season, and (3) the maximum number of iterations is reached. When these three conditions are fulfilled, the t*i* with the best TQ_i of the system is returned as the final solution.

13. COMPUTATIONAL RESULTS

All experiments were performed on Microsoft Windows 7 operating systems, i5 core CPU Processor, 4G memory, 500G Hard disk, using Matlab R2010b software. In order to evaluate the performance of the proposed algorithm, we report our computational results and compare them with those from the existing literature.

Each problem is presented as a file which includes: the identifier of each node, these coordinates, their time windows, their service time, quantities to be transported, transport capacity, precedence and succession constraints.

These parameters are defined in three classes as follows:

- The class that defines the parameters of each customer:
  - *Id*: customer identifier,
  - (*x, y*): customer coordinate,
  - *q*: quantities to be transported to the customer,
  - (*e, l*): time windows of the customer,
  - *s*: customer service time,
  - *pred(i)*: predecessor of the customer i,
  - *succ(i)*: successor of the customer i.
- The class that defines the depot:
  - *(x, y):* depot coordinate,
  - *(e, l):* time windows of the depot.
- The class that defines the vehicles:
  - *Q(t):* the capacity of the vehicle *t,*
  - *V(t):* the speed of the vehicle t,
  - *A(t):* Number of available vehicles .

The Golden Ball parameters were set as follows:

- The number of teams = 04;
- The number of players/team depends on the number of available vehicles A(t) for each problem;
- The number of stronger team = 01;
- The number of iterations/stronger team = 1000;
- The number of iterations/physical abilities and technical skills training = 10;
- The number of iteration/tactical game training = 10;
- The number of seasons = 10;
- The number of iterations of the proposed GB algorithm = 1000.

14. TEST PROBLEMS

The performance of our algorithm was tested on the MDVRPTW Euclidean benchmark data sets which can be downloading from the VRP Web at the website (NEO-WEB, 2017). Namely p01, p02, p03, p04, p05, p06, p07, p08, p09, p10, p11, p12, p15, p18, and p21 (instances p13, p14, p16, p17, p19, and p20 were not included in the experiments as they incorporate the constraint on the maximum length of a single route, which the algorithm does not support). It should be noted that the depot and customer vertices are defined by points in the Cartesian plane, where the cost for each edge (i, j) is the Euclidean distance between customers i and j.

In order to verify the variation between the best know solutions achieved by various algorithms during the history of benchmark instances (BKS), and our best solutions (OBS) in terms of optimal distance, we presented the Cordeau's (NEO-WEB, 2017) MDVRPTW instances by a histogram as described in Figure 9. We conducted 100 tests on each instance and registered the best solution found.

Figure 9 shows clearly that there is only a slight difference between the obtained results from our approach and the best know solutions except for the instance p18. However, our approach reaches the best-known solutions three times (P1, P11 and P12), and it should be noted that in the instance P11 our GB algorithm has a reduction of (0.11%) in the total travelled distance.

Now, we discuss the performance of the proposed Golden Ball Algorithm using Clustering approach as compared to the best know results achieved by various algorithms in the literature, and summarize the results in (Table 1). Route costs are measured by average Euclidian distance. The columns labeled GA1 (Thangiah, 2001), GA2 (Surekha, 2011), GC (Thangiah, 2001), gives the best published solutions based on genetic algorithm principles, whereas the GC (Thangiah, 2001) algorithm is also based clustering. SM (Stodola, 2015) stands for Stodola and Mazal's algorithm, and finally FIND (Renaud, 1996) (Fast improvement, INtensification, and Diversification) is a tabu search based algorithm. Best solutions values in (Table 1) are indicated by bold numbers.

We can see that our algorithm managed to find better solutions in all cases (except for the instance p05) when compared with the genetic principle based algorithms (GA1, GA2). The results are also better in 7 cases in comparison with the GC algorithm, in 9 cases (and in 1 case the same) in

Figure 9. Computational Results for the Benchmark Instances Using GB algorithm

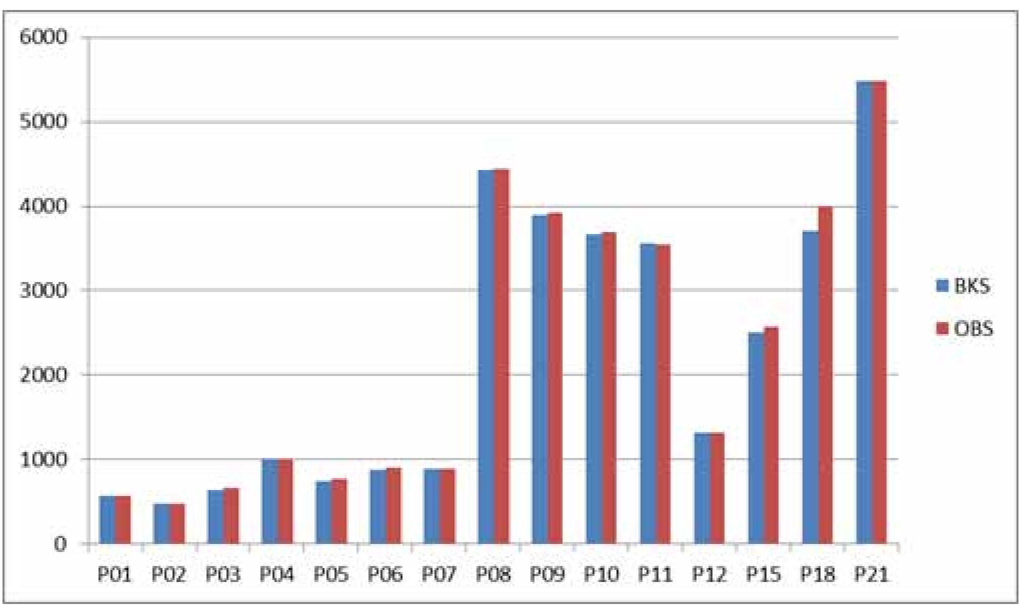


Table 1. Best solutions values obtained by various algorithms

Inst.	N° customer	N° Depots	OBS	GA1	GA2	GC	SM	FIND
P01	50	4	**576.9**	591.7	598. 5	591.7	**576.9**	**576.9**
P02	50	4	479.2	483.1	478.7	**463.2**	485.9	473.5
P03	75	5	659.2	694.5	699.2	694.5	644.5	**641.2**
P04	100	2	**1002.8**	1062.4	1011.4	1062.4	1018.5	1003.9
P05	100	2	773.6	754.8	756.4	755.8	755.7	**750.0**
P06	100	3	910.7	976.0	882.5	976.0	885.8	**876.5**
P07	100	4	**887.9**	976.5	-	-	895.5	892.6
P08	249	2	**4439.2**	4812.5	-	-	4445.5	4485.1
P09	249	3	**3915.6**	4284.6	-	-	3990.2	3937.8
P10	249	4	3690.7	4291.5	-	-	3751.5	**3669.4**
P11	249	5	**3550.2**	4092.7	-	-	3657.2	3649.0
P12	80	2	**1319.0**	1421.9	-	-	**1319.0**	**1319.0**
P15	160	4	2576.6	3059.2	-	3059.2	**2510.1**	2551.5
P18	240	6	4001.5	5462.9	-	5462.9	3741.8	**3181.0**
P21	360	9	**5485.1**	6872.1	-	5544.8	5631.1	5656.5
Number of best solutions			**8**	0	0	1	3	7

comparison with the algorithm SM, and in 6 cases (and in 2 cases the same) in comparison with the algorithm FIND. The bolded values indicate the problem instances where one algorithm out performs the other algorithms by reducing the total cost.

Finally, after having compared the performance of our Golden Ball Algorithm on MDVRPTW instances with (GA1, GA2, GC, SM, FIND) in terms of the optimal distance, we notice that our proposed GB Algorithm is the best, and reaches the best-known solutions three times (P1, P11 and P12) with a maximum cost reduction of (–0.11%) in the instance P11.

15. CONCLUSION

This work introduces a new Golden Ball algorithm based clustering approach for the multi-depot vehicle routing problem with time window. In this approach, we aim to integrate the clustering algorithm into one of the Golden Ball optimization steps, which uses different solution representation from the original one. For that purpose, a mathematical formulation for this general MDVRPTW type was first developed, implemented in MATLAB then compared with different extensions proposed in the literature,

The problem instances were initially grouped to assign the customers to their corresponding depots based on Euclidean distance. The customers of the same depot are assigned to several routes in the routing phase by Clarke and Wright saving method and each route is sequenced in the scheduling phase. The scheduled routes are optimized using Golden Ball Algorithm. The simulation results of the proposed heuristic algorithms were compared to the best knows results achieved by various algorithms in the literature, in terms of optimal distance. It was observed from the conducted experiments, that the performance of Golden Ball algorithm was exceptional because of its simplicity and yet gives efficient solution quality compared to the existing algorithms in the literature.

A possible future work is to impose more realistic constraints on the problem structure and better tune the Golden Ball parameters.

15. REFERENCES

Chiang, W. C., & Russell, R. A. (2004). A metaheuristic for the vehicle-routing problem with soft time windows. *The Journal of the Operational Research Society*, *55*(12), 1298–1310. doi:10.1057/palgrave.jors.2601791

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, *12*(4), 568–581. doi:10.1287/opre.12.4.568

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, *6*(1), 80–91. doi:10.1287/mnsc.6.1.80

Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, *344*(2-3), 243–278. doi:10.1016/j.tcs.2005.05.020

Filipec, M., Škrlec, D., & Krajcar, S. (2000). Genetic algorithm approach for multiple depot capacitated vehicle routing problem solving with heuristic improvements. *International Journal of Modelling and Simulation*, *20*(4), 320–328. doi:10.1080/02286203.2000.11442173

Goldberg, D. E. (1990). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading: Addison-Wesley.

Groër, C., Golden, B., & Wasil, E. (2010). A library of local search heuristics for the vehicle routing problem. *Mathematical Programming Computation*, *2*(2), 79–101. doi:10.1007/s12532-010-0013-5

Ho, W., Ho, G. T., Ji, P., & Lau, H. C. (2008). A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, *21*(4), 548–557. doi:10.1016/j.engappai.2007.06.001

Jeon, G., Leep, H. R., & Shim, J. Y. (2007). A vehicle routing problem solved by using a hybrid genetic algorithm. *Computers & Industrial Engineering*, *53*(4), 680–692. doi:10.1016/j.cie.2007.06.031

Kennedy, J. (2011). Particle swarm optimization. In Encyclopedia of machine learning (pp. 760-766). Springer US.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. science, 220(4598), 671-680.

Lenstra, J. K., & Kan, A. H. G. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, *11*(2), 221–227. doi:10.1002/net.3230110211

Mirabi, M. (2014). A hybrid electromagnetism algorithm for multi-depot periodic vehicle routing problem. *International Journal of Advanced Manufacturing Technology*, *71*(1-4), 509–518. doi:10.1007/s00170-013-5501-0

Osaba, E., Diaz, F., & Onieva, E. (2013, July). A novel meta-heuristic based on soccer concepts to solve routing problems. In *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation* (pp. 1743-1744). ACM. doi:10.1145/2464576.2480776

Osaba, E., Diaz, F., & Onieva, E. (2014). Golden ball: A novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts. *Applied Intelligence*, *41*(1), 145–166. doi:10.1007/s10489-013-0512-y

Pisinger, D., & Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, *34*(8), 2403–2435. doi:10.1016/j.cor.2005.09.012

Polacek, M., Hartl, R. F., Doerner, K., & Reimann, M. (2004). A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of Heuristics*, *10*(6), 613–627. doi:10.1007/s10732-005-5432-5

Renaud, J., Laporte, G., & Boctor, F. F. (1996). A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research*, *23*(3), 229–235. doi:10.1016/0305-0548(95)O0026-P

Ruttanateerawichien, K., Kurutach, W., & Pichpibul, T. (2014). An improved golden ball algorithm for the capacitated vehicle routing problem. In *Bio-Inspired Computing-Theories and Applications* (pp. 341–356). Berlin, Heidelberg: Springer. doi:10.1007/978-3-662-45049-9_56

Ruttanateerawichien, K., Kurutach, W., & Pichpibul, T. (2016, June). A new efficient and effective golden-ball-based technique for the capacitated vehicle routing problem. In *Proceedings of the 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. IEEE. doi:10.1109/ICIS.2016.7550764

Salhi, S., Thangiah, S. R., & Rahman, F. (1998). A genetic clustering method for the multi-depot vehicle routing problem. In *Artificial Neural Nets and Genetic Algorithms* (pp. 234–237). Vienna: Springer. doi:10.1007/978-3-7091-6492-1_51

Stodola, P., & Mazal, J. (2015). The Tactical Model based on a Multi-Depot Vehicle Routing Problem. *New Developments in Pure and Applied Mathematics*, (12), 26-31.

Surekha, P., & Sumathi, S. (2011). Solution to multi-depot vehicle routing problem using genetic algorithms. *World Applied Programming*, *1*(3), 118–131.

Thangiah, S. R., & Salhi, S. (2001). Genetic clustering: An adaptive heuristic for the multidepot vehicle routing problem. *Applied Artificial Intelligence*, *15*(4), 361–383. doi:10.1080/08839510151087293

Tillman, F. A. (1969). The multiple terminal delivery problem with probabilistic demands. *Transportation Science*, *3*(3), 192–204. doi:10.1287/trsc.3.3.192

Toth, P., & Vigo, D. (Eds.). (2002). *The vehicle routing problem*. Society for Industrial and Applied Mathematics. doi:10.1137/1.9780898718515

Web, N. E. O. (2017). *Networking and Emerging Optimization*. Malaga: University of Malaga. Retrieved 15 January 2017 from http://neo.lcc.uma.es/vrp/vrp-instances/multiple-depotvrp- instances/

Lahcene Guezouli received the bachelor's degree in science of nature from the High School of Batna, Algeria, in 2006 and the Master degree in computer science from the University of Batna, Algeria, in 201l. Lahcene is currently a PhD student in Industrial Engineering at the University of Batna, Algeria.

Mohamed Bensakhria is a PhD student in the Industrial Engineering Department at the University of Batna, Algeria. He received his master's degree in industrial engineering and a bachelor's degree in computer science from University of Batna. His current research focuses on the analysis and development of robust and efficient techniques for the design and operation of production and inventory routing.

Samir Abdelhamid received his master's degree in computer science from the University of Constantine, Algeria in 1985 and his PhD degree in computer science from the University of Batna, Algeria, in 2010. He was in charge of post-graduation studies in University of Batna (Algeria) from 1991 to 2001. He was also head of Computer Science Department in University of Batna from 2003 to 2006. He is the Vice Dean of the faculty of technology. His major interests are speech recognition, expert systems. He is an author of many conference proceedings papers and journal papers in his research areas.

International Journal of Applied Evolutionary Computation
Volume 9 • Issue 1 • January-March 2018

An Improved Generalized Quantum-Inspired Evolutionary Algorithm for Multiple Knapsack Problem

Sulabh Bansal, School of Computing and Information Technology, Manipal University Jaipur, Jaipur, India
C. Patvardhan, Faculty of Engineering, Dayalbagh Educational Institute (DEI), Agra, India

ABSTRACT

This article describes how the 0/1 Multiple Knapsack Problem (MKP), a generalization of popular 0/1 Knapsack Problem, is NP-hard and harder than simple Knapsack Problem. Solution of MKP involves two levels of choice – one for selecting an item to be placed and the other for selecting the knapsack in which it is to be placed. Quantum Inspired Evolutionary Algorithms (QIEAs), a subclass of Evolutionary algorithms, have been shown to be effective in solving difficult problems particularly NP-hard combinatorial optimization problems. QIEAs provide a general framework which needs to be customized according to the requirements of a given problem to obtain good solutions in reasonable time. An existing QIEA for MKP (QIEA-MKP) is based on the representation where a Q-bit collapse into a binary number. But decimal numbers are required to identify the knapsack where an item is placed. The implementation based on such representation suffers from overhead of frequent conversion from binary numbers to decimal numbers and vice versa. The generalized QIEA (GQIEA) is based on a representation where a Q-bit can collapse into an integer and thus no inter conversion between binary and decimal is required. A set of carefully selected features have been incorporated in proposed GQIEA-MKP to obtain better solutions in lesser time. Comparison with QIEA-MKP shows that GQIEA-MKP outperforms it in providing better solutions in lesser time for large sized MKPs. The generalization proposed can be used with advantage in other Combinatorial Optimization problems with integer strings as solutions.

KEYWORDS

Combinatorial Optimization, Hybrid Evolutionary Algorithm, Multiple Knapsack Problem, Quantum Inspired Evolutionary Algorithm

DOI: 10.4018/IJAEC.2018010102

1. INTRODUCTION

0-1 Multiple Knapsack Problem (MKP) is a generalization of the standard 0-1 Knapsack Problem (KP) where multiple knapsacks are required to be filled instead of one.

Given a set of n items with their profits p_j and weights w_j, $j \in \{1,\ldots,n\}$. and m knapsacks with capacities c_i, $i \in \{1,\ldots,m\}$. the MKP is to select a subset of items to fill given m knapsacks such that the total profit is maximized and sum of weights in each knapsack i doesn't exceed its capacity c_i.

$$\text{maximize: } \sum_{i=1}^{m}\sum_{j=1}^{n} p_j x_{ij} \tag{1}$$

$$\text{subject to: } \sum_{j=1}^{n} w_j x_{ij} \le c_i, i \in \{1,\ldots,m\}, \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} \le 1, j \in \{1,\ldots,n\}, \tag{3}$$

$$x_{ij} \in \{0,1\}, \forall i \in \{1,\ldots,m\}, \forall j \in \{1,\ldots,n\}, \tag{4}$$

where x_{ij} .= 1 if item j is assigned to knapsack i, x_{ij} .= 0 otherwise and coefficients p_j, w_j and c_i are positive integers.

In order to avoid any trivial case, the following assumptions are made

1. Every item has a chance to be placed at least in largest knapsack:

$$\max_{j \in N} w_j \le \max_{i \in \{1,\ldots,m\}} c_i. \tag{5}$$

2. The smallest knapsack can be filled at least by the smallest item:

$$\min_{i \in \{1,\ldots,m\}} c_i \le \min_{j \in N} w_j. \tag{6}$$

3. There is no knapsack which can be filled with all N items:

$$\sum_{j=1}^{n} w_j \ge c_i, \forall i \in \{1,\ldots,m\} \tag{7}$$

The subset sum variant of MKP having $p_j = w_j, j \in \{1,\ldots,n\}$, is known as Multiple Subset Sum Problem (MSSP).

The MKP have several applications. An application is seen when scheduling jobs on processors where some machines are unavailable for a fixed duration or some high priority jobs are pre-assigned to processors (Diedrich & Jansen, 2009). A real-world application of MKP is the problem of cargo loading where some containers need to be chosen from a set of *n* containers to be loaded in m vessels with different loading capacities for the shipment of the containers (Eilon & Christofides, 1971). Another real-world problem for MSSP is mentioned in (Kellerer, Pferschy, & Pisinger, 2004, p. 287) from a company producing objects of marble.

The MKP problem is strongly NP-complete. Some approximation algorithms exist for MKP. Kellerer (Kellerer H., 1999) presented a Polynomial Time Approximation Scheme for MKP with identical capacities. Chekuri & Khanna (Chekuri & Khanna, 2006) generalized it and presented the PTAS for MKP. However, no Fully Polynomial Time Approximation Scheme is possible for MKP (Chekuri & Khanna, 2006).

Martello & Toth(Martello & Toth, Solution of the zero-one multiple knapsack problem, 1980; Martello & Toth, Knapsack Problems: Algorithms and Computer Implementations, 1990) propsoed a heuristic algorithm, called MTHM, for solving the MKP. This algorithm consists of three phases as follows. During first phase, an initial feasible solution is obtained by applying the Greedy algorithm to the first knapsack; a set of remaining items is obtained, then the same procedure is applied for all knapsacks iteratively. The initial solution is improved during the second phase by swapping each pair of items assigned to different knapsack. Then a new item is inserted if possible such that the total profit is increased. Each selected item is replaced by one or more remaining items during the last phase if it enhances the profit sum. The details are available in (Martello & Toth, Knapsack Problems: Algorithms and Computer Implementations, 1990, pp. 179-181).

Quantum-Inspired Evolutionary Algorithm (QIEA) is a population based search technique where the representation of individuals and operators involved in generation of new individuals are both designed based on concepts from Quantum Computing. Various forms of QIEAs have been used to solve a variety of difficult problems (Yang, Wang, & Jiao, 2004; Patvardhan, Narayan, & Srivastav, 2007; Platel, Schliebs, & Kasabov, 2007; Sailesh Babu, Bhagwan Das, & Patvardhan, 2008; Mani & Patvardhan, 2010; Wang & Li, 2010; Yang, Wang, & Jiao, 2010; Xiao, Yan, Zhang, & Tang, 2010; Arpaia, Maisto, & Manna, 2011; Patvardhan, Prakash, & Srivastav, (ICOREM 2009), 2012). QIEA, originally proposed by Han & Kim (Han & Kim, 2002), is directly applicable in problems where a Q-bit individual represents a linear superposition of binary solutions. Alegria & Tupac (Alegria & Tupac, 2013) proposed a generalization of the QIEA (GQIEA), where a Q-bit individual is a superposition of combinatorial solutions represented as a string of non-binary integers, to improve performance in combinatorial optimization.

QIEA-MKP (Patvardhan, Bansal, & Srivastav, Balanced quantum-inspired evolutionary algorithm for multiple knapsack problem, 2014) is a hybridized QIEA to solve MKP. It is based on the popular representation used in QIEAs where Q-bit represents the superposition of binary solutions. The observation of Q-bit individuals results in binary solutions. Thus, such a representation puts an overhead of frequent conversions from binary to decimal and vice versa. A selected item can be placed only in single knapsack at a time for a solution of MKP. But the probability of an item to be selected in to different knapsacks keeps on changing during evolution. These probabilities should be normalized such that their sum remains equal to 1. QIEA-MKP doesn't normalize these probabilities in that way. GQIEA provides a solution for the issue through the use a generalized representation of Q-bits (GQ-bit) and their update operator (Alegria & Tupac, 2013).

In this paper an effective GQIEA is proposed for MKP. The representation for GQ-bit, the process for observing a GQ-bit and the process to update a GQ-bit has been defined as required for MKP. The proposed GQIEA, dubbed GQIEA-MKP, is hybridized with an existing heuristic for MKP known as MTHM (Martello & Toth, Solution of the zero-one multiple knapsack problem, 1980). Various other features like biased initialization of GQ-bit individuals, local search, mutation, re-initialization etc. have been designed and incorporated in GQIEA-MKP. GQIEA-MKP balances heuristics to promote exploitation for obtaining good solutions and several other features like mutation and re-initialization to increase randomness and power to explore unreached areas. A comparison shows on a variety of randomly generated instances of MKP that GQIEA-MKP outperforms QIEA-MKP.

The methodology adopted here to design GQIEA can be used to design similar GQIEAs to solve other combinatorial problems where solutions are represented as sequences of integers.

The rest of the paper is organized as follows. A brief description of the QIEA-MKP is given in section 2. In section 3 GQIEA_MKP is described. Computational performance of QIEA-MKP and GQIEA-MKP is presented in section 4. Conclusions are presented in section 5.

2. QIEA -MKP

The QIEA introduced in (Han & Kim, 2002) use a vector of Q-bits to represent the probabilistic state of individual. Each Q-bit is represented as $q_i = \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}$ where $\left|\alpha_i\right|^2$ .s the probability of state being 1 and $\left|\beta_i\right|^2$ is the probability of state being 0 such that $\left|\alpha_i\right|^2 + \left|\beta_i\right|^2 = 1$. Thus, a Q-bit string with n bits represents a superposition of 2^n binary states. QIEA uses the Q-gate, for example a rotation gate to update the Q-bits as follows:

$$\begin{bmatrix} \alpha_i^{t+1} \\ \beta_i^{t+1} \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta_i) & -\sin(\Delta\theta_i) \\ \sin(\Delta\theta_i) & \cos(\Delta\theta_i) \end{bmatrix} \begin{bmatrix} \alpha_i^t \\ \beta_i^t \end{bmatrix} \tag{8}$$

where, α_i^{t+1} and β_i^{t+1} denote probabilities for i^{th} Q-bit in $(t + 1)^{th}$ iteration and $\Delta\theta_i$ is equivalent to the step size in typical iterative algorithms in the sense that it defines the rate of movement towards the currently perceived optimum.

The implementation of QIEAs to solve MKP requires special consideration as it calls for subset allocation and selection. A solution of MKP indicates the selection status of each item and the knapsack in which it is placed. A solution consists of a string of length n of integer values ranging from 0 to m. 0 indicates that the corresponding item is not selected. Otherwise, it mentions the knapsack in which the item is placed.

To implement QIEA-MKP (Patvardhan, Bansal, & Srivastav, Balanced quantum-inspired evolutionary algorithm for multiple knapsack problem, 2014) two binary strings (arrays) are used to represent one complete solution. A binary string of length n, having one bit for each item conveying about its selection status, and another of length (n* $\log_2 m$) contains the index in binary of knapsack in which it is packed. The integers in {1, …, m} are represented by a bit string of length $\log_2 m$. A qubit individual is represented by two components of lengths n and n* $\log_2 m$ corresponding to two binary strings of solution.

Two components of population of qubit individuals after t^{th} iteration are represented using Q1(t) and Q2(t), P1(t) and P2(t) represent the first and second components of population of individual solutions, B1(t) and B2(t) is the set of best solutions corresponding to first and second component of each individual. The individuals of populations q_j^t is composed of $q1_j^t$ and $q2_j^t$, p_j^t is composed of $p1_j^t$ and $p2_j^t$, and b_j^t is composed of $b1_j^t$ and $b2_j^t$ for each $j \in \{1,\ldots,n\}$; b have b1 and b2. In the following paragraphs a brief description of procedures implemented in QIEA-MKP is given.

Initialize (q_j^t): The first component (say $q1_j^t$) is initialized as follows. The items are considered divided in to 3 classes based on where they lie in order of preference. Qubits for items lying in first class (third class) are assigned values closer to 1 (0) so that they have high (low) probability of collapsing to value 1. Items lying in the second class require more processing for convergence to either 0 or 1, hence intermediate values between 0 and 1 are assigned to them. As a result, the hybridised algorithm starts exploiting the area or region in solution space having solutions closer to optimal with higher probability. Qubits of second component (say $q2_j^t$) are assigned the value $1/\sqrt{2}$.

Observe: The procedure collapses the first (second) component qubit individual/s to generate first (second) component of solution individual/s.

Repair: Solutions are repaired based on phase 1 of description of MTHM in (Martello & Toth, Knapsack Problems: Algorithms and Computer Implementations, 1990). Overheads add up due to conversion required for binary representation of decimal numbers used in second component of solution representation.

Update q_j^t based on b_j^t: Rotates the qubits $q1_j^t$ towards bits in $b2_j^t$ and $q2_j^t$ towards bits in $b1_j^t$ as explained earlier and defined in (Han & Kim, 2002) using the rotation angle as 0.01.

Evaluate p_j^t **:** Sum the profits of items set to 1 in $p1_j^t$.

3. GQIEA-MKP

A GQ-bit individual in the proposed GQIEA-MKP, inspired from (Alegria & Tupac, 2013), is represented as an array of m+1 real

$$GQ_i = \begin{bmatrix} \alpha_0 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix} \tag{9}$$

where, $0 \le \alpha_i \le 1$ and $\sum_{i=1}^{m} \alpha_i = 1$. Total number of different possible states it represents is m+1, instead of just two. In this representation, α_0 is the probability of an item not getting selected in any of the knapsacks, $\pm_i$, for $i \in \{1, \ldots, m\}$ is the probability of item being put in i[th] knapsack. Similar to the Q-gate in the original QIEA, two GQ-gate operators: the arithmetic GQ-gate and the geometric GQ-gate have been proposed in (Alegria & Tupac, 2013). Here arithmetic GQ-gate is used to update GQ-bit individuals as described in section 3.3.

The items having a greater profit by weight ratio are considered to have higher probability of their inclusion in the optimal solution. Thus, the items in input are sorted in the decreasing order of their profit by weight ratio. This sorting is used to initialize GQ-bit individuals as explained in section 3.1 so that they can generate better solutions. The sorted input is utilized to improve repair procedure also, section 3.4, so that it provides better solutions.

A brief description of the primary operations in GQIEA-MKP and the features applied for improvement follows.

Figure 1. Pseudo-code for Initialize function in GQIEA-MKP

```
  Procedure Initialize(q_j^t)
1 let C be the sum of all capacities c_i and c_a be average capacity;
2 let W be sum of weights w_i of all items and w_a be the average weight
3 for i from 1 to n do{
4       α_0 of q_j^t[i] is set to 1.0 – q[i];
5       for j from 1 to m
6           α_j of q_j^t[i] is set to q[i] * c_j / C + (w_i-w_a)* (c_j-c_a) / (w_a*c_a);
7 }//* for i*//
```

3.1. Initialize

The best solution is initialized with the good solution obtained using MTHM heuristic as described in(Martello & Toth, Solution of the zero-one multiple knapsack problem, 1980; Martello & Toth, Knapsack Problems: Algorithms and Computer Implementations, 1990).

To initialize GQ-bit individuals an array, q, of real numbers of length n is first initialized as follows. The items are considered divided in to 3 classes based on where they lie in the order of preference. The values for items lying in first class (third class) are assigned values closer to 1 (0) so that they have high (low) probability of collapsing to value 1. Items lying in the second class require more processing for convergence to either 0 or 1, hence intermediate values between 0 and 1 are assigned to them. As a result, the hybridised algorithm is expected to start exploiting the area or region in solution space having solutions closer to optimal with higher probability. The initialization of q_j^t is further done using q as in Pseudo-code in Figure 1.

Since a knapsack with larger capacity can hold more items, the probability of an item being put into a particular knapsack is considered to be proportional to the weight of the knapsack divided by the sum of weights of all knapsacks. The probability of an item also depends on the relation between weight of an item and the capacity of knapsack. Initially, lighter items have higher probability to be placed in smaller knapsacks and heavier items have higher probability to be placed in larger knapsacks. This is reflected in steps 5 and 6 in Figure 1.

The GQ-bit component of i^{th} item corresponding to j^{th} knapsack is set to $q[i] * c_j / C + (w_i-w_a)* (c_j-c_a) / (w_a*c_a)$, where q[i] is the value of Q-bit assigned assined to i^{th} item based on above method, c_j is the capacity of j^{th} knapsack, C is the sum total of capacities of all knapsacks, w_i is the weight of i^{th} item, w_a is the average weight of an item, c_a is the average capacity of a knapsack.

3.2. Observe

The procedure is described in Pseudo-code of Figure 2. To understand the approach, consider a 3 state GQ-bit system $\begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix}$. An integer, p, need to be generated ranging between 0 and 2 to define the state of an item in the solution. if $(\alpha0+\alpha1+ \alpha2)$ is not equal to 1 then it indicates an error. Otherwise a random variable r_1 is generated and p is assigned the value as follows

Figure 2. Pseudo-code for Observe function in GQIEA-MKP

```
Procedure Observe(q_j^t) //return p_j^t
for i from 1 to n do{
    let r be a random number; p_j^t[i]=0; N=0.0;
    for j from 1 to m
        N=N + α_j of q_j^t[i];
        if(r < N){ p_j^t[i]=j; break;}
    }//*for j*//
}//* for i*//
```

$$p = \begin{cases} 0, & r_1 < \alpha_0 \\ 1, \alpha_0 < r_1 < \alpha_1 \\ 2, \alpha_1 < r_1 < 1 \end{cases}$$

This idea extends to the m+1 GQ-bit system naturally when there are m knapsacks.

3.3 Update

q_j^t based on b_j^t: The procedure is described in pseudo-code in Figure 3. Let the j^th GQ-bit of a GQ-bit individual represent a superposition of m+1 states $\{s_0, s_2, \ldots, s_m\}$. If the state s_ϕ was found in j^th position of the best individual b_j^t of the generation, the new values of α_i are determined according to the arithmetic GQ-gate as defined in equation (10), where Δd is a parameter of the algorithm. Each GQ-bit is then normalized to ensure that $\sum_{i=1}^{r} \alpha_i = 1$ as shown in step 5 in Figure 3.

$$\alpha_i = \begin{cases} \alpha_i + \Delta d & if\, i = \phi \\ \alpha_i & if\, i \neq \phi \end{cases} \tag{10}$$

Figure 3. Pseudo-code for Update in GQIEA-MKP

```
  Procedure Update( q_j^t based on b_j^t) //
1 let p_j^t is solution observed using q_j^t;
2 for i from 1 to n do{
3     if( b_j^t is better than p_j^t){ Δ=0.1; d=1.0- α_{b_j^t[i]} of q_j^t[i];}
4     for j from 1 to m{
5         if( j <> b_j^t[i] ) q_j^t[i] = q_j^t[i]* (d-Δ/d);
6         else q_j^t[i] = q_j^t[i] + Δ;
7     }//*for j*//
8 }//* for i*//
```

Figure 4. Pseudo-code for Repair function in GQIEA-MKP

```
Procedure Repair(x)
let R_i, ∀i ∈ {1, … , m} be residue in knapsack i;
for i from 1 to m{
   while (R_i < 0) {
     let u be item of lowest preference in sorted input.
     remove u from knapsack i; R_i = R_i + w_u;
    }//*while*//
}//* for i*//
for j from 1 to n {
     let k be knapsack of minimum capacity such that R_i ≥ w_j )
     {x_kj ←1; R_k = R_k − w_j;}
} //*for j*//
```

3.4. Improved Repair Function

In GQIEA-MKP, the repair function is modified to improve the quality of solutions while making them feasible based on the phase 1 of MTHM heuristic as described in (Martello & Toth, Knapsack Problems: Algorithms and Computer Implementations, 1990). This improves the speed of convergence. As explained earlier the items are sorted in order of their preference to include them into a knapsack. So, in each repair step, items having low preference (closest to the end in the sorted order) are removed and items having high preference (closest to the beginning in the sorted order) are added as necessary. The knapsacks are assumed to be sorted in order of their increasing capacity and that's the order in which they are considered when items are added into knapsacks (see Figure 4).

3.5. Improving the Local Best Solutions

The local best solutions are further improved in two stages based on the phases 2 and 3 of MTHM heuristic (Martello & Toth, Solution of the zero-one multiple knapsack problem, 1980; Martello & Toth, Knapsack Problems: Algorithms and Computer Implementations, 1990). In first stage, Improve1, it tries to exchange every pair of items assigned to different knapsacks along with inserting a new item

Figure 5. Pseudo-code for Improve1

```
Procedure Improve1(x)
let R_i ← c_i − Σ_{j=1}^{n} w_j x_ij,  ∀i ∈ {1, … , m};
for each pair i and j in {1,…,n}{
  if     (∃ u, v in {1, … , m}| x_ui = 1 and x_vj = 1 and (R_u ≥ w_j − w_i) and (R_v ≥ w_i −
  w_j)){
     if (∃ k ∈ {1, … , n} | (R_u ≥ w_j + w_k − w_i) or (R_v ≥ w_i + w_k − w_j)){
        let k = ( min_{∀s∈{1,…,n}} s | (R_u ≥ w_j + w_s − w_i) or (R_v ≥ w_i + w_s − w_j))
        if ((R_u ≥ w_j + w_s − w_i){ x_ui ← 0; x_vj ← 0; x_uj ← 1; x_vi ← 1; x_uk ← 1; }
        else { x_ui ← 0; x_vj ← 0; x_uj ← 1; x_vi ← 1; x_vk ← 1; }
      }//* if*//
    }//* if*//
}//*for pair i and j*//
```

Figure 6. Pseudo-code for Improve2

Procedure Improve2(x)

let $R_i \leftarrow c_i - \sum_{j=1}^{n} w_j x_{ij}, \quad \forall i \in \{1, \ldots, m\}$;
for $\forall i \in \{1, \ldots, n\}$ {
  if ($\exists u \in \{1, \ldots, m\} | x_{ui} = 1$){
    for $\forall j \in \{1, \ldots, n\} | x_{vj} = 0 \;\; \forall v \in \{1, \ldots, m\}$ {
      if($R_u + w_i - w_j \geq 0$) $P_j \leftarrow p_j - p_i$; else $P_j \leftarrow 0$;
    }//* for j*/
    let $P_k = \max_{\forall j \in \{1,\ldots,n\}} P_j$;
    if($P_k > 0$) { $x_{ui} \leftarrow 0$; $x_{uk} \leftarrow 1$; $R_u \leftarrow R_u + w_i - w_k$; }
  }//*If u *//
}//*for i *//

so that total profit is increased. In second stage, Improve2, every selected item (starting from last in the sorted order) is tried to be replaced by one of the remaining items so that the total profit sum is increased. The pseudo-codes of these procedures are given in Figure 5 and Figure. 6.

3.6. Mutation of Solutions Appearing to be Stuck in Local Optimum

The proposed GQIEA-MKP have a the tendency of getting stuck in local optima due several heuristic based deterministic features discussed above in sections 3.1, 3.4 and 3.5. To combat this tendency, the generated solution is mutated such that randomly selected 2-3 bits in the solution vector are changed to 0, when it is close to global best solution (Hamming distance is less than 2) found so far it. The partial solution is improved using ImroveStage1. This operator improves diversity with small computational effort. It helps to explore the solution space around a current solution such that an optimal in vicinity is not missed. This improves the chances of finding optimal in case it is in vicinity of the converging solution.

3.7. Re-Initialization of GQ-Bit Individuals

After a sequence of generations a GQ-bit individual converges such that all solutions generated from it are same. No new solutions can be generated using such individuals. Thus, when a GQ-bit individual generates same solution for more than 3 times out of 5, it is reset as in initialization. It increases the diversity of solutions explored through the Q-bit individuals with small computational effort.

3.8 Faster Local Exploitation of GQ-Bit Individuals Before Global Exploration

The evolution steps of a basic QIEA are executed with the specialized local search procedure for some iteration to update some of the GQ-bit individuals. The steps listed in the following are performed on half of the GQ-bit individuals for small number of times (empirically set as 15 for this work).

- Make
- Repair
- Improve1
- Improve2
- Update

Figure 7. Pseudo-code for GQIEA-MKP

```
Improved GQIEA-MKP
SortGreedy the Input;
t ← 0; b ← 0;
Initialize Q-bit Individuals Q(t);
Initialize b
Observe P(t) from Q(t);
Repair (P(t));
Initialize B(T) as P(T);
for each j ∈ {1, ... , n/4} {
    Observe p_j^t from q_j^t;
    Repair ( p_j^t);
    Improve1( p_j^t);
    Improve2( p_j^t);
    if ( p_j^t is better than b_j^t) b_j^t ← p_j^t;
}
while ( t<MaxIterations) {
   t ← t+1;cnt_j ← 0;
   for r from 0 to η_1 do{
      for s from 0 to η_2 do{
            Observe P(s) from Q(t);
            Repair(P(s));
            for each j ∈ {1, ... , n} {
                 if (HamDistance( p_j^s,b)<2) {Mutate( p_j^s);Improve1( p_j^s);}
            }//*for j*//
            for each j ∈ {1, ... , n}
               if ( p_j^s better than p_j^t ) p_j^t ← p_j^s; else if ( p_j^s equal p_j^t ) cnt_j ← cnt_j + 1;
      } //*for s*//
      if(cnt_j > 3){
            ReInitialize( q_j^t) for each j ∈ {1, ... , n};
      }//*if cnt_j*//
      for j ∈ {1, ... , n} Improve2( p_j^t) ;
      for each j ∈ {1, ... , n} if ( p_j^t is better than b_j^t) b_j^t ← p_j^t;
      for each j ∈ {1, ... , n} if (b_j^t is better than b) b ← b_j^t ;
      for each j ∈ {1, ... , n} Update q_j^t based on b_j^t ;
   } //*for r*//
   for each j ∈ {1, ... , n} Update q_j^t based on b;
} // *while*//
```

The execution of these steps will exploit intensively the area represented by the initial GQ-bit individuals. This execution may solve some of the easier instances earlier. The resulting GQ-bit individuals favour the smaller region in search space around better solutions.

The features and steps explained above are put together to design the proposed GQIEA-MKP. The Pseudo-code of the GQIEA-MKP is given in Figure 7. In the pseudo-code; t refers to the current iteration, population of GQ-bit individuals after t^{th} iteration is represented using Q(t), P(t) represent the population of individual solutions, B(t) is the population of best solutions corresponding to each GQ-bit individual, c_i is the capacity of the i^{th} knapsack. Individuals represented by Q(t) are referred to as q_j^t, individuals in P(t) are referred to as p_j^t, similarly individuals in B(t) are referred to as b_j^t for each $j \in n$; b refers to global best solution found so far. Some other functions used are as follows:

HamDistance (p_j^s, b): Hamming distance is returned as number of places in two strings having different bits.

4. RESULTS AND DISCUSSION

The experiments are done on Intel® Xeon® Processor E5645 (12M Cache, 2.40 GHz, 5.86 GT/s Intel® QPI). The machine uses Red Hat Linux Enterprise 6.

The solutions converged for most of the problem instances considered here within 10 iterations hence maxIterations is set to 10. Empirically, η_1 and η_2 are set to 5 and population size is set to 10.

The performance is observed on randomly generated instances having elements 1000, 5000 and 10000 with number of knapsacks as 2, 5 and 10.These instances are randomly generated, using the generator of instances available at the Pisinger's home page i.e. http://www.diku.dk/~pisinger/codes.html. Two types of instances have been generated, first the strongly correlated where weights w_j are distributed in [1, R] and profits p_j are calculated as $p_j = w_j + R/10$ and second is Uncorrelated where weights w_j and profits p_j are independently distributed in [1, R]. Such instances correspond to a real-life situation where the return is proportional to the investment plus some fixed charge for each project.

Two different cases of capacities are considered: similar and dissimilar.

In case of similar capacities the first m-1 capacities $c_i, i \in \{1,\ldots,m-1\}$ are distributed in the range

$$\left[0.4\sum_{j=1}^{n} w_j / m, 0.6\sum_{j=1}^{n} w_j / m\right] \forall i \in \{1,\ldots,m-1\} \tag{11}$$

In case of dissimilar capacities the first m-i capacities c_i, are distributed in the range

$$\left[0, 0.5\sum_{j=1}^{n} w_j - \sum_{k=1}^{i-1} c_k\right] \forall i \in \{1,\ldots,m-1\} \tag{12}$$

The last capacity c_m is chosen as

$$c_m = 0.5\sum_{j=1}^{n} w_j - \sum_{i=1}^{m-1} c_i \tag{13}$$

45 instances are generated using code available in (Pisinger D.) in each of the three classes of instances namely Strongly Correlated instances with dissimilar capacities (STRCORRDISS), Strongly Correlated instances with similar capacities (STRCORRSIM), Uncorrelated instances with dissimilar capacities (UNCORRDISS) and Uncorrelated instance with similar capacities (UNCORRSIM). The Optimal profit (found using the Mulknap algorithm of (Pisinger D., 1999) available at (Pisinger D.)), total capacity constraint (actually distributed in multiple knapsacks) and the value obtained using heuristic has been reported for all of these instances in Table 1. The number of knapsacks (m) in these

Table 1. Details of Instances generated in four different classes.

n_m_s	STRCORRDISS			STRCORRSIM			UNCORRDISS			UNCORRSIM		
	Optimal	Capacity	Heuristic	Optimal	Capacity	Heuristic	Optimal	Capacity	Heuristic	Optimal	Capacity	Heuristic
1000_2_1	258384	251354	258290	258384	251354	258383	391675	256767	391653	391675	256767	391653
1000_2_2	260485	253465	260483	260485	253465	260452	409026	250741	408994	409026	250741	408990
1000_2_3	258942	251912	258721	258942	251912	258940	406439	248256	406426	406439	248256	406413
1000_2_4	255726	248626	255725	255726	248626	255681	408580	248569	408564	408580	248569	408564
1000_2_5	265225	258235	265074	265225	258235	265213	415860	254070	415846	415860	254070	415835
5000_2_1	1311854	1276724	1311854	1311854	1276724	1311752	2003828	1262124	2003806	2003828	1262124	2003821
5000_2_2	1311739	1276659	1311656	1311739	1276659	1311703	2045748	1253440	2045737	2045748	1253440	2045737
5000_2_3	1287821	1252561	1287755	1287821	1252561	1287821	2019647	1250828	2019633	2019647	1250828	2019630
5000_2_4	1298230	1262920	1298230	1298230	1262920	1298230	2034403	1265161	2034400	2034403	1265161	2034387
5000_2_5	1315856	1280766	1315807	1315856	1280766	1315856	2026015	1271477	2025987	2026015	1271477	2025987
10000_2_1	2607305	2536995	2607147	2607305	2536995	2607281	4040310	2536030	4040301	4040310	2536030	4040301
10000_2_2	2584205	2513685	2584015	2584205	2513685	2584205	4067232	2531903	4067224	4067232	2531903	4067221
10000_2_3	2587728	2517308	2587728	2587728	2517308	2587637	4037917	2510003	4037911	4037917	2510003	4037906
10000_2_4	2601537	2531107	2601537	2601537	2531107	2601537	4054712	2506649	4054704	4054712	2506649	4054700
10000_2_5	2619338	2549058	2619333	2619338	2549058	2619338	4029938	2533368	4029933	4029938	2533368	4029915
1000_5_1	258384	251354	258125	258384	251354	258224	391675	256767	391627	391675	256767	391605
1000_5_2	260485	253465	260231	260485	253465	260270	409026	250741	408994	409026	250741	408922
1000_5_3	258942	251912	258652	258942	251912	258907	406439	248256	406377	406439	248256	406364
1000_5_4	255726	248626	255711	255726	248626	255438	408580	248569	408552	408580	248569	408484
1000_5_5	265225	258235	264857	265225	258235	265157	415860	254070	415835	415860	254070	415823
5000_5_1	1311854	1276724	1311710	1311854	1276724	1311745	2003828	1262124	2003805	2003828	1262124	2003805
5000_5_2	1311739	1276659	1311656	1311739	1276659	1311729	2045748	1253440	2045725	2045748	1253440	2045714
5000_5_3	1287821	1252561	1287411	1287821	1252561	1287811	2019647	1250828	2019630	2019647	1250828	2019605
5000_5_4	1298230	1262920	1297988	1298230	1262920	1298152	2034403	1265161	2034387	2034403	1265161	2034366
5000_5_5	1315856	1280766	1315466	1315856	1280766	1315614	2026015	1271477	2025962	2026015	1271477	2025962
10000_5_1	2607305	2536995	2606951	2607305	2536995	2607295	4040310	2536030	4040263	4040310	2536030	4040288
10000_5_2	2584205	2513685	2583961	2584205	2513685	2584194	4067232	2531903	4067211	4067232	2531903	4067206
10000_5_3	2587728	2517308	2587616	2587728	2517308	2587724	4037917	2510003	4037886	4037917	2510003	4037911
10000_5_4	2601537	2531107	2601523	2601537	2531107	2601439	4054712	2506649	4054695	4054712	2506649	4054689
10000_5_5	2619338	2549058	2619087	2619338	2549058	2619328	4029938	2533368	4029927	4029938	2533368	4029920
1000_10_1	258383	251354	258126	258384	251354	258091	409602	253519	409452	391675	256767	391526

continued on following page

Table 3. Continued

n_m_s	STRCORRDISS			STRCORRSIM			UNCORRDISS			UNCORRSIM		
	Optimal	Capacity	Heuristic	Optimal	Capacity	Heuristic	Optimal	Capacity	Heuristic	Optimal	Capacity	Heuristic
1000_10_2	260485	253465	259936	260485	253465	260447	401741	255797	401615	409026	250741	408876
1000_10_3	258942	251912	258347	258942	251912	258449	406439	248256	406360	406439	248256	406256
1000_10_4	255726	248626	255370	255726	248626	254850	403660	256177	403617	408580	248569	408441
1000_10_5	265225	258235	264857	265225	258235	265087	406072	256221	405983	415860	254070	415706
5000_10_1	1311854	1276724	1311262	1311854	1276724	1311535	2024918	1275798	2024879	2003828	1262124	2003772
5000_10_2	1311739	1276659	1311569	1311739	1276659	1311710	2045748	1253440	2045696	2045748	1253440	2045621
5000_10_3	1287821	1252561	1287411	1287821	1252561	1287749	2019647	1250828	2019607	2019647	1250828	2019576
5000_10_4	1298230	1262920	1297988	1298230	1262920	1297908	2034403	1265161	2034373	2034403	1265161	2034333
5000_10_5	1304030	1268840	1303896	1315856	1280766	1315633	2026015	1271477	2025962	2026015	1271477	2025927
10000_10_1	2607305	2536995	2607178	2607305	2536995	2607256	4040310	2536030	4040253	4040310	2536030	4040263
10000_10_2	2584205	2513685	2584132	2584205	2513685	2583550	4067232	2531903	4067187	4067232	2531903	4067179
10000_10_3	2587728	2517308	2587616	2587728	2517308	2587570	4061148	2509396	4061129	4037917	2510003	4037853
10000_10_4	2601537	2531107	2601011	2601537	2531107	2601507	4035606	2513796	4035592	4054712	2506649	4054669
10000_10_5	2619338	2549058	2619179	2619338	2549058	2619318	4029938	2533368	4029914	4029938	2533368	4029889

instances is 2, 5 or 10, total number of items (n) is 1000, 5000 or 10,000 and 5 different instances have been generated for each combination (total combinations 9) which are distinguished using the seed value as 1 to 5. The instances in each class have been distinguished using acronym n_m_s where n is the number of elements, m is the number of knapsacks and the s refers to seed.

The experiments have been performed using QIEA-MKP of (Patvardhan, Bansal, & Srivastav, Balanced quantum-inspired evolutionary algorithm for multiple knapsack problem, 2014) and proposed GQIEA-MKP. For both of these cases the tests have been done using population size of 20 and 100. Tables 2 to 5 compares the results obtained using QIEA-MKP and GQIEA-MKP. The best value obtained and the difference of best value from the average value over 30 runs has been shown in same column separated using pipe '|'. The average time taken to reach the best solution over 30 runs and average FES (Function Evaluations) has been shown in same column similarly. The largest best value provided by any of the four implementations of algorithms that is GQIEA_MKP(100), GQIEA_MKP(20) QIEA_MKP(100) and QIEA_MKP(20) in the table is darkened. The algorithm is made to terminate before the fixed maximum number of iterations (MAXIterations), when the optimal solution is found.

For strongly correlated instances having dissimilar capacities the results are as follows

- GQIEA_MKP(100) provides largest best value for 38 out of 45 instances.
- GQIEA_MKP(20) provides largest best value for 25 out of 45 instances.
- QIEA_MKP(100) provide largest best value in 13 instances
- while QIEA_MKP(20) provide only for 7 instances.
- However, Out of 45 instances for 7 instances all four algorithms provide same best value, for 6 instances only QIEA_MKP(100) provides largest best value, for 11 instances only two versions of GQIEA provide largest best value, for 2 instances only GQIEA_MKP(20) provides largest best value while for 19 instances only GQIEA_MKP(100) provides largest best value.
- Where all four types of instances provide same largest value, versions of GQIEAs are more consistent with better average values and significantly faster.
- In the instances for which both GQIEAs provide same best value, GQIEA_MKP(100) provides better average value or takes less time on an average.
- In the instances for which GQIEA_MKP(20) provides largest best value, GQIEA_MKP(100) provides better average value.

For strongly correlated instances having similar capacities the results are as follows

- GQIEA_MKP(100) provides largest best value for 44 out of 45 instances.
- GQIEA_MKP(20) provides largest best value for 42 out of 45 instances.
- QIEA_MKP(100) provide largest best value in 17 instances
- while QIEA_MKP(20) provide only for 11 instances.
- However, Out of 45 instances for 11 instances all four algorithms provide same best value, for 6 instances QIEA_MKP(100), GQIEA_MKP(20) and GQIEA_MKP(100) provide same largest best value, for 25 instances only two versions of GQIEA provide largest best value, for 1 instance only GQIEA_MKP(20) provides largest best value while for 2 instances only GQIEA_MKP(100) provides largest best value.
- Where all four types of instances provide same largest value, versions of GQIEAs are more consistent with better average values and significantly faster.
- In the instances where both GQIEA_MKP(20) and GQIEA_MKP(100) provide same best value, for 22 instances GQIEA_MKP(100) provide better average value. But the time required by GQIEA_MKP(100) is typically more than GQIEA_MKP(20).

Table 2. Comparison of results obtained for strongly correlated instances having dissimilar capacities using QIEA-MKP and GQIEA-MKP with population sizes of 20 and 100. The values obtained using optimal values and heuristic are shown as reference. Best value and difference between best and average is shown for QIEAs. Time to reach best solution in seconds and Function Evaluations (FES) are given.

n_m_s	QIEA-MKP(20)		QIEA-MKP(100)		GQIEA-MKP(20)		GQIEA_MKP(100)	
	Best |Best-Avg	T(sec)|FES	Best |Best-Avg	T(sec)|FES	Best |Best-Avg	T(sec)|FES	Best |Best-Avg	T(sec)|FES
1000_2_1	**258384**|2.1	3.336|56.1	**258384**|0.27	12.636|222.37	**258384**|1.9	0.45|54	**258384**|0.5	2.024|215.2
1000_2_2	**260485**|1.8	0.652|4.87	**260485**|1	18.396|162	**260485**|0	0.246|2.2	**260485**|0	0.249|2.2
1000_2_3	**258942**|6.8	5.43|47.7	**258942**|3.3	23.303|211.53	**258942**|0	0.311|3.8	**258942**|0	0.315|3.8
1000_2_4	255725|0	0.045|0	255725|0	0.044|0	**255726**|0	0.077|3.3	**255726**|0	0.079|3.3
1000_2_5	**265225**|3.5	3.924|63.53	**265225**|0.9	14.33|240.27	**265225**|1.6	0.62|54.3	**265225**|0.2	3.807|309.5
5000_2_1	**1311854**|0	1.075|0	**1311854**|0	1.04|0	**1311854**|0	0.088|0	**1311854**|0	0.062|0
5000_2_2	1311679|20	122.914|12.6	1311679|16.03	1681.838|181.77	**1311739**|0	11.17|1	**1311739**|0	11.186|1
5000_2_3	1287791|10	414.583|68.77	1287800|9.4	1732.235|295.7	**1287821**|0	16.33|2.2	**1287821**|0	16.343|2.2
5000_2_4	**1298230**|0	7.94|0	**1298230**|0	7.704|0	**1298230**|0	6.713|0	**1298230**|0	6.722|0
5000_2_5	1315835|23	184.401|32.63	1315836|15.4	1673.487|311.37	**1315856**|2	52.578|40.5	**1315856**|0	63.868|38.4
10000_2_1	2607248|27	1050.292|63.47	2607255|19.3	4416.376|283.17	**2607305**|1	211.426|29.6	**2607305**|0	1014.753|128.3
10000_2_2	2584146|25	2284.621|75.53	2584153|20.47	8888.503|313.57	**2584205**|1	863.77|42	**2584205**|0	3020.713|148.6
10000_2_3	2587728|0	86.604|0	2587728|0	84.957|0	**2587728**|0	77.947|0	**2587728**|0	78.008|0
10000_2_4	**2601537**|0	4.31|0	**2601537**|0	4.229|0	**2601537**|0	0.193|0	**2601537**|0	0.214|0
10000_2_5	2619333|0	4.308|0	2619333|0	4.202|0	**2619338**|2	227.629|29.1	**2619338**|0	955.099|128.1
1000_5_1	258379|9.1	9.876|55.93	258382|7.7	51.308|302.27	258383|4.1	1.803|59.2	**258384**|0.4	6.626|219.9
1000_5_2	260475|9.2	9.542|56.43	260476|3.2	44.469|272.83	260480|5.2	0.921|63.2	**260484**|4.7	3.76|252.4
1000_5_3	258931|8.2	10.168|54.43	258932|5.43	49.137|274.9	**258942**|6.3	1.696|61.8	**258942**|0.9	8.082|279.5
1000_5_4	255715|3.8	1.458|6.9	255716|3.67	16.344|90.03	**255726**|4	2.286|57.9	**255726**|1.2	11.3|277.2
1000_5_5	265202|14.2	5.548|60.6	265213|13.2	29.861|754.9	265219|18.1	0.384|63.1	**265223**|7.9	1.845|316
5000_5_1	1311797|29	526.492|66	1311797|17.6	2194.247|282.27	1311849|11	42.019|69	**1311853**|5	158.319|269.3
5000_5_2	1311690|22	534.489|53.53	1311693|14.3	2620.31|271.97	1311734|10	14.669|60.8	**1311739**|7	78.675|291.7
5000_5_3	1287770|18	500.59|60.9	1287779|17.43	2245.453|281.83	1287816|16	89.544|73.4	**1287820**|9	302.204|259.2
5000_5_4	1298162|23	823.385|69.3	1298174|23.03	3268.562|284.3	**1298230**|7	117.392|51.5	**1298230**|1	553.283|242
5000_5_5	1315789|21	540.043|67.27	1315814|28.57	2258.925|292.57	1315854|13	52.204|59.8	**1315856**|7	243.346|287.7
10000_5_1	2607139|32	6226.748|65.83	2607144|19.7	28216.31|304.9	2607299|18	1611.487|54.9	**2607304**|8	7463.16|250.6
10000_5_2	2584135|28	5753.947|58.93	2584145|24.63	25850.299|272.03	2584195|4	3716.762|48.9	**2584195**|0	16264.166|220.9
10000_5_3	2587616|0	47.905|0	2587616|0	47.328|0	2587726|12	1619.371|71.4	**2587728**|2	6238.245|292.7
10000_5_4	2601523|0	64.643|0	2601523|0	66.757|0	2601529|5	137.361|5	**2601535**|9	1560.024|92.7

continued on following page

Table 2. Continued

<table>
<tr><th>n_m_s</th><th>QIEA-MKP(20)</th><th></th><th>QIEA-MKP(100)</th><th></th><th>GQIEA-MKP(20)</th><th></th><th>GQIEA_MKP(100)</th><th></th></tr>
<tr><th></th><th>Best |Best-Avg</th><th>T(sec)|FES</th><th>Best |Best-Avg</th><th>T(sec)|FES</th><th>Best |Best-Avg</th><th>T(sec)|FES</th><th>Best |Best-Avg</th><th>T(sec)|FES</th></tr>
<tr><td>**10000_5_5**</td><td>2619168|28</td><td>5726.428|63.03</td><td>2619208|46.27</td><td>26414.639|278.4</td><td>2619335|5</td><td>2139.161|56.7</td><td>**2619338**|4</td><td>10443.43|270.8</td></tr>
<tr><td>**1000_10_1**</td><td>258363|15.6</td><td>14.848|54.9</td><td>**258366**|8.17</td><td>80.653|302.93</td><td>258297|55.4</td><td>0.433|76.3</td><td>258292|21.4</td><td>1.782|331.1</td></tr>
<tr><td>**1000_10_2**</td><td>260473|8.5</td><td>15.916|61.7</td><td>**260475**|5.5</td><td>72.851|288</td><td>260280|275.8</td><td>0.044|4</td><td>260455|217.7</td><td>0.193|40</td></tr>
<tr><td>**1000_10_3**</td><td>258921|17.8</td><td>15.08|55.97</td><td>**258922**|10.97</td><td>69.641|262</td><td>258904|49.5</td><td>0.488|81.1</td><td>258914|17.7</td><td>1.532|295.2</td></tr>
<tr><td>**1000_10_4**</td><td>255703|31.6</td><td>16.588|59.93</td><td>**255696**|15.27</td><td>77.615|285.3</td><td>255564|30.5</td><td>0.224|46.8</td><td>255686|83.1</td><td>1.198|270.8</td></tr>
<tr><td>**1000_10_5**</td><td>265175|30.2</td><td>15.081|649.47</td><td>**265184**|22.53</td><td>61.724|1835.87</td><td>265023|125.1</td><td>0.021|2</td><td>265132|69.4</td><td>0.033|3</td></tr>
<tr><td>**5000_10_1**</td><td>1311808|28</td><td>517.08|55.7</td><td>**1311808**|14.53</td><td>2865.905|312.9</td><td>1311739|57</td><td>12.586|63.8</td><td>1311738|33</td><td>34.544|316.2</td></tr>
<tr><td>**5000_10_2**</td><td>1311656|26</td><td>710.89|65.6</td><td>1311666|21.7</td><td>2947.104|276.6</td><td>**1311703**|46</td><td>13.563|42.9</td><td>1311701|25</td><td>32.528|232.7</td></tr>
<tr><td>**5000_10_3**</td><td>1287772|25</td><td>582.302|58.5</td><td>1287777|19.13</td><td>2848.431|292.3</td><td>**1287800**|122</td><td>4.376|14.4</td><td>1287798|45</td><td>25.507|179.6</td></tr>
<tr><td>**5000_10_4**</td><td>1298148|21</td><td>780.136|59.17</td><td>1298160|19.13</td><td>3590.579|274.13</td><td>1298198|55</td><td>13.906|55.4</td><td>**1298206**|23</td><td>90.309|349.5</td></tr>
<tr><td>**5000_10_5**</td><td>1303954|24</td><td>353.44|330.83</td><td>1303967|24.3</td><td>1676.855|1012.77</td><td>1303964|59</td><td>0.323|2</td><td>**1303996**|62</td><td>0.38|3</td></tr>
<tr><td>**10000_10_1**</td><td>2607178|0</td><td>125.391|0</td><td>2607178|0</td><td>127.033|0</td><td>2607293|13</td><td>794.163|62.9</td><td>**2607305**|11</td><td>3916.469|328.5</td></tr>
<tr><td>**10000_10_2**</td><td>2584132|0</td><td>85.966|0</td><td>2584132|0</td><td>87.271|0</td><td>2584176|35</td><td>110.65|14.8</td><td>**2584200**|48</td><td>291.33|111.5</td></tr>
<tr><td>**10000_10_3**</td><td>2587616|0</td><td>51.587|0</td><td>2587616|0</td><td>52.461|0</td><td>2587712|17</td><td>275.129|49.9</td><td>**2587716**|7</td><td>1505.956|294.8</td></tr>
<tr><td>**10000_10_4**</td><td>2601396|38</td><td>6129.751|66.4</td><td>2601398|21.63</td><td>27199.691|290.7</td><td>2601507|20</td><td>190.08|68.1</td><td>**2601513**|9</td><td>916.235|378.1</td></tr>
<tr><td>**10000_10_5**</td><td>2619260|32</td><td>5851.741|65.33</td><td>2619265|18.97</td><td>24789.69|273.9</td><td>2619326|18</td><td>451.893|71.8</td><td>**2619327**|7</td><td>1724.228|305.9</td></tr>
</table>

Table 3. Comparison of results obtained for strongly correlated instances having similar capacities using QIEA-MKP and GQIEA-MKP with population sizes of 20 and 100. The values obtained using optimal values and heuristic are shown as reference. Best value and difference between best and average is shown for QIEAs. Time to reach best solution in seconds and Function Evaluations (FES) are given.

<table>
<tr><th>n_m_s</th><th>QIEA-MKP(20)</th><th></th><th>QIEA-MKP(100)</th><th></th><th>GQIEA-MKP(20)</th><th></th><th>GQIEA_MKP(100)</th><th></th></tr>
<tr><th></th><th>Best (Best-Avg)</th><th>Time sec (FES)</th><th>Best (Best-Avg)</th><th>Time sec (FES)</th><th>Best (Best-Avg)</th><th>Time sec (FES)</th><th>Best (Best-Avg)</th><th>Time sec (FES)</th></tr>
<tr><td>1000_2_1</td><td>258384|0.93</td><td>0.499|3.3</td><td>258384|0.87</td><td>5.248|42.6</td><td>258384|0</td><td>1.669|25.2</td><td>258384|0</td><td>7.131|105</td></tr>
<tr><td>1000_2_2</td><td>260485|11</td><td>5.303|45</td><td>260485|8.4</td><td>22.298|184.77</td><td>260485|0</td><td>1.646|25.1</td><td>260485|0</td><td>7.253|106.2</td></tr>
<tr><td>1000_2_3</td><td>258940|0</td><td>0.087|0</td><td>258942|1.8</td><td>4.502|37.87</td><td>258942|0</td><td>0.17|2</td><td>258942|0</td><td>0.173|2</td></tr>
<tr><td>1000_2_4</td><td>255716|6.67</td><td>6.009|51.33</td><td>255726|10.3</td><td>30.502|256.83</td><td>255726|0</td><td>0.186|4.3</td><td>255726|0</td><td>0.19|4.3</td></tr>
<tr><td>1000_2_5</td><td>265224|9.1</td><td>6.311|52.73</td><td>265225|8.7</td><td>22.696|185.63</td><td>265225|0</td><td>3.083|51.6</td><td>265225|0</td><td>6.935|140.6</td></tr>
<tr><td>5000_2_1</td><td>1311764|10.3</td><td>236.73|23.63</td><td>1311764|8.13</td><td>1679.777|163</td><td>1311854|5.2</td><td>284.46|599.5</td><td>1311854|0</td><td>1782.758|262.3</td></tr>
<tr><td>5000_2_2</td><td>1311703|0</td><td>1.125|0</td><td>1311703|0</td><td>1.13|0</td><td>1311739|0</td><td>245.139|28.9</td><td>1311739|0</td><td>910.884|109.9</td></tr>
<tr><td>5000_2_3</td><td>1287821|0</td><td>7.317|0</td><td>1287821|0</td><td>7.333|0</td><td>1287821|0</td><td>6.361|0</td><td>1287821|0</td><td>6.376|0</td></tr>
<tr><td>5000_2_4</td><td>1298230|0</td><td>8.263|0</td><td>1298230|0</td><td>8.28|0</td><td>1298230|0</td><td>7.307|0</td><td>1298230|0</td><td>7.324|0</td></tr>
<tr><td>5000_2_5</td><td>1315856|0</td><td>1.035|0</td><td>1315856|0</td><td>1.039|0</td><td>1315856|0</td><td>0.05|0</td><td>1315856|0</td><td>0.06|0</td></tr>
<tr><td>10000_2_1</td><td>2607281|0</td><td>4.65|0</td><td>2607281|0</td><td>4.655|0</td><td>2607295|6.8</td><td>3465.05|52.6</td><td>2607295|2</td><td>16114.388|247.2</td></tr>
<tr><td>10000_2_2</td><td>2584205|0</td><td>4.166|0</td><td>2584205|0</td><td>4.189|0</td><td>2584205|0</td><td>0.187|0</td><td>2584205|0</td><td>0.212|0</td></tr>
<tr><td>10000_2_3</td><td>2587637|0</td><td>9.509|0</td><td>2587637|0</td><td>9.672|0</td><td>2587728|1</td><td>2140.474|38</td><td>2587728|0</td><td>6873.695|115.8</td></tr>
<tr><td>10000_2_4</td><td>2601537|0</td><td>4.163|0</td><td>2601537|0</td><td>4.226|0</td><td>2601537|0</td><td>0.189|0</td><td>2601537|0</td><td>0.212|0</td></tr>
<tr><td>10000_2_5</td><td>2619338|0</td><td>4.156|0</td><td>2619338|0</td><td>4.27|0</td><td>2619338|0</td><td>0.187|0</td><td>2619338|0</td><td>0.211|0</td></tr>
<tr><td>1000_5_1</td><td>258384|4.3</td><td>8.451|48.23</td><td>258384|0.03</td><td>35.997|203.63</td><td>258384|0</td><td>2.213|28.3</td><td>258384|0</td><td>8.313|107.6</td></tr>
<tr><td>1000_5_2</td><td>260485|4.4</td><td>8.98|50.67</td><td>260485|0.2</td><td>45.882|257.8</td><td>260485|0</td><td>3.36|40.3</td><td>260485|0</td><td>10.122|119.4</td></tr>
<tr><td>1000_5_3</td><td>258942|5.1</td><td>9.157|52.63</td><td>258942|0.9</td><td>49.569|283.9</td><td>258942|1.5</td><td>4.218|59.3</td><td>258942|0</td><td>15.74|219.6</td></tr>
<tr><td>1000_5_4</td><td>255725|9.2</td><td>9.397|52.23</td><td>255726|9.67</td><td>26.327|145.1</td><td>255726|5.2</td><td>2.204|34.7</td><td>255726|0</td><td>13.131|201.7</td></tr>
<tr><td>1000_5_5</td><td>265225|8.1</td><td>7.558|43</td><td>265225|1.77</td><td>46.225|263.63</td><td>265225|0</td><td>2.512|33.9</td><td>265225|0</td><td>9.085|121.4</td></tr>
<tr><td>5000_5_1</td><td>1311814|11.07</td><td>669.911|53.77</td><td>1311824|10.57</td><td>2912.924|228.43</td><td>1311854|7</td><td>104.071|12</td><td>1311854|1.6</td><td>1242.516|159.4</td></tr>
<tr><td>5000_5_2</td><td>1311729|0</td><td>10.666|0</td><td>1311729|0</td><td>10.699|0</td><td>1311739|0.9</td><td>416.223|44.7</td><td>1311739|0</td><td>1048.534|115.2</td></tr>
<tr><td>5000_5_3</td><td>1287811|0</td><td>14.775|0</td><td>1287811|0</td><td>14.81|0</td><td>1287821|4.1</td><td>281.316|28.7</td><td>1287821|0.1</td><td>1776.958|176.5</td></tr>
<tr><td>5000_5_4</td><td>1298190|7.3</td><td>647.247|52.23</td><td>1298200|8.67</td><td>2975|236.83</td><td>1298230|2</td><td>292.907|35.7</td><td>1298230|0</td><td>659.457|74.7</td></tr>
<tr><td>5000_5_5</td><td>1315826|13.7</td><td>671.82|54.57</td><td>1315836|15.57</td><td>2960.667|236.7</td><td>1315846|4.9</td><td>384.526|44.5</td><td>1315846|0</td><td>1559.462|181.8</td></tr>
<tr><td>10000_5_1</td><td>2607295|0</td><td>88.168|0</td><td>2607295|0</td><td>88.869|0</td><td>2607305|0</td><td>161.407|1.6</td><td>2607305|0</td><td>161.607|1.6</td></tr>
<tr><td>10000_5_2</td><td>2584194|0</td><td>40.256|0</td><td>2584194|0</td><td>41.623|0</td><td>2584205|6.3</td><td>4038.215|489.2</td><td>2584205|2.3</td><td>14592.94|200.3</td></tr>
<tr><td>10000_5_3</td><td>2587724|0</td><td>64.109|0</td><td>2587724|0</td><td>65.548|0</td><td>2587728|0</td><td>984.901|12</td><td>2587728|0</td><td>1045.213|13.4</td></tr>
<tr><td>10000_5_4</td><td>2601448|7.1</td><td>1564.415|17.1</td><td>2601457|13.07</td><td>14911.223|159.73</td><td>2601527|0</td><td>697.854|10.2</td><td>2601527|0</td><td>803.027|11.2</td></tr>
</table>

continued on following page

Table 3. Continued

<table>
<tr><th>n_m_s</th><th>QIEA-MKP(20)</th><th></th><th>QIEA-MKP(100)</th><th></th><th>GQIEA-MKP(20)</th><th></th><th>GQIEA_MKP(100)</th><th></th></tr>
<tr><th></th><th>Best (Best-Avg)</th><th>Time sec (FES)</th><th>Best (Best-Avg)</th><th>Time sec (FES)</th><th>Best (Best-Avg)</th><th>Time sec (FES)</th><th>Best (Best-Avg)</th><th>Time sec (FES)</th></tr>
<tr><td>**10000_5_5**</td><td>2619328|0</td><td>52.611|0</td><td>2619328|0</td><td>54.181|0</td><td>**2619338**|3</td><td>3126.132|45.6</td><td>**2619338**|0</td><td>11391.508|163.8</td></tr>
<tr><td>**1000_10_1**</td><td>258382|8.37</td><td>15.043|53.1</td><td>**258384**|8.2</td><td>54.273|192.77</td><td>**258384**|1.7</td><td>4.58|57.7</td><td>**258384**|0.1</td><td>16.624|205.9</td></tr>
<tr><td>**1000_10_2**</td><td>260485|10.97</td><td>17.905|63.3</td><td>260482|5.97</td><td>75.512|267.67</td><td>**260485**|1</td><td>3.797|65.1</td><td>**260485**|0</td><td>14.959|241.7</td></tr>
<tr><td>**1000_10_3**</td><td>258932|3.2</td><td>15.558|55.13</td><td>258932|0.03</td><td>60.475|215.3</td><td>**258942**|2.3</td><td>4.4|62.4</td><td>**258942**|0.1</td><td>16.231|217.7</td></tr>
<tr><td>**1000_10_4**</td><td>255716|7.93</td><td>15.818|55.3</td><td>255716|5.37</td><td>72.463|255.1</td><td>**255723**|6.7</td><td>3.602|69.5</td><td>255716|0</td><td>12.919|237.7</td></tr>
<tr><td>**1000_10_5**</td><td>265224|10</td><td>14.499|51.67</td><td>265219|3.93</td><td>57.067|204.53</td><td>**265225**|3.3</td><td>4.519|173.8</td><td>**265225**|0.5</td><td>16.537|229.9</td></tr>
<tr><td>**5000_10_1**</td><td>1311814|14.7</td><td>890.522|56.27</td><td>1311824|17.7</td><td>3571.798|229.3</td><td>1311844|0</td><td>271.441|34.2</td><td>**1311854**|6</td><td>2078.941|1609.3</td></tr>
<tr><td>**5000_10_2**</td><td>1311710|0</td><td>14.542|0</td><td>1311710|0</td><td>14.434|0</td><td>**1311739**|0.3</td><td>359.14|50.6</td><td>**1311739**|0</td><td>1133.592|156.9</td></tr>
<tr><td>**5000_10_3**</td><td>1287791|21</td><td>897.482|56.87</td><td>1287801|21.37</td><td>3479.035|221.83</td><td>**1287821**|0</td><td>339.068|47.4</td><td>**1287821**|0</td><td>825.37|115.5</td></tr>
<tr><td>**5000_10_4**</td><td>1298190|11.7</td><td>1027.679|65.63</td><td>1298200|15.03</td><td>3856.234|250.7</td><td>**1298230**|2.3</td><td>285.63|47.5</td><td>**1298230**|0</td><td>1229.778|213.1</td></tr>
<tr><td>**5000_10_5**</td><td>1315816|15.7</td><td>906.021|56.87</td><td>1315816|8.67</td><td>4033.564|258.6</td><td>**1315856**|9</td><td>302.421|35.6</td><td>**1315856**|8</td><td>1544.415|673.9</td></tr>
<tr><td>**10000_10_1**</td><td>2607256|0</td><td>71.154|0</td><td>2607256|0</td><td>71.44|0</td><td>**2607305**|11.5</td><td>4001.634|188.4</td><td>**2607305**|6</td><td>12893.323|399.1</td></tr>
<tr><td>**10000_10_2**</td><td>2584125|20.7</td><td>6884.654|63.27</td><td>2584125|11.33</td><td>29960.623|273.13</td><td>**2584205**|4.8</td><td>3588.676|171</td><td>**2584205**|2</td><td>16402.387|522.2</td></tr>
<tr><td>**10000_10_3**</td><td>2587658|27.3</td><td>6785.086|61.53</td><td>2587658|21.43</td><td>28702.974|263.77</td><td>2587718|0</td><td>2886.294|53.9</td><td>**2587724**|4.4</td><td>9550.744|173.2</td></tr>
<tr><td>**10000_10_4**</td><td>2601507|0</td><td>108.156|0</td><td>2601507|0</td><td>107.101|0</td><td>**2601527**|0</td><td>2082.553|33.1</td><td>**2601527**|0</td><td>7356.117|109.8</td></tr>
<tr><td>**10000_10_5**</td><td>2619318|0</td><td>99.04|0</td><td>2619318|0</td><td>98.674|0</td><td>**2619338**|8.3</td><td>2217.364|142.83</td><td>**2619338**|3.3</td><td>13319.07|693.9</td></tr>
</table>

Table 4. Comparison of results obtained for uncorrelated instances having dissimilar capacities using QIEA-MKP and GQIEA-MKP with population sizes of 20 and 100. The values obtained using optimal values and heuristic are shown as reference. Best value and difference between best and average is shown for QIEAs. Time to reach best solution in seconds and Function Evaluations (FES) are given.

<table>
<tr><th>n_m_s</th><th>QIEA-MKP(20)</th><th></th><th>QIEA-MKP(100)</th><th></th><th>GQIEA-MKP(20)</th><th></th><th>GQIEA_MKP(100)</th><th></th></tr>
<tr><th></th><th>Best (Best-Avg)</th><th>Time sec (FES)</th><th>Best (Best-Avg)</th><th>Time sec (FES)</th><th>Best (Best-Avg)</th><th>Time sec (FES)</th><th>Best (Best-Avg)</th><th>Time sec (FES)</th></tr>
<tr><td>1000_2_1</td><td>391656|2.8</td><td>0.28|2.43</td><td>391675|17.1</td><td>8.872|158.6</td><td>391675|9.1</td><td>1.262|1465.3</td><td>391675|2.7</td><td>6.632|9226.2</td></tr>
<tr><td>1000_2_2</td><td>409022|9.4</td><td>3.865|45.4</td><td>409023|1.5</td><td>8.939|171.6</td><td>409017|3.6</td><td>0.346|397.8</td><td>409017|2.4</td><td>0.635|98</td></tr>
<tr><td>1000_2_3</td><td>406439|9.8</td><td>1.358|17.4</td><td>406439|5.9</td><td>8.007|184.57</td><td>406438|7.8</td><td>0.424|906.3</td><td>406439|3.6</td><td>4.081|9130.9</td></tr>
<tr><td>1000_2_4</td><td>408580|11.3</td><td>2.829|33.3</td><td>408580|2.1</td><td>11.918|257.73</td><td>408580|0</td><td>0.241|58.5</td><td>408580|0</td><td>0.436|79</td></tr>
<tr><td>1000_2_5</td><td>415855|6.4</td><td>1.78|21.23</td><td>415856|4.7</td><td>7.733|162</td><td>415853|6.3</td><td>0.169|555.6</td><td>415860|9.9</td><td>0.703|838.2</td></tr>
<tr><td>5000_2_1</td><td>2003812|5</td><td>28.149|11.87</td><td>2003821|9</td><td>310.976|206.4</td><td>2003826|2</td><td>15.765|158.2</td><td>2003827|3</td><td>105.889|3275.3</td></tr>
<tr><td>5000_2_2</td><td>2045737|0</td><td>1.744|0</td><td>2045737|0</td><td>1.211|0</td><td>2045746|4</td><td>4.416|233.5</td><td>2045746|0</td><td>5.934|95.1</td></tr>
<tr><td>5000_2_3</td><td>2019633|0</td><td>1.279|0</td><td>2019640|6</td><td>110.026|72.47</td><td>2019640|1</td><td>6.905|34.9</td><td>2019647|6</td><td>43.56|2267.2</td></tr>
<tr><td>5000_2_4</td><td>2034400|0</td><td>0.983|0</td><td>2034400|0</td><td>1.163|0</td><td>2034400|0</td><td>0.067|0</td><td>2034400|0</td><td>0.079|0</td></tr>
<tr><td>5000_2_5</td><td>2025987|0</td><td>1.176|0</td><td>2025991|4</td><td>52.449|34.83</td><td>2026013|6</td><td>41.998|1758.7</td><td>2026013|3</td><td>240.168|12345.4</td></tr>
<tr><td>10000_2_1</td><td>4040301|0</td><td>3.96|0</td><td>4040302|1</td><td>30.694|3.73</td><td>4040305|3</td><td>131.698|1520.2</td><td>4040305|3</td><td>415.46|4327.7</td></tr>
<tr><td>10000_2_2</td><td>4067224|0</td><td>4.778|0</td><td>4067224|0</td><td>6.686|0</td><td>4067229|3</td><td>138.038|1745.3</td><td>4067230|1</td><td>458.547|5723.9</td></tr>
<tr><td>10000_2_3</td><td>4037911|0</td><td>4.219|0</td><td>4037911|0</td><td>4.252|0</td><td>4037915|0</td><td>81.585|1151.8</td><td>4037915|0</td><td>144.655|1010.6</td></tr>
<tr><td>10000_2_4</td><td>4054704|0</td><td>5.036|0</td><td>4054704|0</td><td>4.112|0</td><td>4054706|2</td><td>7.007|14.3</td><td>4054706|0</td><td>754.763|6359.3</td></tr>
<tr><td>10000_2_5</td><td>4029933|0</td><td>6.34|0</td><td>4029933|0</td><td>4.202|0</td><td>4029936|2</td><td>112.168|1868.3</td><td>4029935|2</td><td>135.278|2398.1</td></tr>
<tr><td>1000_5_1</td><td>391649|18.9</td><td>3.207|18.57</td><td>391659|17.2</td><td>25.608|278.7</td><td>391661|20.5</td><td>0.326|70</td><td>391664|8.5</td><td>1.594|336.8</td></tr>
<tr><td>1000_5_2</td><td>409016|19.1</td><td>3.073|17.43</td><td>409011|9.7</td><td>13.67|146.77</td><td>409013|15.6</td><td>0.037|7.5</td><td>409013|13</td><td>0.773|193.9</td></tr>
<tr><td>1000_5_3</td><td>406425|19.6</td><td>8.74|53.77</td><td>406425|8.6</td><td>19.424|218.87</td><td>406420|31.8</td><td>0.204|136.7</td><td>406421|20.1</td><td>1.584|1336.8</td></tr>
<tr><td>1000_5_4</td><td>408553|1</td><td>0.307|0.87</td><td>408580|26.1</td><td>2.762|28.93</td><td>408564|7.2</td><td>0.051|12.8</td><td>408564|1.2</td><td>0.944|231.1</td></tr>
<tr><td>1000_5_5</td><td>415835|0</td><td>0.155|0</td><td>415843|6.8</td><td>5.453|59.93</td><td>415835|0</td><td>0.005|0</td><td>415837|1.8</td><td>0.086|18.5</td></tr>
<tr><td>5000_5_1</td><td>2003805|0</td><td>2.913|0</td><td>2003805|0</td><td>2.275|0</td><td>2003821|5</td><td>7.304|52.3</td><td>2003825|6</td><td>50.709|341.1</td></tr>
<tr><td>5000_5_2</td><td>2045725|0</td><td>2.32|0</td><td>2045725|0</td><td>2.313|0</td><td>2045737|7</td><td>3.978|31.5</td><td>2045737|3</td><td>30.356|244.3</td></tr>
<tr><td>5000_5_3</td><td>2019630|0</td><td>2.178|0</td><td>2019630|0</td><td>2.17|0</td><td>2019633|3</td><td>1.229|6.2</td><td>2019630|0</td><td>0.192|0</td></tr>
<tr><td>5000_5_4</td><td>2034387|0</td><td>2.142|0</td><td>2034388|1</td><td>25.208|10.63</td><td>2034387|0</td><td>0.114|0</td><td>2034387|0</td><td>0.144|0</td></tr>
<tr><td>5000_5_5</td><td>2025977|14</td><td>30.312|10.37</td><td>2025979|14</td><td>223.411|82.87</td><td>2026001|11</td><td>13.04|49.9</td><td>2026000|6</td><td>86.026|317.9</td></tr>
<tr><td>10000_5_1</td><td>4040263|0</td><td>10.147|0</td><td>4040274|10</td><td>520.58|43.07</td><td>4040301|8</td><td>68.847|40</td><td>4040302|2</td><td>566.873|323.6</td></tr>
<tr><td>10000_5_2</td><td>4067211|0</td><td>10.542|0</td><td>4067211|0</td><td>9.528|0</td><td>4067221|5</td><td>50.505|40.9</td><td>4067224|4</td><td>287.364|241.6</td></tr>
<tr><td>10000_5_3</td><td>4037886|0</td><td>8.563|0</td><td>4037896|9</td><td>355.441|40.6</td><td>4037903|3</td><td>59.194|1954.2</td><td>4037907|3</td><td>185.641|6010.6</td></tr>
<tr><td>10000_5_4</td><td>4054695|0</td><td>8.659|0</td><td>4054695|0</td><td>8.656|0</td><td>4054695|0</td><td>0.537|0</td><td>4054703|7</td><td>56.08|2199.4</td></tr>
</table>

continued on following page

Table 4. Continued

n_m_s	QIEA-MKP(20)		QIEA-MKP(100)		GQIEA-MKP(20)		GQIEA_MKP(100)	
	Best (Best-Avg)	Time sec (FES)	Best (Best-Avg)	Time sec (FES)	Best (Best-Avg)	Time sec (FES)	Best (Best-Avg)	Time sec (FES)
10000_5_5	4029927|0	10.296|0	4029927|0	9.252|0	4029931|4	8.483|5.8	**4029933**|3	141.389|111.6
1000_10_1	409556|24	15.534|52.83	409563|15.7	52.866|288.37	409452|0	0.012|0	**409484**|24.6	0.257|71.6
1000_10_2	401691|41.3	13.203|45.17	**401692**|21.9	41.332|226.4	401641|22.4	0.083|17.1	401636|13.4	1.025|232.3
1000_10_3	406408|20.7	10.877|38.37	**406410**|11.9	29.028|164.37	406360|0	0.01|0	406360|0	0.023|0
1000_10_4	**403617**|0	0.274|0	**403617**|0	0.175|0	**403617**|0	0.01|0	**403617**|0	0.022|0
1000_10_5	**406064**|24.1	13.141|46.77	406056|6	38.72|220.23	405983|0	0.007|0	405990|6.3	0.019|0.2
5000_10_1	2024879|0	7.296|0	2024881|2	39.348|7.4	2024879|0	0.392|0	**2024883**|4	2.389|22
5000_10_2	2045697|1	8.592|0.67	**2045713**|16	129.461|26.8	2045696|0	0.282|0	2045698|2	4.361|33.9
5000_10_3	2019607|0	4.43|0	**2019614**|6	197.319|42.4	2019607|0	0.255|0	2019607|0	0.317|0
5000_10_4	2034383|8	66.036|13.37	**2034387**|8	1097.098|249.23	2034373|0	0.138|0	2034373|0	0.201|0
5000_10_5	2025966|4	8.044|0.7	**2025985**|22	50.367|9.57	2025962|0	0.469|0	2025977|12	26.898|179.8
10000_10_1	4040253|0	19.938|0	4040254|1	19.968|0	4040266|6	36.842|44.6	**4040269**|4	227.858|279.8
10000_10_2	4067187|0	18.618|0	4067188|1	189.681|8.8	4067205|12	24.698|52	**4067211**|11	142.384|311.3
10000_10_3	**4061129**|0	17.78|0	**4061129**|0	17.386|0	**4061129**|0	0.832|0	**4061129**|0	1.1|0
10000_10_4	**4035592**|0	17.89|0	**4035592**|0	17.495|0	**4035592**|0	0.98|0	**4035592**|0	1.264|0
10000_10_5	**4029914**|0	18.303|0	**4029914**|0	17.844|0	**4029914**|0	1.382|0	**4029914**|0	1.651|0

Table 5. Comparison of results obtained for uncorrelated instances having similar capacities using QIEA-MKP and GQIEA-MKP with population sizes of 20 and 100. The values obtained using optimal values and heuristic are shown as reference. Best value and difference between best and average is shown for QIEAs. Time to reach best solution in seconds and Function Evaluations (FES) are given.

<table>
<tr><th>n_m_s</th><th>QIEA-MKP(20)</th><th></th><th>QIEA-MKP(100)</th><th></th><th>GQIEA-MKP(20)</th><th></th><th>GQIEA_MKP(100)</th><th></th></tr>
<tr><th></th><th>Best (Best-Avg)</th><th>Time sec (FES)</th><th>Best (Best-Avg)</th><th>Time sec (FES)</th><th>Best (Best-Avg)</th><th>Time sec (FES)</th><th>Best (Best-Avg)</th><th>Time sec (FES)</th></tr>
<tr><td>**1000_2_1**</td><td>391661|7.2</td><td>0.476|8.8</td><td>391663|7.13</td><td>4.936|102.5</td><td>**391675**|6.7</td><td>1.207|1677</td><td>**391675**|1.3</td><td>5.494|9285.2</td></tr>
<tr><td>**1000_2_2**</td><td>409010|9.83</td><td>2.512|45.13</td><td>409022|11.3</td><td>12.98|246.3</td><td>409022|7.8</td><td>0.934|664.5</td><td>**409023**|3.4</td><td>3.769|1117.2</td></tr>
<tr><td>**1000_2_3**</td><td>406422|8.17</td><td>0.48|9</td><td>406424|9.37</td><td>4.399|93.03</td><td>**406439**|6.7</td><td>1.114|1904.7</td><td>**406439**|3.1</td><td>5.114|7475.4</td></tr>
<tr><td>**1000_2_4**</td><td>**408580**|14.8</td><td>0.229|3.83</td><td>**408580**|13.9</td><td>2.202|46.03</td><td>**408580**|0.9</td><td>0.962|1673.5</td><td>**408580**|0</td><td>3.324|3341</td></tr>
<tr><td>**1000_2_5**</td><td>415850|12.7</td><td>1.017|19.7</td><td>415847|5.73</td><td>9.352|195.17</td><td>415856|11.5</td><td>0.831|1524.3</td><td>**415860**|5</td><td>2.964|2907.3</td></tr>
<tr><td>**5000_2_1**</td><td>2003821|0</td><td>1.256|0</td><td>2003821|0</td><td>1.224|0</td><td>2003821|0</td><td>0.339|0</td><td>**2003825**|3.6</td><td>34.872|1799.2</td></tr>
<tr><td>**5000_2_2**</td><td>2045737|0</td><td>1.147|0</td><td>2045741|3.87</td><td>13.636|9.23</td><td>**2045746**|3.3</td><td>35.44|2231.8</td><td>**2045746**|0.6</td><td>178.761|11601.7</td></tr>
<tr><td>**5000_2_3**</td><td>2019630|0</td><td>1.038|0</td><td>2019630|0</td><td>1.011|0</td><td>2019642|1.9</td><td>41.626|1549.9</td><td>**2019647**|4.6</td><td>195.269|6047.4</td></tr>
<tr><td>**5000_2_4**</td><td>2034387|0</td><td>1.116|0</td><td>2034387|0</td><td>1.081|0</td><td>**2034402**|3.7</td><td>36.691|2276.6</td><td>2034400|0.3</td><td>180.495|11105.4</td></tr>
<tr><td>**5000_2_5**</td><td>2025989|1.93</td><td>4.853|2.6</td><td>2025995|7.37</td><td>38.198|26.07</td><td>**2026014**|5.6</td><td>51.029|2533.5</td><td>**2026014**|3.1</td><td>278.746|18965.5</td></tr>
<tr><td>**10000_2_1**</td><td>4040301|0</td><td>4.026|0</td><td>4040301|0</td><td>3.913|0</td><td>4040305|2.8</td><td>83.004|740.1</td><td>**4040309**|1.1</td><td>1411.442|16011.1</td></tr>
<tr><td>**10000_2_2**</td><td>4067221|0</td><td>4.8|0</td><td>4067221|0</td><td>4.659|0</td><td>**4067230**|1.8</td><td>193.09|2398.2</td><td>**4067230**|1.1</td><td>829.242|8990.8</td></tr>
<tr><td>**10000_2_3**</td><td>4037906|0</td><td>5.312|0</td><td>4037906|0</td><td>5.145|0</td><td>4037915|1.6</td><td>217.681|1763.6</td><td>**4037916**|0.9</td><td>1270.973|12146.7</td></tr>
<tr><td>**10000_2_4**</td><td>4054700|0</td><td>4.935|0</td><td>4054700|0</td><td>4.779|0</td><td>4054709|3.3</td><td>206.126|2381.7</td><td>**4054710**|1</td><td>994.818|14873.8</td></tr>
<tr><td>**10000_2_5**</td><td>4029915|0</td><td>4.315|0</td><td>4029915|0</td><td>4.203|0</td><td>**4029935**|2.2</td><td>267.215|2377.1</td><td>**4029935**|1.1</td><td>1106.812|6426</td></tr>
<tr><td>**1000_5_1**</td><td>391632|24.4</td><td>1.189|11.73</td><td>391639|25.77</td><td>18.315|199.37</td><td>391643|17.2</td><td>0.276|26.1</td><td>**391653**|15.5</td><td>1.321|129.6</td></tr>
<tr><td>**1000_5_2**</td><td>408978|33.33</td><td>5.554|54.8</td><td>408979|20.13</td><td>27.363|281.57</td><td>409000|51.5</td><td>1.096|1405.4</td><td>**408990**|25.8</td><td>7.391|7024.2</td></tr>
<tr><td>**1000_5_3**</td><td>406379|14.5</td><td>0.336|2.67</td><td>406389|20.4</td><td>13.903|152.2</td><td>406381|15.3</td><td>0.174|298.8</td><td>**406403**|27.7</td><td>4.983|9641.4</td></tr>
<tr><td>**1000_5_4**</td><td>408516|24.3</td><td>3.084|31.23</td><td>408533|21.83</td><td>28.01|299.67</td><td>408543|25.7</td><td>0.255|24.8</td><td>**408564**|25</td><td>2.643|2632.6</td></tr>
<tr><td>**1000_5_5**</td><td>415823|0</td><td>0.089|0</td><td>415824|0.97</td><td>0.427|3.7</td><td>415823|0</td><td>0.008|0</td><td>**415843**|16.7</td><td>0.014|2.9</td></tr>
<tr><td>**5000_5_1**</td><td>**2003805**|0</td><td>2.353|0</td><td>**2003805**|0</td><td>2.297|0</td><td>**2003805**|0</td><td>0.346|0</td><td>**2003805**|0</td><td>0.374|0</td></tr>
<tr><td>**5000_5_2**</td><td>2045714|0</td><td>2.194|0</td><td>2045714|0</td><td>2.141|0</td><td>2045735|16.4</td><td>7.893|187.5</td><td>**2045736**|11</td><td>67.929|170.9</td></tr>
<tr><td>**5000_5_3**</td><td>2019616|10</td><td>19.069|6.67</td><td>2019619|10.53</td><td>350.567|140.73</td><td>2019623|12.9</td><td>14.007|631</td><td>**2019625**|6.5</td><td>140.698|2108.7</td></tr>
</table>

continued on following page

Table 5. Continued

n_m_s	QIEA-MKP(20)		QIEA-MKP(100)		GQIEA-MKP(20)		GQIEA_MKP(100)	
	Best (Best-Avg)	Time sec (FES)	Best (Best-Avg)	Time sec (FES)	Best (Best-Avg)	Time sec (FES)	Best (Best-Avg)	Time sec (FES)
5000_5_4	2034366|0	2.294|0	2034368|1.87	16.308|5.77	2034372|5.2	7.701|280.1	2034387|17	134.339|4952.6
5000_5_5	2025970|7.2	21.876|7.17	2025978|14.63	140.097|52.1	2025984|16	16.235|30.6	2025983|9.3	117.958|220.3
10000_5_1	4040288|0	9.48|0	4040288|0	9.218|0	4040295|6.3	2.841|0.7	4040295|5.9	208.456|1267.3
10000_5_2	4067206|0	9.344|0	4067206|0	9.101|0	4067206|0	1.224|0	4067209|2.7	28.326|12.7
10000_5_3	4037911|0	9.483|0	4037911|0	9.22|0	4037911|0	1.356|0	4037911|0	1.416|0
10000_5_4	4054689|0	9.403|0	4054689|0	9.202|0	4054696|6.1	30.143|13.8	4054692|2.6	114.524|53
10000_5_5	4029920|0	10.115|0	4029920|0	9.946|0	4029922|1.8	1.919|0	4029920|0	1.977|0
1000_10_1	391526|0	0.176|0	391550|21.9	10.954|61.13	391596|50.7	0.31|30.1	391598|27.8	2.315|478.3
1000_10_2	408876|0	0.181|0	408897|20.1	2.163|10.67	408922|29.2	0.407|177.1	408935|26.5	3.019|279.3
1000_10_3	406270|13.53	0.769|3.17	406277|18.2	17.718|97.2	406351|62	1.111|133.3	406346|31	4.707|6120.6
1000_10_4	408441|0	0.181|0	408441|0	0.176|0	408481|22.3	0.198|19.6	408504|25.2	3.684|3892.3
1000_10_5	415733|26	0.179|0	415771|58.9	6|32.27	415816|60.8	0.655|622.3	415789|20.7	2.294|439.2
5000_10_1	2003772|0	4.597|0	2003772|0	4.446|0	2003772|0	0.424|0	2003787|12.3	45.888|94.8
5000_10_2	2045662|32.8	157.481|30.67	2045663|24.03	1367.027|282.53	2045693|16.9	15.94|42.5	2045701|12.9	91.645|234
5000_10_3	2019576|0	4.576|0	2019576|0	4.427|0	2019580|3.6	0.396|0	2019586|6.8	43.232|107.1
5000_10_4	2034333|0	4.536|0	2034333|0	4.386|0	2034367|22.2	21.002|930.7	2034373|22	76.408|238.8
5000_10_5	2025927|0	4.691|0	2025949|21.13	73.826|14.37	2025980|34.2	22.314|81.6	2025980|21.6	103.613|2638.8
10000_10_1	4040263|0	18.925|0	4040263|0	18.309|0	4040273|5.5	84.604|39.3	4040288|15.8	501.661|238.6
10000_10_2	4067179|0	19.213|0	4067179|0	18.58|0	4067183|3.6	8.189|3.5	4067192|10	212.005|1738.4
10000_10_3	4037853|0	18.932|0	4037855|1.9	284.767|13.07	4037890|14.1	100.294|50.8	4037902|13.3	517|268.3
10000_10_4	4054669|0	18.608|0	4054669|0	18.169|0	4054683|9.9	86.143|47.3	4054685|3.9	633.418|2627.2
10000_10_5	4029889|0	18.46|0	4029889|0	18.298|0	4029916|12.3	87.074|39.8	4029914|1.3	298.142|138.2

For uncorrelated instances having dissimilar capacities the results are as follows

- GQIEA_MKP(100) provides largest best value for 30 out of 45 instances.
- GQIEA_MKP(20) provides largest best value for 14 out of 45 instances.
- QIEA_MKP(100) provide largest best value in 19 instances
- while QIEA_MKP(20) provide only for 9 instances.
- However, Out of 45 instances
    - for 5 instances all four algorithms provide same best value,
    - for 1 instance QIEA_MKP(100), GQIEA_MKP(20) and GQIEA_MKP(100) provide same largest best value,
    - for 2 instance QIEA_MKP(20), QIEA_MKP(100) and GQIEA_MKP(100) provide same largest best value,
    - for 7 instances only GQIEA_MKP(20) and GQIEA_MKP(100) provide largest best value,
    - for 1 instance only QIEA_MKP(20) provide largest best value,
    - for 10 instance only QIEA_MKP(100) provide largest best value,
    - for 3 instances only GQIEA_MKP(20) provides largest best value
    - while for 16 instances only GQIEA_MKP(100) provides largest best value.
- Where all four types of instances provide same largest value, versions of GQIEAs are more consistent with better average values and significantly faster.
- The instances where both GQIEA_MKP(20) and GQIEA_MKP(100) provide same best value, for 22 instances GQIEA_MKP(100) provide better average value. But the time required by GQIEA_MKP(100) is typically more than GQIEA_MKP(20).

For uncorrelated instances having similar capacities the results are as follows

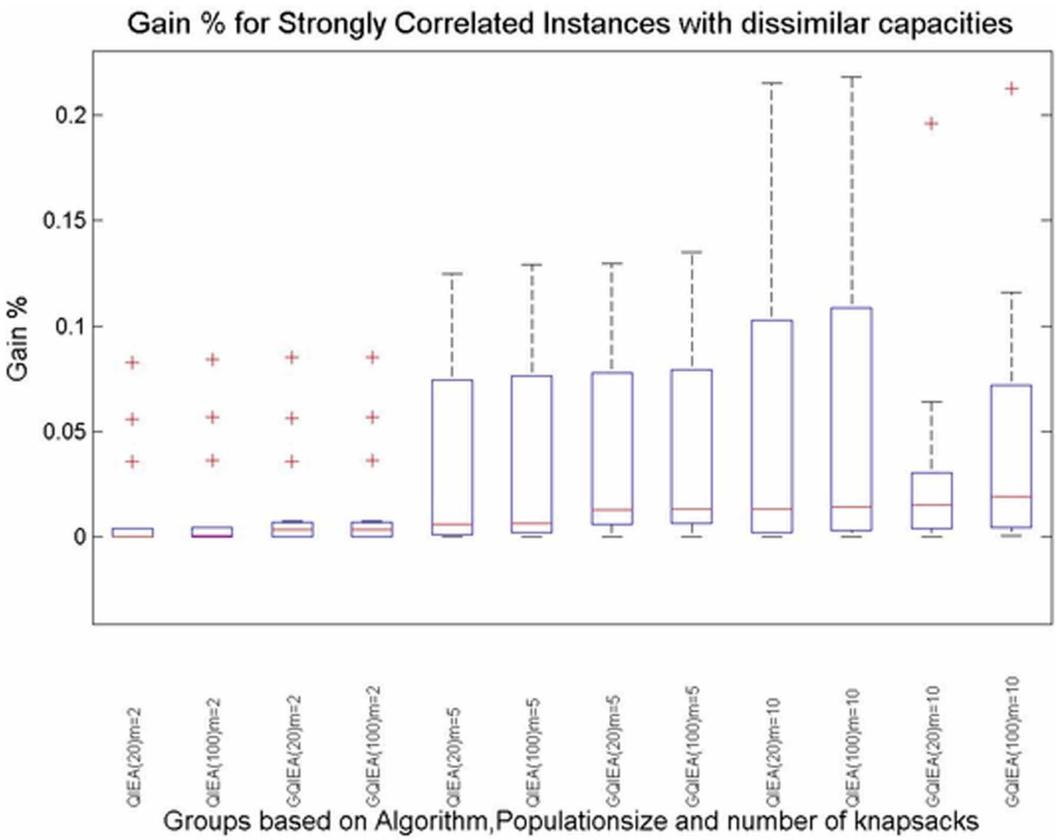
- GQIEA_MKP(100) provides largest best value for 38 out of 45 instances.
- GQIEA_MKP(20) provides largest best value for 18 out of 45 instances.
- QIEA_MKP(100) provide largest best value in 3 instances
- while QIEA_MKP(20) provide only for 3 instances.
- However, Out of 45 instances
    - for 3 instances all four algorithms provide same best value,
    - for 9 instances only GQIEA_MKP(20) and GQIEA_MKP(100) provide largest best value,
    - for 7 instance only QIEA_MKP(20) provide largest best value,
    - while for 26 instances only GQIEA_MKP(100) provides largest best value.
- Where all four types of instances provide same largest value, versions of GQIEAs are more consistent with better average values and significantly faster.
- The instances 12 instances where both GQIEA_MKP(20) and GQIEA_MKP(100) provide same best value, for 8 instances GQIEA_MKP(100) provide better average value. But the time required by GQIEA_MKP(100) is typically more than GQIEA_MKP(20).

Gain % in best solution and average solution obtained using applied approach over the values obtained from heuristic are calculated to elucidate the comparison. They are used to compare different algorithms using box plots in Figures 8 to 11 when instances grouped on the basis of same number of knapsacks and Figures 16 and 17 when instances are grouped on the basis of number of elements.

The following observations are made from the results.

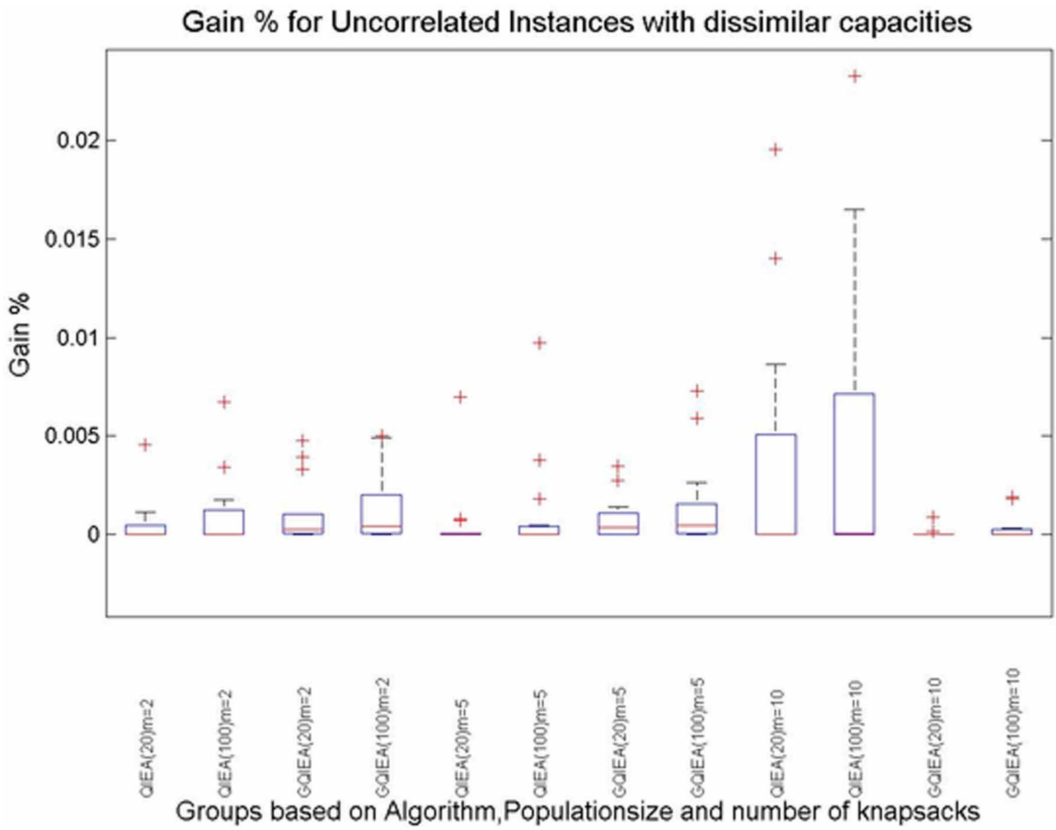
- Both forms of QIEAs viz. QIEA-MKP and GQIEA-MKP studied here to solve MKP with different number of variables and knapsacks shows considerable improvement in quality of solutions obtained as compared to the solutions provided by the popular heuristic, the MTHM.

Figure 8. Boxplots for gain % in observed profit values over the heuristic values using different strategies and population size to solve Strongly Correlated Instances of MKP with dissimilar capacities grouped by number of knapsacks



- From Figures 8 to 15 it is clear that, GQIEA-MKP have shown a significant improvement in quality of solutions and also in time taken to find good solutions as compared to QIEA-MKP for the instances having number of knapsacks up to 5. In some instances the time taken to reach the best solution by GQIEA-MKP is larger than observed in case of QIEA-MKP. It is because in these cases QIEA-MKP could bring no improvement in the quality of the initialized solutions, so best solution is the initialized solution and no time has been taken to reach the best solution. (Based on experiments conducted it is observed that even after providing time to execute to QIEA-MKP much more than GQIEA-MKP, quality of solution is not improved in these cases).
- In case of instances having higher number of knapsacks (as 10), the one with similar capacities are solved in better way by GQIEA-MKP on an average as compared to QIEA-MKP while in case of dissimilar capacities QIEA-MKP provides better solutions than GQIEA-MKP. This observation is made in Figures 8 to 13. In these figures, the group-wise averages are considered such that all instances having different sizes but same number of knapsacks are grouped together.
- To illustrate another perspective for instances having dissimilar capacities, separate grouping is formed. All instances having same size but different number of knapsacks are grouped together. The separate box plots in Figure 16 and 17 have been drawn where the instances have been grouped in this way. It is observed that irrespective of what is the number of knapsacks, GQIEA-MKP performs better than QIEA-MKP for larger sizes. It is clear from these that GQIEA-MKP performs better as compared to QIEA-MKP as size of the problem increases. Similar graphs

Figure 9. Boxplots for gain % in observed profit values over the heuristic values using different strategies and population size to solve Uncorrelated Instances of MKP with dissimilar capacities grouped by number of knapsacks



have not been shown for other classes of instances where also the performance of GQIEA-MKP improves over QIEA-MKP as size of problem increases.

5. CONCLUSION

The GQIEA-MKP has been designed based on generalized representation in QIEA to solve MKP effectively. Due to its effective representation for MKP, GQIEA-MKP has an edge over QIEA-MKP which uses the standard representation of Q-bits and operators. The proposed GQIEA-MKP is hybridized by influencing the initialization of generalized Q-bit individuals and the local search procedure based on a known effective heuristics for MKP with an objective to improve exploitation. Apart from it some techniques viz. mutation of solutions appearing to be close to local optima, and reinitializing Q-bit individuals found incapable to generate new solutions are also induced in order to improve the power to explore the search space. Hence, the balanced capability to exploit and explore the search space is established in proposed algorithm.

The hybridized versions of generalized QIEA (GQIEA-MKP) is studied in comparison to simple QIEA (QIEA-MKP) proposed earlier in (Patvardhan, Bansal, & Srivastav, Balanced quantum-inspired evolutionary algorithm for multiple knapsack problem, 2014) to solve Multiple Knapsack Problem (MKP). The GQIEA-MKP is shown to be better choice than QIEA-MKP to solve problems having non-binary integer solutions like MKP.

Figure 10. Boxplots for gain % in observed profit values over the heuristic values using different strategies and population size to solve Strongly correlated Instances of MKP with similar capacities grouped by number of knapsacks

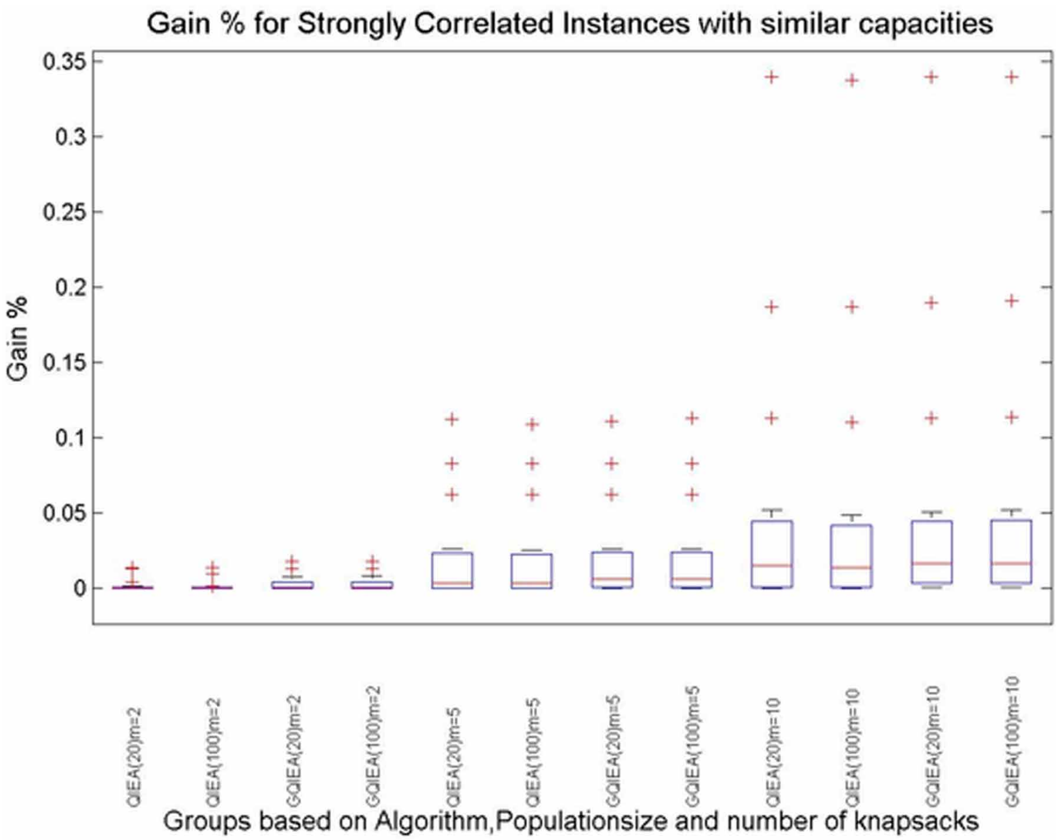


Figure 11. Boxplots for gain % in observed profit values over the heuristic values using different strategies and population size to solve Uncorrelated Instances of MKP with similar capacities grouped by number of knapsacks

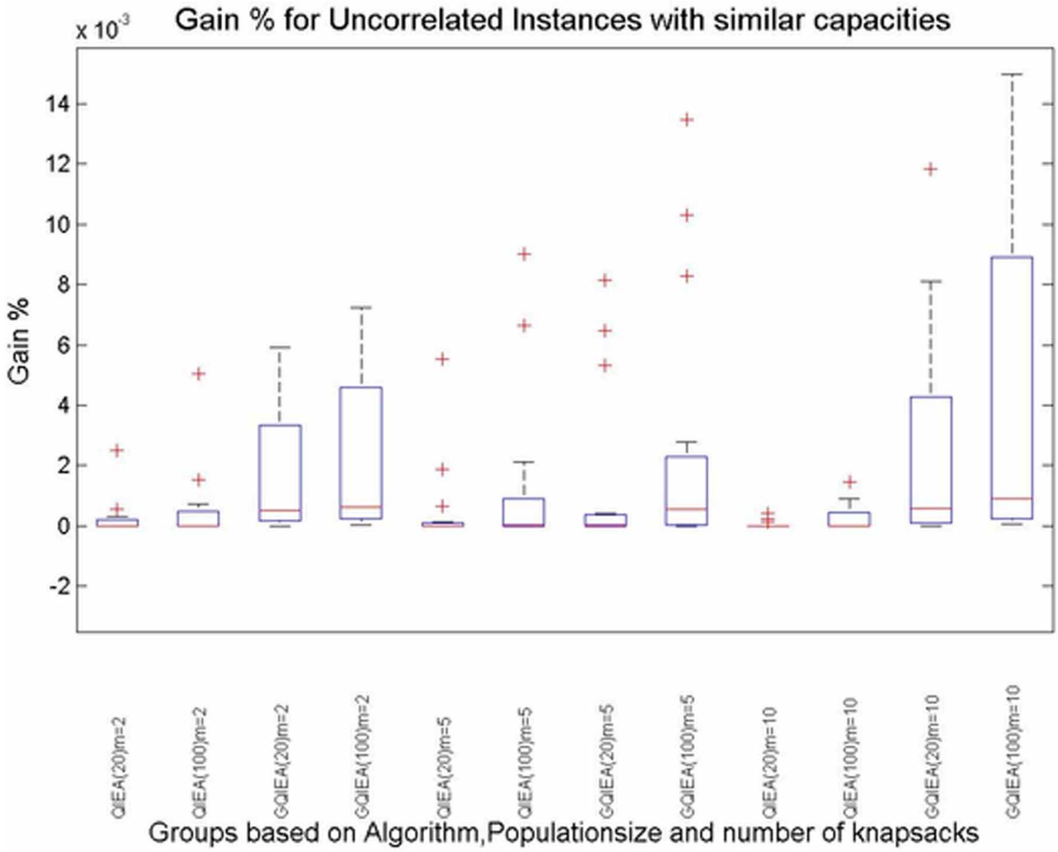


Figure 12. Boxplots for time taken using different strategies and population size to solve Strongly correlated Instances of MKP with dissimilar capacities grouped by number of knapsacks

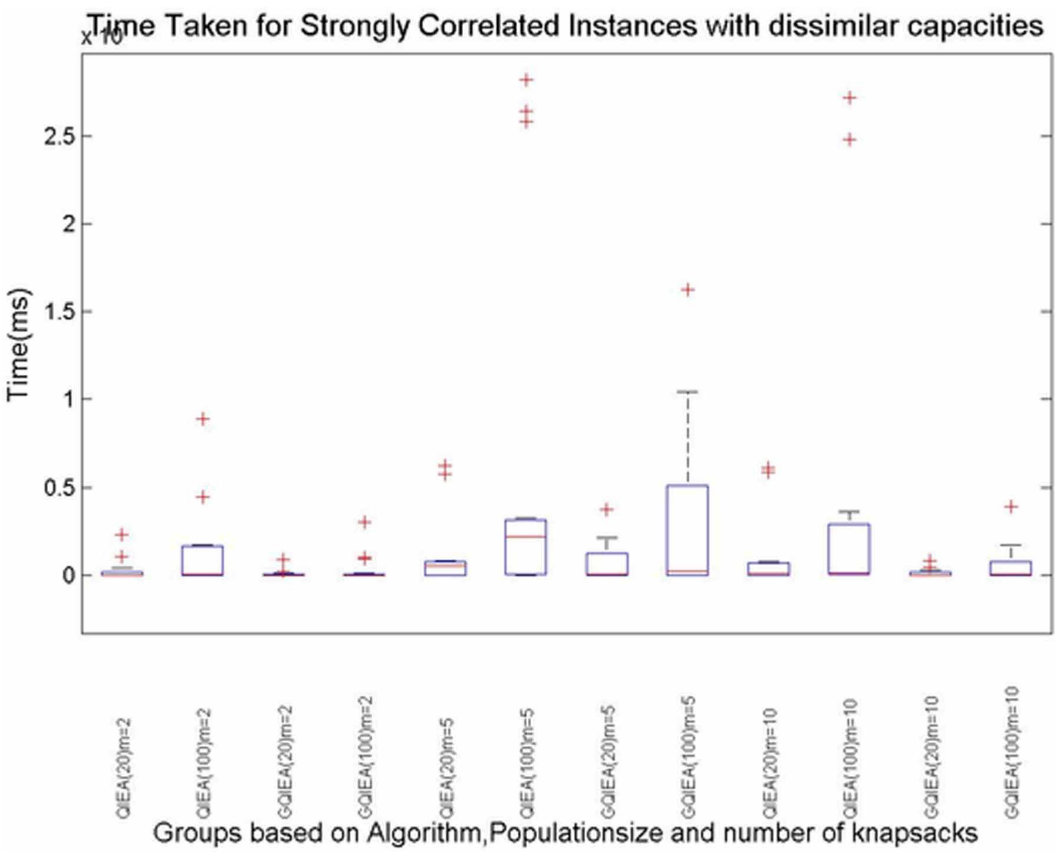


Figure 13. Boxplots for time taken using different strategies and population size to solve Uncorrelated Instances of MKP with dissimilar capacities having grouped by number of knapsacks

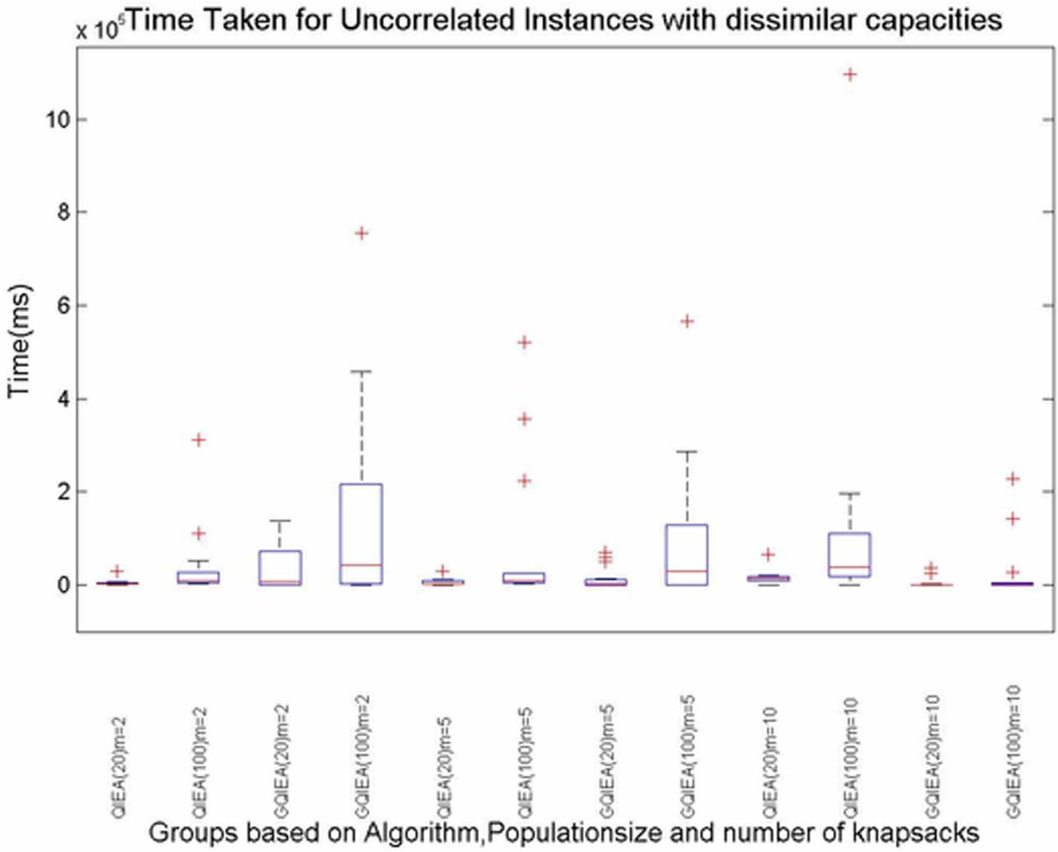


Figure 14. Boxplots for time taken using different strategies and population size to solve Strongly correlated Instances of MKP with similar capacities grouped by number of knapsacks

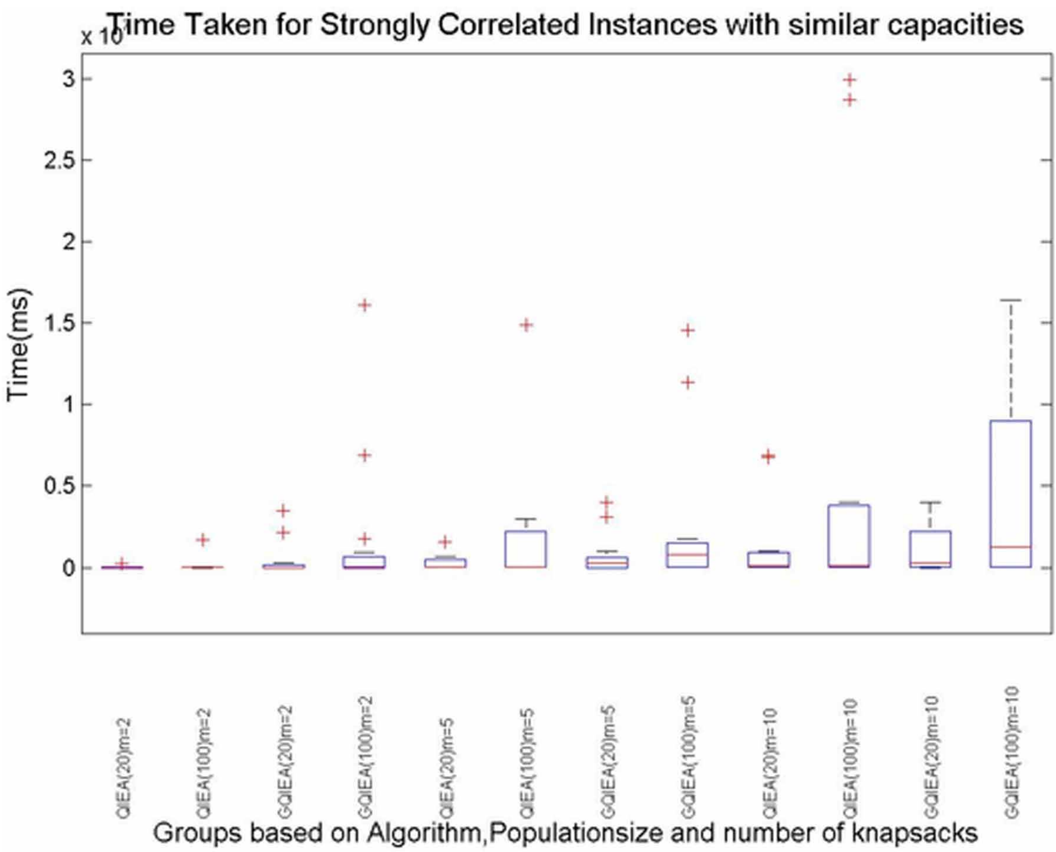


Figure 15. Boxplots for time taken using different strategies and population size to solve Uncorrelated Instances of MKP with similar capacities grouped by number of knapsacks

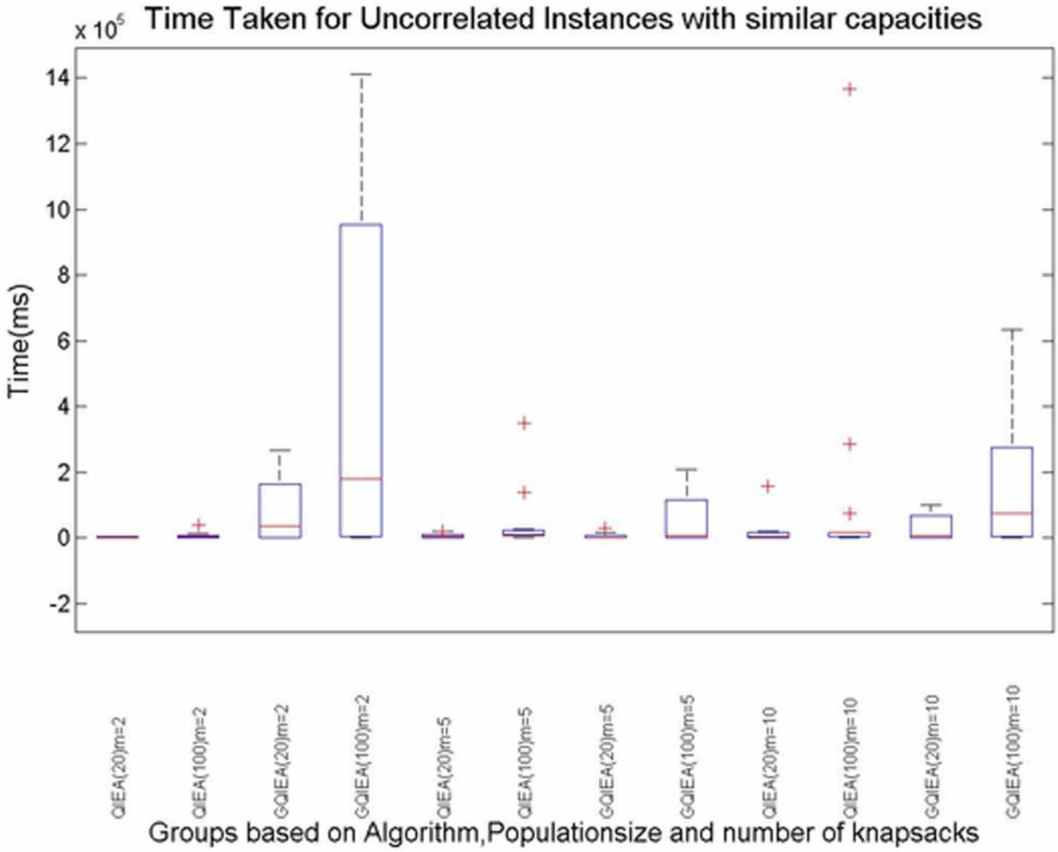


Figure 16. Boxplots for gain % in observed profit values over the heuristic values using different strategies and population size to solve Strongly Correlated Instances of MKP with dissimilar capacities grouped by size of problem

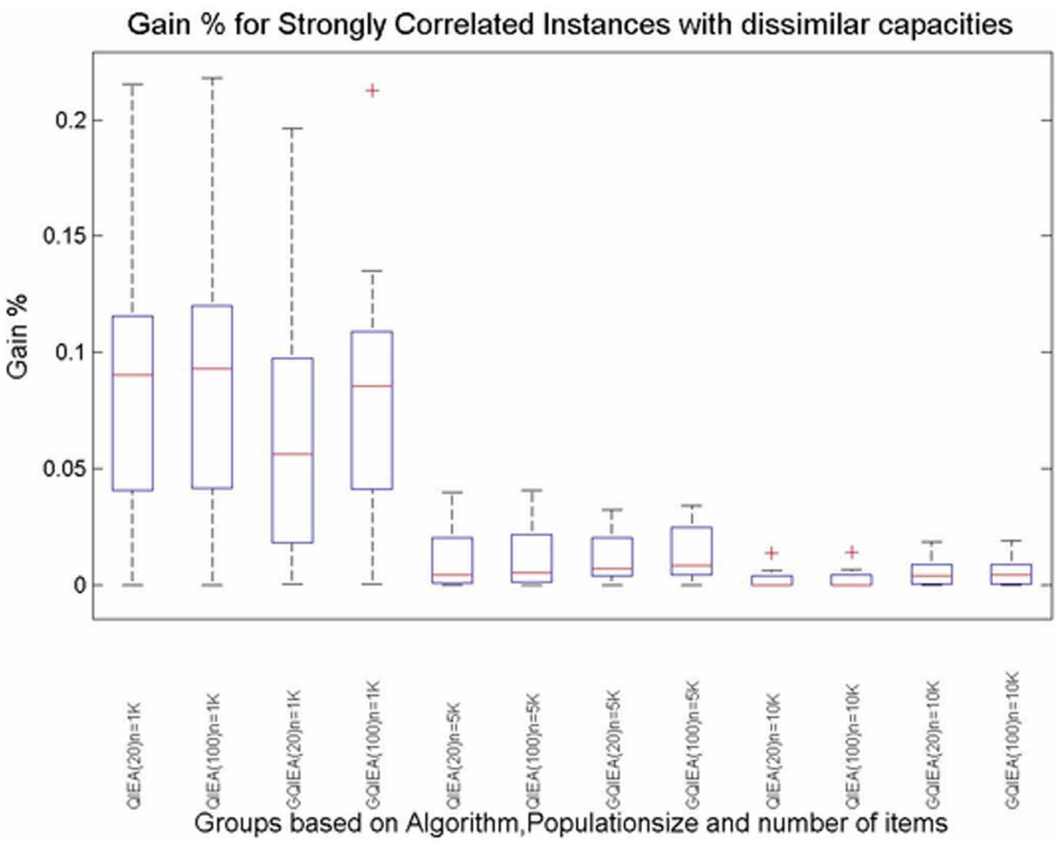
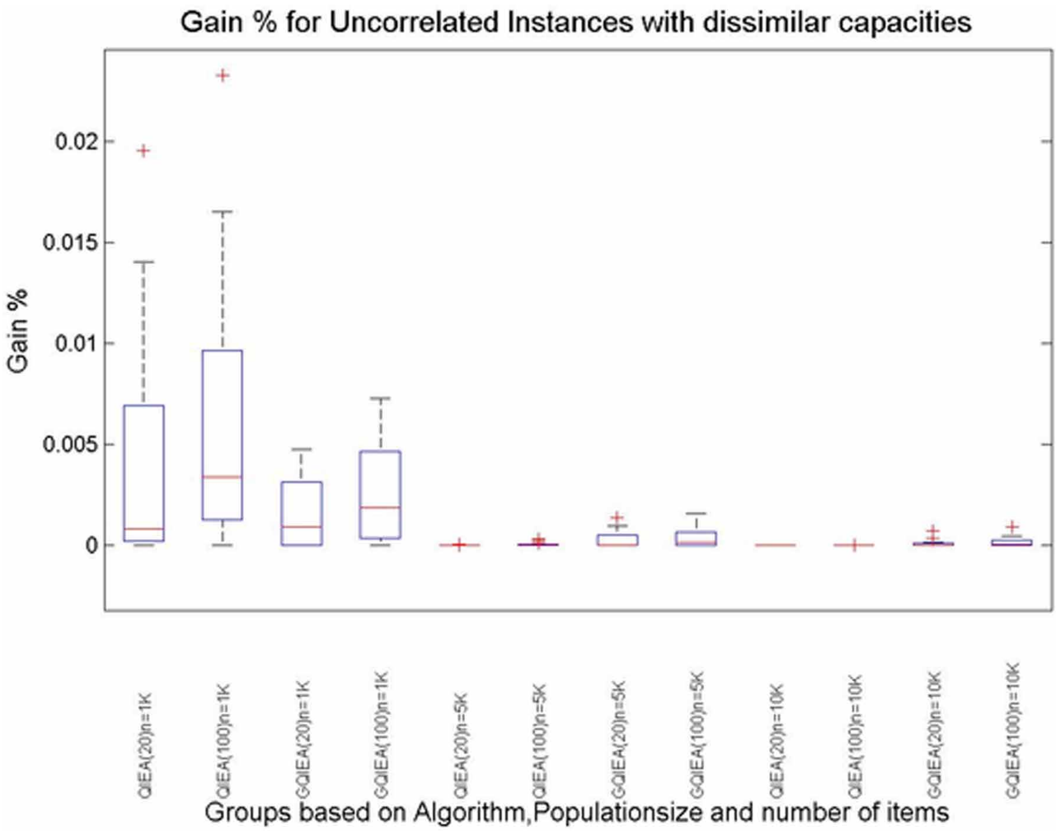


Figure 17. Boxplots for gain % in observed profit values over the heuristic values using different strategies and population size to solve Uncorrelated Instances of MKP with dissimilar capacities grouped by size of problem



REFERENCES

Alegria, J. M., & Tupac, Y. J. (2013). A Generalized Quantum-Inspired Evolutionary Algorithm for Combinatorial Optimization Problems. In *Proceedings of the XXXII International Conference of the Chilean Computer Science Society (SCCC),* San Pablo.

Arpaia, P., Maisto, D., & Manna, C. (2011). A Quantum-inspired Evolutionary Algorithm with a competitive variation operator for Multiple-Fault Diagnosis. *Applied Soft Computing*, *11*(8), 4655–4666. doi:10.1016/j.asoc.2011.07.017

Blum, C., Puchinger, J., Raidl, G. R., & Roli, A. (2011). Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing*, *11*(6), 4135–4151. doi:10.1016/j.asoc.2011.02.032

Chekuri, C., & Khanna, S. (2006). A PTAS for the multiple knapsack problem. *SIAM Journal on Computing*, *35*, 713–728. doi:10.1137/S0097539700382820

Diedrich, F., & Jansen, K. (2009). Improved approximation algorithms for scheduling with fixed jobs. In *Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms (SODA2009)* (pp. 675-684). doi:10.1137/1.9781611973068.74

Eilon, S., & Christofides, N. (1971). The loading problem. *Management Science*, *17*(5), 259–268. doi:10.1287/mnsc.17.5.259

Han, K.-H., & Kim, J.-H. (2002, December). Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization. *IEEE Transactions on Evolutionary Computation*, *6*(6), 580–593. doi:10.1109/TEVC.2002.804320

Kellerer, H. (1999). A polynomial time approximation scheme for the multiple knapsack problem. In *Proceedings of the 2nd Workshop on Approximation Algorithms for Combinatorial Optimization Problems, LNCS* (Vol. *1671*, pp. 51-62). doi:10.1007/978-3-540-48413-4_6

Kellerer, H., Pferschy, U., & Pisinger, D. (2004). Knapsack Problems. Berlin: Springer-Verlag. doi:10.1007/978-3-540-24777-7

Mani, A., & Patvardhan, C. (2010). A Hybrid quantum evolutionary algorithm for solving engineering optimization problems. *International Journal of Hybrid Intelligent Systems*, *7*(3), 225–235. doi:10.3233/HIS-2010-0115

Martello, S., & Toth, P. (1980). Solution of the zero-one multiple knapsack problem. *European Journal of Operational Research*, *4*(4), 276–283. doi:10.1016/0377-2217(80)90112-5

Martello, S., & Toth, P. (1990). *Knapsack Problems: Algorithms and Computer Implementations*. Chichester, UK: Wiley.

Patvardhan, C., Bansal, S., & Srivastav, A. (2014). Balanced quantum-inspired evolutionary algorithm for multiple knapsack problem. *International Journal of Intelligent Systems and Applications*, *11*(11), 1–11. doi:10.5815/ijisa.2014.11.01

Patvardhan, C., Narayan, A., & Srivastav, A. (2007). Enhanced Quantum Evolutionary Algorithms for Difficult Knapsack Problems. In *PReMI'07 Proceedings of the 2nd international conference on Pattern recognition and machine intelligence* (pp. 252-260). Springer-Verlag Berlin.

Patvardhan, C., Prakash, P., & Srivastav, A. (2012). A novel quantum-inspired evolutionary algorithm for the quadratic knapsack problem. *Int. J. Mathematics in Operational Research*, *4*(2), 114–127. doi:10.1504/IJMOR.2012.046373

Pisinger, D. (1999, May). An exact algorithm for large multiple knapsack problems. *European Journal of Operational Research*, *114*(3), 528–541. doi:10.1016/S0377-2217(98)00120-9

Pisinger, D. (n.d.). *David Pisinger's optimization codes.* Retrieved November 2, 2012, from http://www.diku.dk/~pisinger/codes.html

Platel, M. D., Schliebs, S., & Kasabov, N. (2007). A versatile quantum-inspired evolutionary algorithm. In *Proc. of CEC* (pp. 423-430).

Sailesh Babu, G. S., Bhagwan Das, D., & Patvardhan, C. (2008). Real-Parameter quantum evolutionary algorithm for economic load dispatch. *IET Generation, Transmission & Distribution*, *2*(1), 22–31. doi:10.1049/iet-gtd:20060495

Wang, L., & Li, L. (2010). An effective hybrid quantum-inspired evolutionary algorithm for parameter estimation of chaotic systems. *Expert Systems with Applications*, *37*(2), 1279–1285. doi:10.1016/j.eswa.2009.06.013

Xiao, J., Yan, Y., Zhang, J., & Tang, Y. (2010). A quantum-inspired genetic algorithm for k-means clustering. *Expert Systems with Applications*, *37*(7), 4966–4973. doi:10.1016/j.eswa.2009.12.017

Yang, S., Wang, M., & Jiao, L. (2004). A novel quantum evolutionary algorithm and its application. In *Proc of CEC* (pp. 820-826).

Yang, S., Wang, M., & Jiao, L. (2010). Quantum-inspired immune clone algorithm and multiscale Bandelet based image representation. *Pattern Recognition Letters*, *31*(13), 1894–1902. doi:10.1016/j.patrec.2009.12.016

International Journal of Applied Evolutionary Computation
Volume 9 • Issue 1 • January-March 2018

Modified Mutation in Asynchronous Differential Evolution

Vaishali, Amity University Rajasthan, Jaipur, India
Tarun K. Sharma, Amity University Rajasthan, Jaipur, India

ABSTRACT

This article presents a variant of an asynchronous differential evolution (ADE) algorithm for solving global optimization problems. The proposed algorithm uses the mean of two randomly chosen variables as a third variable to perform a mutation operation. The modification in a mutation operation is done to exploit the chosen random variables and to accelerate the convergence. The proposed algorithm is tested on a set of benchmark functions and compared with differential evolution (DE) and ADE. Results and comparisons show that the proposed work outperforms other algorithms in terms of number of function evaluation, standard deviation and convergence rate. Results are validated through non-parametric statistical analysis.

KEYWORDS

Asynchronous Differential Evolution, Convergence, Mutation, Optimization

INTRODUCTION

Evolutionary and population based metaheuristic algorithms have become very popular from last two decades. And these algorithms have proved their efficiency over other algorithms in finding the optimal solution more quickly (Back, 1996). These metaheuristic algorithms and their variants (Rajpurohit et al. 2017) have been used to optimize various benchmark functions, engineering design problems and real-life problems (Gupta et al., 2017; Kumar et al., 2013; Sharma & Pant, 2017; Sharma & Pant, 2017; Sharma & Pant, 2013; Sharma et al., 2013). In this paper Asynchronous differential evolution (ADE) algorithm has been used which supports asynchronous strategy to solve optimization problems (Zhabitskaya & Zhabitsky, 2012). ADE has been derived from Differential Evolution (DE). DE, a stochastic population based optimization algorithm was introduced by (Storn & Price,1995). DE is a generation based evolution strategy in which selection, mutation and crossover operations are performed on the population synchronously (Storn & Price,1997). Over the time DE has been proved its efficient in solving global and real-time optimization problems. Because DE is simple yet robust, it has been used efficiently in various fields of engineering like communication (Storn, 1996), mechanical engineering (Rogalsky et al., 2000), pattern recognition (Ilonen et al.,2003) and other fields (Dehmollaian, 2011; Gao et al., 2014; Gao et al., 2016; Monakhov et al., 2016).

ADE is well suited for parallelization and global optimization problems (Zhabitskaya & Zhabitsky, 2012). ADE also performs selection, mutation and crossover operations but asynchronously. ADE is not generation based, it iteratively improves the population. Unlike DE, the better vectors become part of the population as soon as they are found.

ADE has been proved better than DE and its variants in most of the cases (Vaishali & Sharma,2016). ADE variants have also been introduced over the past years. These variants have performed better

DOI: 10.4018/IJAEC.2018010103

than ADE itself and other algorithms. In (Zhabitskaya & Zhabitsky, 2012) authors introduced ADE and tested it by choosing either random or worst target vector. (Milani & Santucci,2010) proposed a way to weaken the problem of ordering of population members in dynamic differential evolution strategy(DDES) which works similar to ADE. They used Synchronization Degree parameter to regulate the current population evolution. In the proposed work reordering of the population was done when the algorithm switches to next generation. synchronization degree synchronizes the number of donor vectors. In (Ntipteni et al., 2006) asynchronous approach with master-slave architecture was used by Ntipteni et al. for parallel optimization. (Zhabitskaya, 2012) suggested the control parameters constraints to avoid premature convergence for one strategy of DE and four strategies of ADE. In (Zhabitskaya & Zhabitsky, 2012) authors proposed ADE with Restart (ADE-R). This approach automatically restarts calculations with an increased population size when stagnation or population degeneration encounters. In (Zhabitsky & Zhabitskaya,2013) a crossover operator based on adaptive correlation matrix was used instead of Crossover rate adaptation. In (Zhabitskaya et al., 2015) authors has shown that ADE outperforms Simplex and Migrad method for solving the optimization problem. In (Vaishali & Sharma,2016) ADE was incorporated with convex mutation (ADE-CM) in which the parent's information is used and hence it accelerates the convergence rate of the algorithm. In (Vaishali et al., 2018) ADE is incorporated with trigonometric mutation operation (ADE-TMO) to enhance the convergence rate of basic ADE. (Vaishali et al., 2018) tunes the trigonometric probability to obtain its best setting and the proposed algorithm is termed as ADE- trigonometric probability tuning (ADE-TPT). In the work done in (Vaishali et al., 2018) is extended and tested the algorithm on unconstrained engineering design problems. ADE has also been successfully implemented in various fields of biotechnology (Zhabitskaya et al., 2016; Zhabitskaya et al., 2014).

This paper incorporates a new mutation strategy in ADE in which the mean of two random variables is taken as the third variable for mutation and hence decreasing the dependency on less number of variables. The proposed work is compared with basic DE, DE with mean of random numbers and basic ADE.

The paper is organised as follows: The next section summarizes the working algorithm of ADE. Then the proposed algorithm is introduced. Next results, performance and statistical analysis of the proposed work are discussed and final section presents conclusions and future scope of the study.

ASYNCHRONOUS DIFFERENTIAL EVOLUTION

ADE performs better than DE for parallel calculations (Vaishali & Sharma, 2016). The asynchronous strategy of evolution used in ADE makes it better performer for various non-differentiable and nonlinear global optimization problems.

The ADE algorithm operates over a population of possible solutions which are randomly distributed in the search space. From the population, a target vector is chosen. Then three random variables r1, r2 and r3 different than the target vector are chosen. Then mutant vector is obtained by performing mutation operation and then target and mutant vectors are recombined using crossover operation in order to get trial vector. At last the selection operation is performed to select the better out of target or trial vector. The target vector is replaced by trial if and only if the trial vector gives better value for optimization function. In ADE if the trial vector provides better value, it is replaced with target vector in the current population and takes part in evolution without any time lag unlike DE where better trial vector can replace its target vector in the next generation only. Figure 1 summarizes the working of ADE.

PROPOSED WORK

As discussed in previous section for performing mutation operation three random variables namely r1, r2 and r3 are chosen from the population. In this paper two variables are chosen randomly but the

Figure 1. ADE algorithm

Step 1: // Population Initialization

$$X_i = \{x_{1,i}, x_{2,i}, x_{3,i}, \ldots, x_{D,i}\}$$

do {

i= target_vector();

Step 2: // Mutation Operation

$$V_i = X_{r1} + F \times (X_{r2} - X_{r3})$$

// i≠ r1≠ r2≠ r3

Step 3: // Crossover Operation

for (j=0;j<D; j=j+1)

$$u_{j,i} = \begin{cases} v_{j,i} & \text{if } rand_{i,j}[0,1] \le Cr \vee j = j_{rand} \\ x_{j,i} & \text{otherwise} \end{cases}$$

Step 4: // Selection Operation (for next iteration)

$$X_i = \begin{cases} U_i & \text{if } f(U_i) \le f(X_i) \\ X_i & \text{Otherwise} \end{cases}$$

} while the termination criteria is met.

third variable is taken as mean of two randomly chosen variable. The mean of variable is considered as third variable to make the algorithm dependent on less number of variables. The algorithm for proposed work is summarized in Figure 2.

EXPERIMENTAL RESULTS

The proposed work is tested over 10 benchmark function taken from (Suganthan et al.,2005) to evaluate the performance. The range of each function's search space in given in Table 1. Each function is optimized over 25 runs in MATLAB environment.

Parameter Settings

Population Size(NP): 50 and 100;

Dimension (D): 20; Crossover Rate (Cr): 0.9;

Value to reach (VTR): 10^{-3} and 10^{-2} (in case of rastrigin & rosenbrock respectively) and 10^{-6} (for others);

Maximum number of function evaluations: 10^6

PC Configuration: Windows 7; CPU-Intel core i3 processor CPU @ 2.20 GHz; RAM-4GB.

Performance Evaluation Criteria

To verify and validate the efficiency of proposed algorithm ADE with mean random value, it is compared with basic DE algorithm, basic ADE algorithm and DE with mean random numbers. This comparison is done on the basis of following metrics taken from (Suganthan et al.,2005): Average number of function evaluations (NFE), Standard deviation (Std. dev.), Convergence graph and Success rate. Success rate is determined by dividing the number of successful runs to the total number of runs.

A non-parametric statistical analysis (Demšar, 2006; Garcia & Herrera,2008) is performed to calculate the efficiency of the algorithm. Significant difference is also calculated using Bonferroni-Dunn test (Dunn,1961). For Bonferroni-Dunn's graph, the critical difference is calculated as:

Figure 2. Proposed ADE algorithm with mean random numbers

Step 1: // Population Initialization

$$X_i = \{x_{1,i}, x_{2,i}, x_{3,i}, \ldots, x_{D,i}\}$$

```
    do {
      i=  target_vector();
```

Step 2: // Mutation Operation

```
        do {
        r1=rand[1,NP];
            } while (r1==i);
        do {
        r2=rand[1,NP];
            } while(r2==i || r2==r1);
        r3=(r1+r2)/2;
        if (r3==i || r3==r2 || r3==r1)
        {
        do{
        r3=rand[1,NP];
           } while (r3==i || r3==r2 || r3==r1);
        }
```

$$V_i = X_{r1} + F \times (X_{r2} - X_{r3})$$

Step 3: // Crossover Operation

for (j=0;j<D; j=j+1)

$$u_{j,i} = \begin{cases} v_{j,i} & \text{if } rand_{i,j}[0,1] \leq Cr \vee j = j_{rand} \\ x_{j,i} & \text{otherwise} \end{cases}$$

Step 4: // Selection Operation (for next iteration)

$$X_i = \begin{cases} U_i & \text{if } f(U_i) \leq f(X_i) \\ X_i & \text{Otherwise} \end{cases}$$

} while the termination criteria is met.

$$CD = Q_\alpha \sqrt{\frac{k\left(k+1\right)}{6N}}$$

$Q\alpha$ is critical value for a multiple non-parametric comparison with control (Zar,1999), k and N represent the total number of algorithms and number of problems considered for comparison respectively. To show two levels of significance at α=0.05 and 0.10, horizontal lines are drawn.

Results

Table 1 depicts the average NFE and Std. dev. for different algorithms considered for comparison when the function approaches VTR at D=20 and NP=50.

Table 2 depicts the average NFE and Std. dev. for different algorithms considered for comparison when the function approaches VTR at D=20 and NP=100. The best values of NFEs are highlighted in Table 1 and Table 2.

Figure 3(a-e) represents the convergence graph with NFE on x-axis and value of fitness function on y-axis to elaborate the performance of proposed algorithm over 5 benchmark functions at D=20

Table 1. Average NFE and Std. Dev. for benchmark function at D=20 and NP=50

Function	Range	DE	DE with Mean Random Number	ADE	ADE with Mean Random Number
		Average NFE Std. Dev	Average NFE Std. Dev	Average NFE Std. Dev	Average NFE Std. Dev
Sphere	[-100,100]	2.353e+04 8.7532e-08	2.260e+04 1.2296e-07	2.075e+04 1.1902e-07	**2.005e+04** 1.7650e-07
Hyper- Ellipsoid	[-50,50]	2.599e+04 9.3753e-08	2.503e+04 1.9525e-07	2.332e+04 9.2798e-08	**2.276e+04** 1.5506e-07
Ackley	[-30,30]	3.291e+04 4.8542e-07	3.162e+04 4.0957e-07	2.961e+04 3.6442e-07	**2.832e+04** 2.8779e-07
Griewank	[-30,30]	2.035e+04 4.3143e-03	1.908 e+04 5.500 e-03	1.919e+04 6.5 e-03	**1.701e+04** 8.3e-03
Rastrigin	[-5.12,5.12]	1.374 e+05 3.0034	7.050 e+04 6.6074	1.1912e+05 2.9898	**6.806 e+04** 5.1127
Rosenbrock	[-2,2]	3.379 e+04 1.5116	**2.489 e+04** 8.386 e-01	1.3485e+05 4.0744	5.7644e+04 3.1459
Sum of different powers	[-1,1]	5.730 e+03 1.8017e-08	5.570 e+03 2.1359e-08	5.5978e+03 2.9020e-08	**5.1696e+03** 1.7215e-08
Alpine	[-10,10]	5.731e+04 4.4527e-07	4.373e+04 2.8359e-07	5.1153e+04 2.0666e-07	**4.1476e+04** 2.9609e-07
Dejong	[-1.28,1.28]	1.029 e+04 6.6213e-08	1.046 e+04 2.0193e-06	9.495 e+03 1.0854e-07	**9.2906e+03** 1.5401e-07
Schwefel 1.2	[-100,100]	2.143 e+04 5.9023e-12	2.031 e+04 1.4694e-11	2.1549e+04 1.0418e-11	**1.9003e+04** 3.8937e-11

and NP=50. DE-M is DE with mean random number and ADE-M is ADE with mean random number. The success rate of every function for each algorithm is reported as 100%.

Table 3 and 4 describes the results of non-parametric statistical analysis based on Friedman's test and Table 5 and 6 summarizes the statistics based on Wilcoxon Signed test. Figure 4 represents the Bonferroni-Dunn's graph corresponding to average NFE.

Figure 4 depicts the significant difference between algorithm using Bonferroni-Dunn's graph.

Performance Analyses

The proposed work focuses on accelerating the ADE by modifying the mutation operation. The results verify that the proposed algorithm outperforms other algorithms which are considered for comparison.

At D=20 and NP=50, the proposed algorithm converges faster as the NFEs are lesser than other algorithms for all benchmark functions except Rosenbrock function. Rosenbrock function converges with least NFEs for DE with mean random number algorithm. However, the standard deviation for rastrigin and rosenbrock function is greater than zero except for DE with mean random number algorithm.

At D=20 and NP=100, except for Griewank and Rosenbrock function, the proposed algorithm outperforms other algorithms. These both functions perform best with ADE. And for rastrigin function the std. dev. is also greater than zero.

Hence except rosenbrock and griewank function, the proposed work performs better than other considered algorithm for all benchmark functions.

Table 2. Average NFE and Std. Dev. for benchmark function at D=20 and NP=100

Function	Range	DE	DE with Mean Random Number	ADE	ADE with Mean Random Number
		Average NFE Std. Dev.	Average NFE Std. Dev.	Average NFE Std. Dev.	Average NFE Std. Dev.
Sphere	[-100,100]	5.983e+04 1.3632e-07	5.664e+04 1.0504e-07	5.4248e+04 7.5856e-08	**5.1668e+04** 9.6748e-08
Hyper- Ellipsoid	[-50,50]	6.625e+04 9.0740e-08	6.280e+04 1.2473e-07	6.008e+04 1.3155e-07	**5.6271e+04** 1.0665e-07
Ackley	[-30,30]	8.441e+04 1.9138e-07	7.975e+04 1.9372e-07	7.7140e+04 4.0277e-07	**7.3068e+04** 2.7606e-07
Griewank	[-30,30]	5.556e+04 1.1602e-07	5.797e+04 5.200 e-03	**4.8299e+04** 2.300 e-03	5.2836e+04 4.300 e-03
Rastrigin	[-5.12,5.12]	5.725e+05 1.4392	2.977e+05 1.5940	6.0300e+05 1.7557	**2.8920e+05** 2.2395
Rosenbrock	[-2,2]	1.624e+05 3.00 e-03	1.710e+05 4.400 e-03	**1.2654e+05** 1.100 e-03	1.3677e+05 2.800 e-03
Sum of different powers	[-1,1]	1.460e+04 1.1636e-08	1.437e+04 1.3905e-08	1.3883e+04 1.2250e-08	**1.3337e+04** 1.0835e-08
Alpine	[-10,10]	2.455e+05 1.8484e-07	1.870e+05 1.5844e-07	2.479e+05 1.6879e-07	**1.770e+05** 1.6149e-07
Dejong	[-1.28,1.28]	2.582e+04 4.5081e-08	2.470e+04 3.2949e-08	2.3424e+04 3.1775e-08	**2.2535e+04** 3.4215e-08
Schwefel 1.2	[-100,100]	6.187e+04 7.0492e-13	5.961e+04 9.6567e-13	6.000e+04 2.9912e-12	**5.521e+04** 1.7115e-12

A rigorous non-parametric statistical analysis (Table 3-6) verifies and validates the proposed algorithm to be best amongst all considered algorithms.

Bonferroni-Dunn's graph represents the significant difference between algorithms. To show two levels of significance, α=0.05 and 0.10, a horizontal line is drawn. The test presents the significant difference with:

- ADE with mean random number as control algorithm:
  - At α=0.05: ADE-M is better than DE, ADE, DE-M .
  - At α=0.10: ADE-M is better than DE, ADE, DE-M.

CONCLUSION AND FUTURE SCOPE

The work done in this paper enhances the ADE algorithm by modifying the way of choosing random variables for performing mutation operation. For mutation operation two variables r1 and r2 are chosen randomly from the population and the mean value of these randomly chosen variable is taken as third variable r3. Hence the dependency on r3 is removed. The proposed method is very simple and easy to implement but it helps the population to converge faster. The experimental results of benchmark functions prove the efficacy and superiority of the proposed approach. In future the proposed algorithm can be tested with higher dimension population and over real life engineering design problems.

Figure 3. (a) Sphere Function; (b)Rastrigin Function (c) Rosenbrock Function (d) Alpine Function (e) Schwefel 1.2

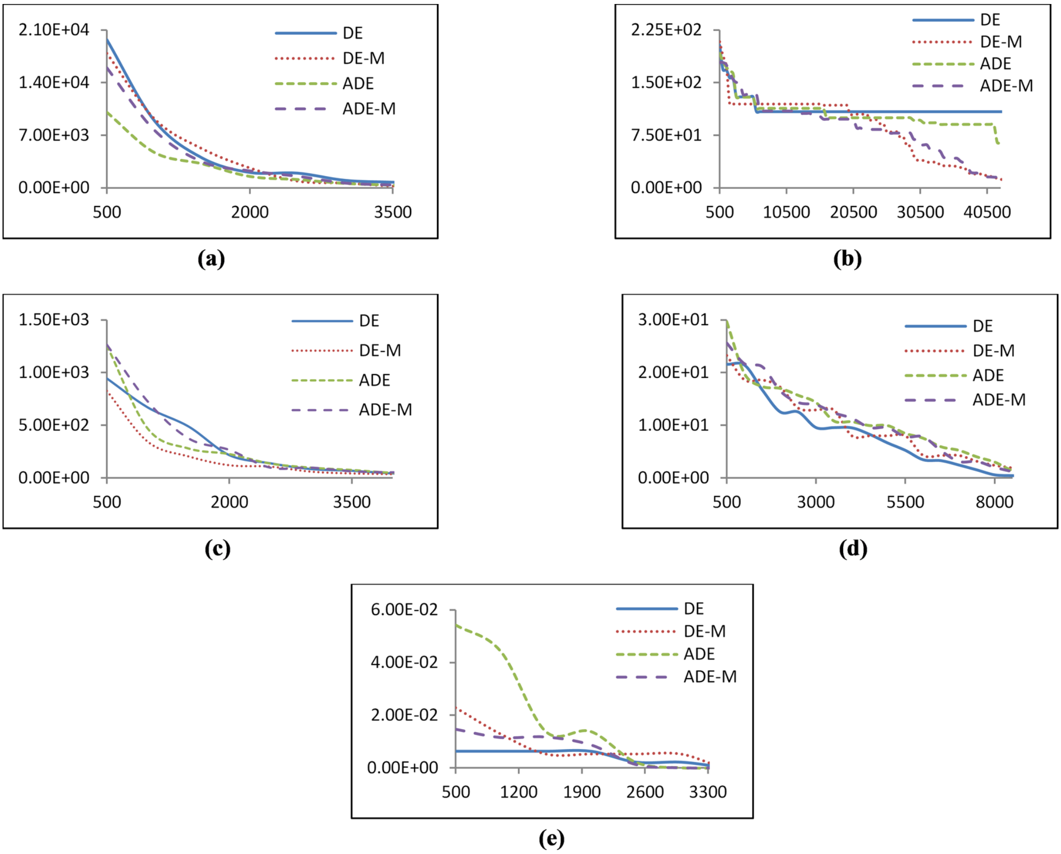


Table 3. Ranking based on Friedman's test

Algorithm	Mean Rank
DE	4.00
DE-M	2.60
ADE	2.40
ADE -M	1.00
CD $\left(\boldsymbol{a} = 0.05\right)$	1.955
CD (a=0.10)	1.740

Table 4. Statistics based on Friedman's test

N	5
Chi-square	13.560
df	3
Asymp. Sig.	.004

Table 5. Ranking based on Wilcoxon Signed rank test

		N	Mean Rank	Sum of Ranks
DE - ADE with mean random number	Negative Ranks	0[a]	.00	.00
	Positive Ranks	5[b]	3.00	15.00
	Ties	0[c]		
	Total	5		
DE with mean random number - ADE with mean random number	Negative Ranks	0[d]	.00	.00
	Positive Ranks	5[e]	3.00	15.00
	Ties	0[f]		
	Total	5		
ADE - ADE with mean random number	Negative Ranks	0[g]	.00	.00
	Positive Ranks	5[h]	3.00	15.00
	Ties	0[i]		
	Total	5		

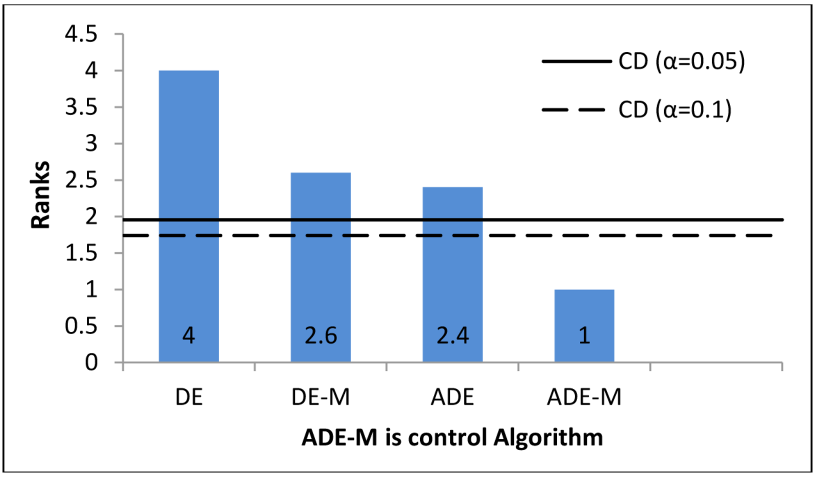
a. DE < ADE-M b. DE > ADE-M c. DE = ADE-M d. DE-M < ADE-M e. DE-M > ADE-M f. DE-M = ADE-M g. ADE < ADE-M h. ADE > ADE-M i. ADE = ADE-M

Table 6. *Statistics based on Wilcoxon Signed test*

	DE - ADE with mean random number	DE with mean random number - ADE with mean random number	ADE - ADE with mean random number
Z	-2.023[a]	-2.023[a]	-2.023[a]
Asymp. Sig. (2-tailed)	.043	.043	.043

a. Based on negative ranks

Figure 4. Bonferroni-Dunn's graph corresponding to average NFE



REFERENCES

Back, T. (1996). *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.

Dehmollaian, M. (2011). Through-wall shape reconstruction and wall parameters estimation using differential evolution. *IEEE Geoscience and Remote Sensing Letters*, *8*(2), 201–205. doi:10.1109/LGRS.2010.2056912

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*(Jan).

Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American Statistical Association*, *56*(293), 52–64. doi:10.1080/01621459.1961.10482090

Gao, Z., Pan, Z., & Gao, J. (2014). A new highly efficient differential evolution scheme and its application to waveform inversion. *IEEE Geoscience and Remote Sensing Letters*, *11*(10), 1702–1706. doi:10.1109/LGRS.2014.2306263

Gao, Z., Pan, Z., & Gao, J. (2016). Multimutation Differential Evolution Algorithm and Its Application to Seismic Inversion. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(6), 3626–3636. doi:10.1109/TGRS.2016.2520978

Garcia, S., & Herrera, F. (2008). An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *Journal of Machine Learning Research*, *9*(Dec), 2677–2694.

Gupta, P., Sharma, T. K., Mehrotra, D., & Abraham, A. (2017). Knowledge building through optimized classification rule set generation using genetic based elitist multi objective approach. *Neural Computing & Applications*.

Ilonen, J., Kamarainen, J. K., & Lampinen, J. (2003). Differential evolution training algorithm for feed-forward neural networks. *Neural Processing Letters*, *17*(1), 93–105. doi:10.1023/A:1022995128597

Kumar, S., Kumar, P., Sharma, T. K., & Pant, M. (2013). Bi-level thresholding using PSO, artificial bee colony and MRLDE embedded with Otsu method. Memetic Computing, 5(4), 323–334. 10.1007/s12293-013-0123-5

Milani, A., & Santucci, V. (2010). Asynchronous differential evolution. In *Proceedings of the IEEE Congr* (pp. 1210–1216). Evol. Comput; . doi:10.1109/CEC.2010.5586107

Monakhov, O. G. E., Monakhova, E. A., & Pant, M. (2016). Application of differential evolution algorithm for optimization of strategies based on financial time series. *Numerical Analysis and Applications*, *9*(2), 150–158. doi:10.1134/S1995423916020063

Ntipteni, M. S., Valakos, I. M., & Nikolos, I. K. (2006). An asynchronous parallel differential evolution algorithm. In *Proceedings of the ERCOFTAC conference on design optimisation: methods and application*.

Rajpurohit, J., Sharma, T. K., & Abraham, A. (2017). Vaishali (2017). Glossary of Metaheuristic Algorithm. *International Journal of Computer Information Systems and Industrial Management Applications*, *9*, 181–205.

Rogalsky, T., Kocabiyik, S., & Derksen, R. W. (2000). Differential evolution in aerodynamic optimization. *Canadian Aeronautics and Space Journal*, *46*(4), 183–190.

Sharma, T. K., & Pant, M. (2013). Enhancing the food locations in an artificial bee colony algorithm. *Soft Computing*, *17*(10), 1939–1965. doi:10.1007/s00500-013-1029-3

Sharma, T. K., & Pant, M. (2017). Opposition based learning ingrained shuffled frog-leaping algorithm. *Journal of Computational Science*, *21*, 307–315. doi:10.1016/j.jocs.2017.02.008

Sharma, T. K., & Pant, M. (2017). Shuffled artificial bee colony algorithm. *Soft Computing*, *21*(20), 6085–6104. doi:10.1007/s00500-016-2166-2

Sharma, T. K., Pant, M., & Singh, M. (2013). Nature-inspired metaheuristic techniques as powerful optimizers in the paper industry. *Materials and Manufacturing Processes*, *28*(7), 788–802. doi:10.1080/10426914.2012.736650

Storn, R. (1996, June). On the usage of differential evolution for function optimization. In *Proceedings of the 1996 Biennial Conference of the North American Fuzzy Information Processing Society* (pp. 519-523). IEEE. 10.1109/NAFIPS.1996.534789

Storn, R., & Price, K. (1995). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces (tech. rep.).

Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359. doi:10.1023/A:1008202821328

Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y. P., Auger, A., & Tiwari, S. (2005). Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization (KanGAL report).

Vaishali, S. T. K. (2016). Asynchronous Differential Evolution with Convex Mutation. In *Proceedings of Fifth International Conference on Soft Computing for Problem Solving* (pp. 915–928). Springer Singapore; . doi:10.1007/978-981-10-0451-3_81

Vaishali, S. T. K. (in press). Asynchronous differential evolution with trigonometric probability tuning. *International Journal of Ambient Computing and Intelligence*.

Vaishali, S. T. K., Abraham, A., & Rajpurohit, J. (2018). Enhanced Asynchronous Differential Evolution Using Trigonometric Mutation. In *Proceedings of the Eighth International Conference on Soft Computing and Pattern Recognition (SoCPaR 2016)*, AISC (Vol. 614). Springer, Cham

Vaishali, S. T. K., Abraham, A., & Rajpurohit, J. (2018) Trigonometric Probability Tuning in Asynchronous Differential Evolution. In *Proceedings of the International Conference on Soft Computing: Theories and Applications, AISC* (Vol. 584). Springer, Singapore.

Zar, J. H. (1999). *Biostatistical Analysis*. Englewood Cliffs, NJ: Prentice-Hall Inc.

Zhabitskaya, E. (2012). Constraints on control parameters of asynchronous differential evolution. In *Mathematical Modeling and Computational Science* (pp. 322–327). Springer Berlin Heidelberg; . doi:10.1007/978-3-642-28212-6_40

Zhabitskaya, E., Zemlyanaya, E., Kiselev, M., & Gruzinov, A. (2016). The Parallel Asynchronous Differential Evolution Method as a Tool to Analyze Synchrotron Scattering Experimental Data from Vesicular Systems. In *EPJ Web of Conferences* (Vol. 108, p. 02047). EDP Sciences; . doi:10.1051/epjconf/201610802047

Zhabitskaya, E., & Zhabitsky, M. (2012). Asynchronous differential evolution. In Mathematical Modeling and Computational Science (pp. 328-333).

Zhabitskaya, E., & Zhabitsky, M. (2012, June). Asynchronous differential evolution with restart. In *Proceedings of the International Conference on Numerical Analysis and Its Applications* (pp. 555-561). Springer Berlin Heidelberg.

Zhabitskaya, E. I., Zemlyanaya, E. V., & Kiselev, M. A. (2014). Unilameller DMPC Vesicles Structure Analysis using Parallel Asynchronous Differential Evolution. BULLETIN of Peoples' Friendship University of Russia. Series Mathematics. Information Sciences. *Physics (College Park, Md.)*, (2): 253–258.

Zhabitskaya, E. I., Zemlyanaya, E. V., & Kiselev, M. A. (2015). Numerical analysis of SAXS-data from vesicular systems by asynchronous differential evolution method. *Matematicheskoe Modelirovanie*, *27*(7), 58–64.

Zhabitskaya, E. I., Zhabitsky, M. V., Zemlyanaya, E. V., & Lukyanov, K. V. (n.d.). Calculation of the parameters of microscopic optical potential by Asynchronous Differential Evolution algorithm.

Zhabitsky, M., & Zhabitskaya, E. (2013, July). Asynchronous differential evolution with adaptive correlation matrix. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation* (pp. 455–462). ACM; . doi:10.1145/2463372.2463428

Vaishali is pursuing her PhD in the domain of evolutionary computing. She is BTech and MTEch in Computer Science in Engineering. She is presently affiliated with Amity University Rajasthan as an Assistant Professor in the Department of CSE/IT, Amity University Rajasthan. She has published several papers in the International Conferences and Journals of repute. She has almost 6 year of teaching experience. Her research interest includes design and analysis of Hybrid Evolutionary Computing and their application in versatile engineering domains.

Tarun K. Sharma received a PhD in the field of Artificial Intelligence from Indian Institute of Technology (IIT), Roorkee, India. Presently he is affiliated with Amity University Rajasthan, Jaipur. His current research interests encompass Swarm Intelligence, Nature inspired algorithm and their applications in software engineering, inventory systems and image processing. He has published more than 60 research papers in refereed International Journals and Conferences of repute. He received MCA degree from SGRRITS, Dehradun, affiliated to HNB Garhwal University, Srinagar, UK, India; MBA (Marketing) from VMOU Kota, Rajasthan and MTech (IT) from KSOU Karnataka. He has almost 16 years of experience and exposure.

International Journal of Applied Evolutionary Computation
Volume 9 • Issue 1 • January-March 2018

In Body Communication:
Assessment of Multiple Homogeneous Human Tissue Models on Stacked Meandered Patch Antenna

Raghvendra Singh, Department of Electronics and Communication Engineering, JK Lakshmipat University, Jaipur, India
Dambarudhar Seth, Department of Electronics and Communication Engineering, JK Lakshmipat University, Jaipur, India
Sanyog Rawat, Department of Electronics and Communication Engineering, Manipal University, Manipal, India
Kanad Ray, Amity School of Applied Sciences, Amity University Rajasthan, Jaipur, India

ABSTRACT

This article makes an effort to assess the performance of a stacked meandered patch antenna in proximity of multiple homogeneous human tissue models. Nowadays, many smart wireless gadgets are being used around human vicinity, hence the performance investigation of these wireless gadgets is major concern. This article discloses the performance of meandered patch antenna for in-body communication in the MICS band. The human tissue is considerably effected by the exposure of electromagnetic radiation. Performance improvement of patch antennas used in wireless gadgets is the key solution of the problem. This article presents the estimation of parameters of stacked meandered patch antennas; reflection coefficient, radiation pattern, directive gain and VSWR in proximity of multiple homogeneous human tissue models.

KEYWORDS

Human Body, Meandered Patch Antenna, Medical Implant Communication Services (MICS) Band, On Body Communication, Reflection Coefficient

1. INTRODUCTION

Body area network is showing tremendous growth in the domains of health monitoring and tracking, soldiers monitoring, sports and entertainment industry. The electronic devices and gadgets use wireless communication to transfer data and antenna is most important part of these devices. The Body area network uses licensed wireless medical telemetry services (WMTS), licensed Medical Implant Communication services (MICS), unlicensed Industrial, Scientific, and medical (ISM) band, and ultra-wide band(UWB), etc. (Ullah et al.., 2009). There are some restraints related to the design of these antennas: small size, robustness, low power consumption and easiness (Gareth A. et al.,2009; Rahmat-Samii et al., 2005).

Integration of antennas over the wearable textile is investigated by many investigators including (Rahmat-Samii et al., 2005; Ouyang et al., 2005; Tanaka et al., 2005), it presents linearly polarized wearable antennas. Electromagnetic band gap structures based fabric antennas are investigated in (Zhu et al., 2007). As a part of Body Area Network MICS band antennas are analyzed by (Guo et al., 2007), working in 402-405MHz.The state-of-the-art in the research field of implanted devices has shown that

DOI: 10.4018/IJAEC.2018010104

MICS (402–405 MHz) band can be utilized in communication with medical implants (IEEE1999). The European Telecommunications Standards Institute (ETSI) and FCC have standardized the MICS: the maximum emission bandwidth to be occupied is 300 kHz, the equivalent radiated power (ERP), i.e. the maximum field-strength in any direction should be equal to, or lower than, what a resonant dipole would give in its maximum direction at the same distance, with the dipole being fed with a signal of 25 micro watt (FCC 1996). International Telecommunication Union (ITU) defined ISM bands (2.4–2.4835 GHz) and the use of radio frequency (RF) energy is for industrial, scientific and medical purposes other than communications.

The performance of planar inverted F antenna (PIFA) for body worn applications and proximity of human body to antenna is discussed by (Terence SP. Et al, 2005). Similarly, different ISM band antennas are reported in (Kamarudin MR., 2005) e.g.; monopole antennas, patch antenna and its arrays in proximity of human body. Large dimensions of the antennas are major limitations for implanted of body worn antennas.

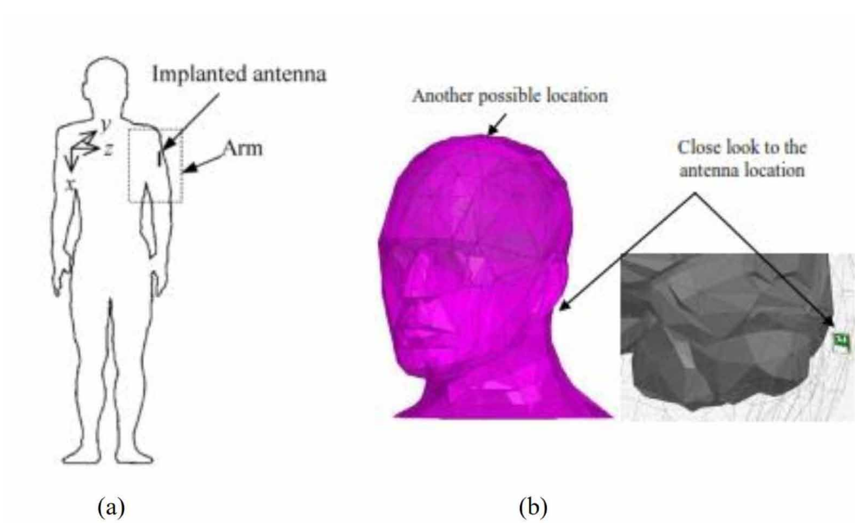
Worldwide interoperability for microwave access (WiMax) is also a renounced field nowadays, operating at 2.5GHz and 5.5 GHz band (Liu et al., 2010), a 2500 mm^2 slot ring antenna is reported for IEEE802.11 and WiMax applications (Sim et al., 2011). In aforesaid reported antenna proposals, the investigators are achieving appropriate antennas comprising radiation performance or gain or fabrication cost. Many locations are suggested for implanted antennas as in Figure 1 by Kiourti et al. & Yazdandoost, 2012.

There is always scope for researchers to do work designing and fabrication of low profile, better radiation profile good gain value antennas, so that the wireless Body Area Network can be more efficient.

2. ANTENNA MODEL DEVELOPMENT

There are several strategies for implantable antenna design for human body but mostly are stated by the fact that antennas are to operate inside the human tissue instead of open atmosphere. According to first strategy the antenna should be designed in free space then refined for covered tissue model. In second strategy an antenna can be designed directly in an environment surrounded by human tissues (http://niremf.ifac.cnr.it).

Figure 1. Locations of Implanted antennas on human body (a) Arm (b) Head (Kiourti et al. & Yazdandoost, 2012)



Here stacked meandered implantable patch antenna is designed directly in an environment surrounded by human tissue (Singh et al, 2015). The wavelength in the tissues is shorter, since the velocity of wave propagation is lowered (Gareth et al., 2009). Hence, the geometrical size of implantable antenna is usually less than 10 percent of wavelength in open environment (Kiourti et al., 2012).

2.1 Antenna Model

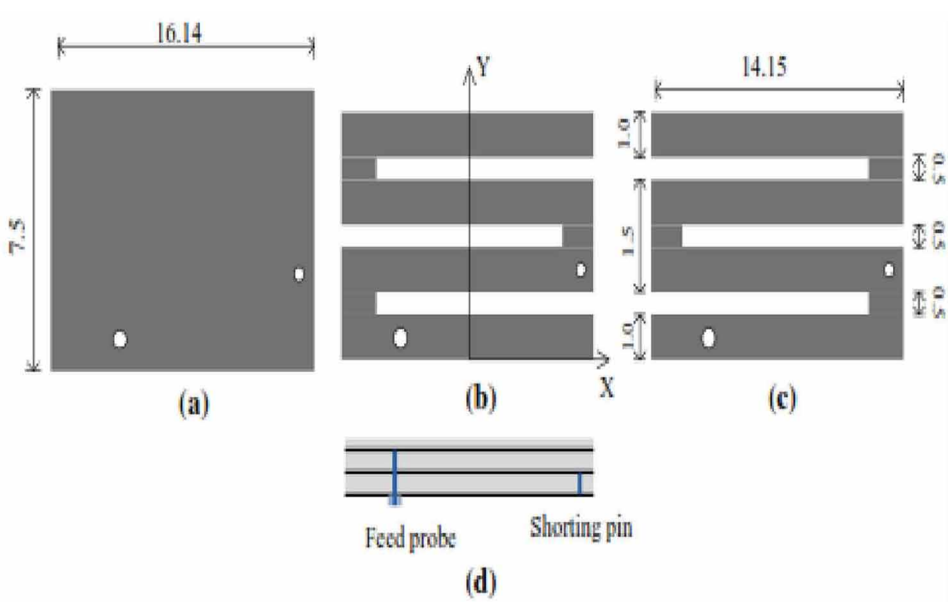
Stacked meandered implantable patch antenna is designed inside human tissue model directly using accelerated design to save time and resources (Singh et al., 2015). Other strategy in which the researchers model an antenna first in air and further simulate in different tissue models take more time and resources. Hence accelerated modeling is used which is also reported by (Kiourti et al., 2012). In optimization of antenna the cost function is always less than -10db.

$$\text{Cost } [S_{11}] < -10\text{dB} \tag{1}$$

Due to the above equation (1) the cost condition in optimization technique the reflection coefficient (S_{11}) is always lesser than -10db. Stacked meandered implantable patch antenna is designed as per the dimensions presented in Figure 2 using Rogers 3010 as substrate.

The dimensions of the proposed antenna are calculated by the standard formulae of rectangular patch antenna at frequency 402 MHz. Proposed antenna is miniaturized well due to stacked patches hence it has surface area only 121.05 mm^2. Figure 2 (a), (b), (c), (d) is depicting ground plane dimensions, lower patch view, upper patch view and side view with all layers respectively. A shorting pin is also used to get better impedance matching and miniaturization of the antenna in MHz range of frequencies.

Figure 2. Geometric view and Dimensions (in millimeter) of Stacked meandered implantable patch antenna (Singh R. et al, 2015) (a) Ground plane (b) lower patch (c) upper patch (d) side view with all layers



3. TISSUE MODEL DEVELOPMENT

Stacked meandered implantable patch antenna is analyzed inside the different homogeneous lossy equivalent dielectrics layers: skin, muscle fat and bone that is electrical equivalent of biological tissues. Usually the biological tissues contain their own permittivity (εr), conductivity (σ), loss tangent and mass-density values that are shown in Table 1. Homogeneous and layered model approach of biological tissues is used here to make fast simulations and for easier design of implantable antennas (Singh R. et al., 2015). Here multiple homogeneous layer model (e.g. Figure 3) is used which is providing a simplified model of Stacked meandered implantable patch antenna implanted inside the biological tissues.

The layered model is inspired from C. Gabriel & S. Gabriel, 1996. Properties of canonical model tissue structure are given in Table 1 (C. Gabriel & S. Gabriel, 1996). All values are generated by Microsoft excel tool on proposed frequency of operation.

3.1. Dielectric Spectrum of a Tissue

Following expression is given by Gabriel et al., (1996) for approximation of complex relative permittivity ε as a function of angular frequency ω:

Figure 3. Four layer tissue model

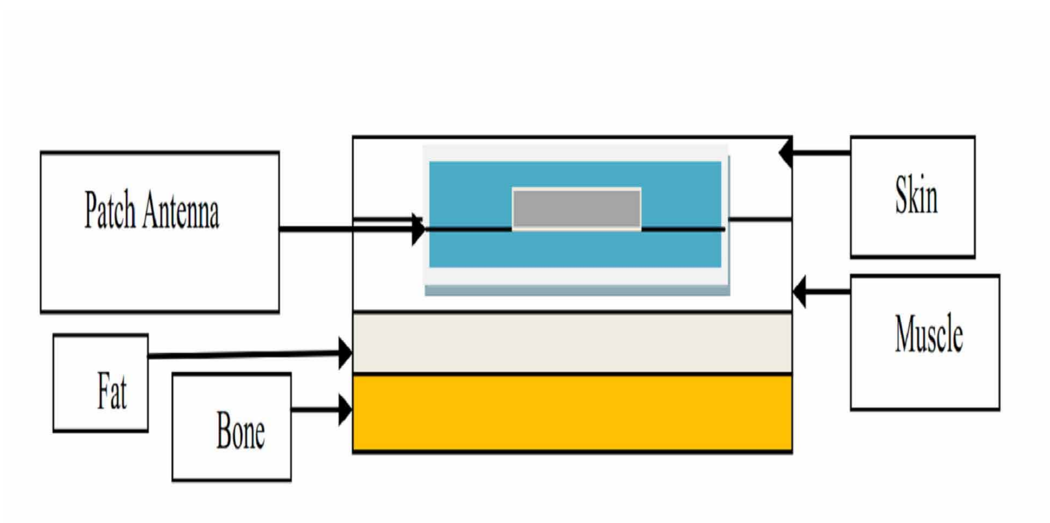


Table 1. Properties of canonical model tissue structure at 402 MHz (S. Gabriel et al., 1996)

Tissue	Permittivity	Conductivity	Loss tangent	Density
Skin	46.74	0.68	0.65	1077
Muscle	57.11	0.79	0.62	1014
Bone	11.62	0.80	0.31	893
Bone	22.43	0.23	0.46	1938

$$\mu = \mu_{\infty} + \frac{\mu_{s} - \mu_{\infty}}{1 + j\text{ÉÄ}} \tag{2}$$

It is called Debye equation which is well known in which μ_{∞} is the permittivity at field frequency where $\text{ÉÄ} \gg 1$, μ_{s} is permittivity at $\text{ÉÄ} \ll 1$ and $j^2 = -1$ (Gabriel et al., 1996)

Hurt in 1985 modeled an expression for dielectric spectrum of muscle to the summation of five Debye dispersions in addition to conductivity term σi

$$\mu_{\text{É}} = \mu_{\infty} + \sum_{n=1}^{5} \left(\frac{\text{”}\mu n}{1 + j\text{ÉÄ}n} + \frac{\sigma i}{j\text{É}\mu 0} \right) \tag{3}$$

An alternate equation is also given by Cole-Cole equation

$$\mu\left(\text{É}\right) = \mu_{\infty} + \frac{\text{”}\mu}{\left(1 + j\text{ÉÄ}\right)^{(1-\pm)}} \tag{4}$$

α is measure of broadening of dispersion. The spectrum of tissue may be described in terms of multiple Cole-Cole dispersion

$$\mu_{\text{É}} = \mu_{\infty} + \sum_{1}^{n} \left(\frac{\text{”}\mu n}{\left(1 + (j\text{ÉÄ}n\right)^{(1-\pm)}} + \frac{\sigma i}{j\text{É}\mu_{0}} \right) \tag{5}$$

By which the dielectric behavior of tissue can be predicted over the desired frequency range.

Based on the above equations the graph of permittivity and conductivity with frequency is plotted as following in Figure 4 (S. Gabriel et al., 1996)

The proposed antenna is for implanting inside the human arm model as Figure 1. Here four layers are considered as skin, muscle, fat and bone and its electrical equivalent is modeled in CST (simulator) environment.

The modeling of electrical equivalent of skin, muscle, fat and bone tissues is according to S. Gabriel model and the graph of permittivity and dispersion of all the tissues with frequency is shown in the Figure 5 (a), (b), (c), (d).

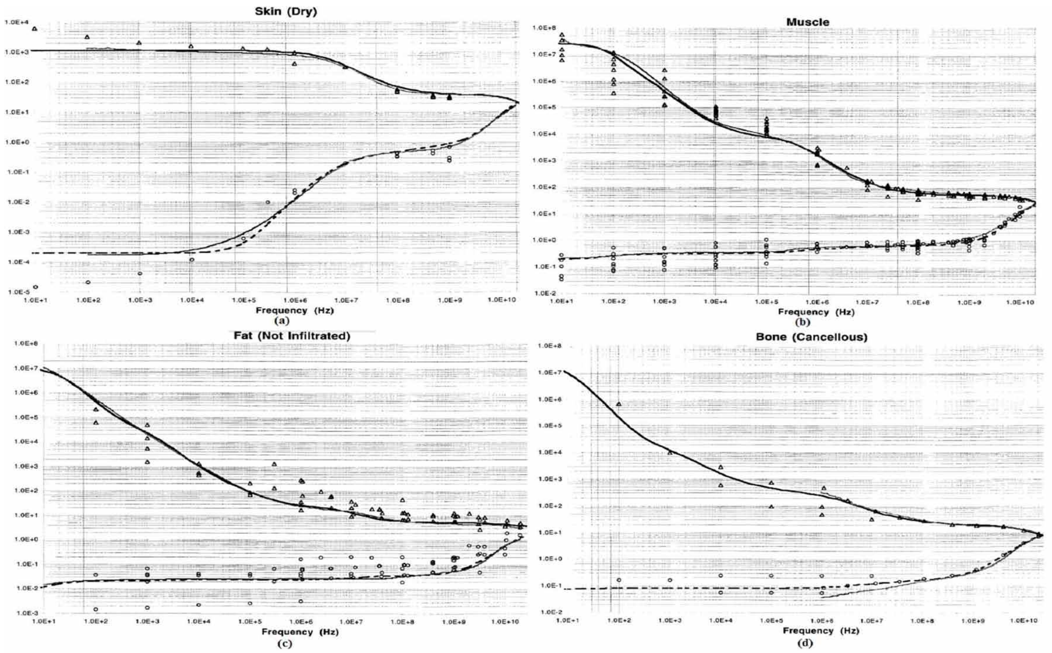
All curves are generated in CST microwave studio environment in frequency band 100 MHz to 500MHz. These graphs are showing similar trends as shown in Figure 4 in Gabriel et al., (1996) literature.

4. ASSESSMENT OF ANTENNA PERFORMANCE

4.1 Reflection Coefficient

Reflection coefficient Vs frequency graph of implanted proposed antenna is shown in Figure 6 which is depicting the impedance bandwidth 23 MHz (413MHz-390MHz). It is showing the maximum reflection coefficient -18.51 at 402 MHz for skin layer only. Figure 7 depicts the reflection coefficient values in different four layers are; -11.47 db in bone layer, -13.08 db in fat layer, -8.68 db in muscle layer and -18.51 db in skin layer.

Figure 4. Permittivity and conductivity of tissues: Prediction of model (Black filled and dotted curves) experimental data at 37° C (grey filled and dotted curves) and data from the literature (triangles and circles) (Gabriel et al., 1996) (a) Skin (b) Muscle (c) Fat (d) Bone



As per the results of different layers in Figure 6, it can be reported that the skin tissue layer is depicting the highest reflection coefficient -18.51db at resonance frequency 402 MHz but other layers are reporting as; muscle tissue layer -16 db at 388 MHz, fat tissue layer -15 db at 408 MHz, bone tissue layer -14 db at 410 MHz. So the best results shown by all layers are of skin layer which is presenting the maximum reflection coefficient at resonant frequency. Proposed antenna is showing the favorable results when it is implanted just below the skin layer and at other layers performance is varying in terms of reflection coefficient and frequency both.

4.2 Directive Gain Performance

The performance of gain in terms of directivity is shown in Figure 8 which is showing 2.5 at 402 MHz. Gain of the proposed antenna is depicting the positive incremental graph up to 500 MHz and electrically the system is stable with such gain value.

4.3 VSWR Performance

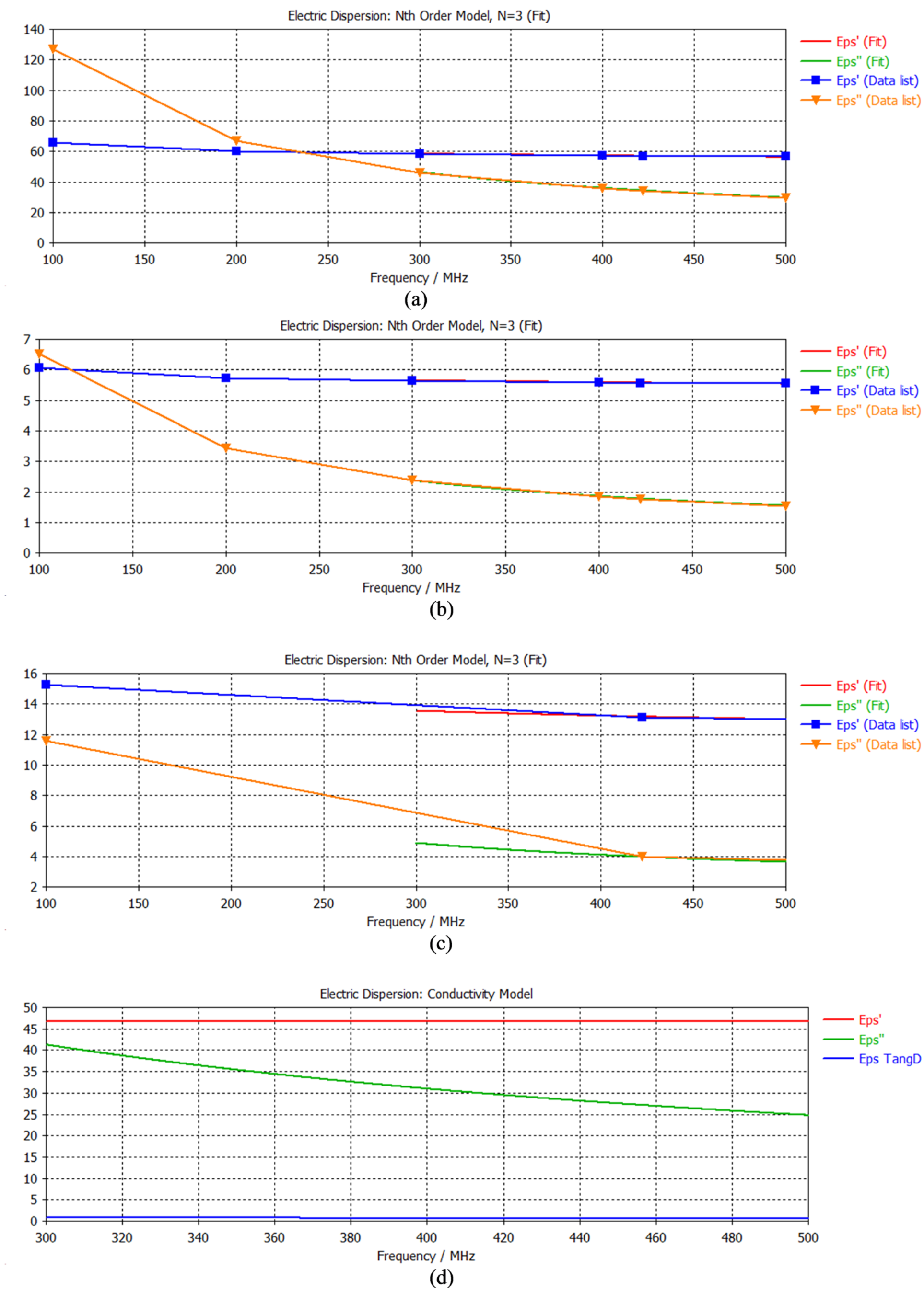
The VSWR is 1.26 at 402 MHz at the frequency of operation as in Figure 9. In the whole range of MICS band (402MHz-405MHz) the value of VSWR is less than 2 which is the favorable result for any radiating antenna.

This graph of VSWR in Figure 9 is justifying the reflection coefficient graph of Figure 6, both are showing the frequency of operation of proposed antenna in MICS band.

4.4 Radiation Pattern

Radiation pattern (3D) is the radiation graph of antenna in all direction. In this measurement the distance for observation of electric field is 1m and in all direction of theta and phi surrounding to the antenna field is simulated.

Figure 5. Dispersion curve of (a) Muscle (b) Fat (c) Bone (d) Skin tissues up to 500 MHz



In Figure 10 the maximum value of field is negative due to the implanted conditions of antenna and lossy material (homogeneous tissue) is attenuating the radiated power and the radiated power at 1 meter is very small. The value of these simulated results are depicting that the maximum field is almost same in frequency band of operation.

Figure 6. Reflection Coefficient in electrical equivalent of homogeneous tissues of human as bone layer, fat layer, muscle layer and skin layer

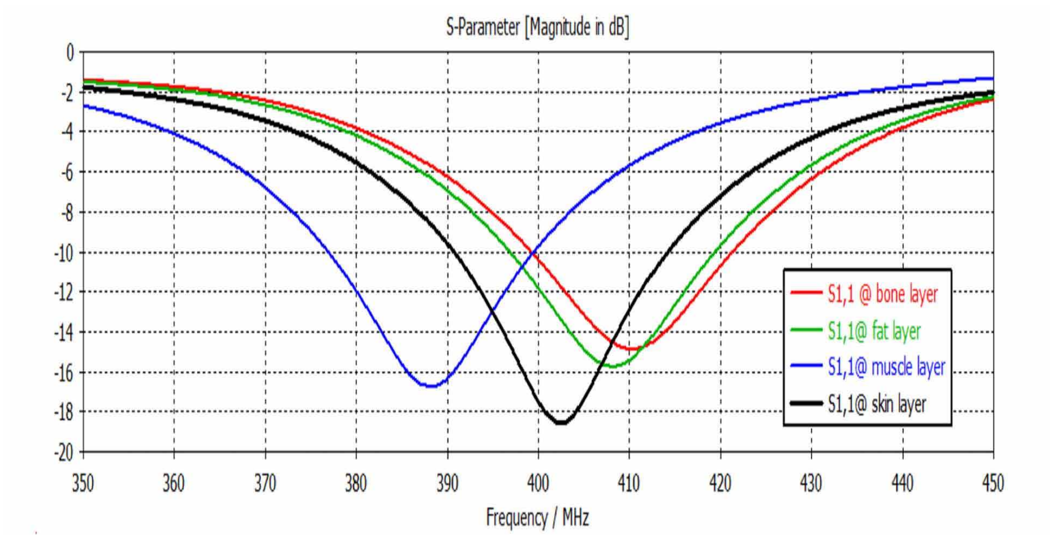
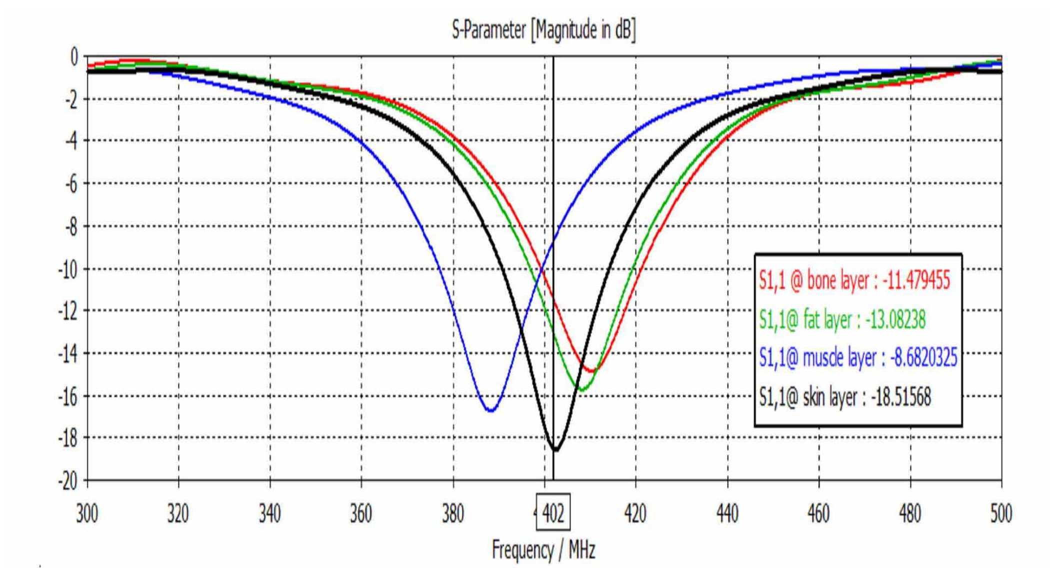


Figure 7. Comparison of reflection coefficient of all homogeneous four layers at 402 MHz frequency



5. CONCLUSION

In this paper multiple homogeneous human tissue models are assessed for meandered stacked implantable patch antenna. Conductivity and permittivity of the surrounded tissues in implanted devices generates restrictions along with the geometrical size of antenna due to which the antenna performance analysis is more complex than the antennas in open environment. A miniaturized meandered stacked patch antenna is implanted in homogeneous skin, muscle, fat and bone tissues and the highest reflection coefficient -18.51db is achieved at 402 MHz. After the comparison of all

Figure 8. Graph of Directivity with frequency

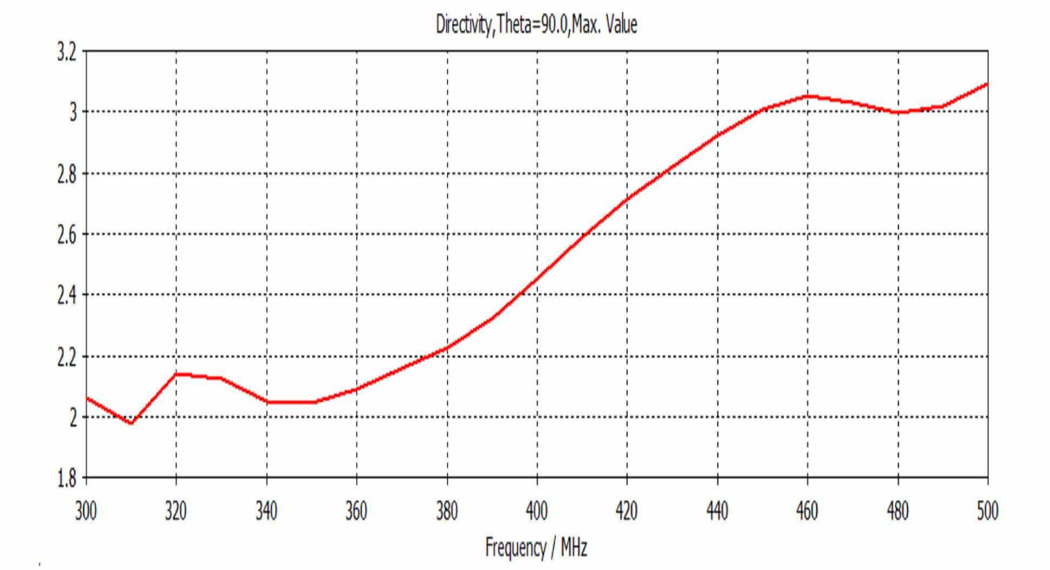
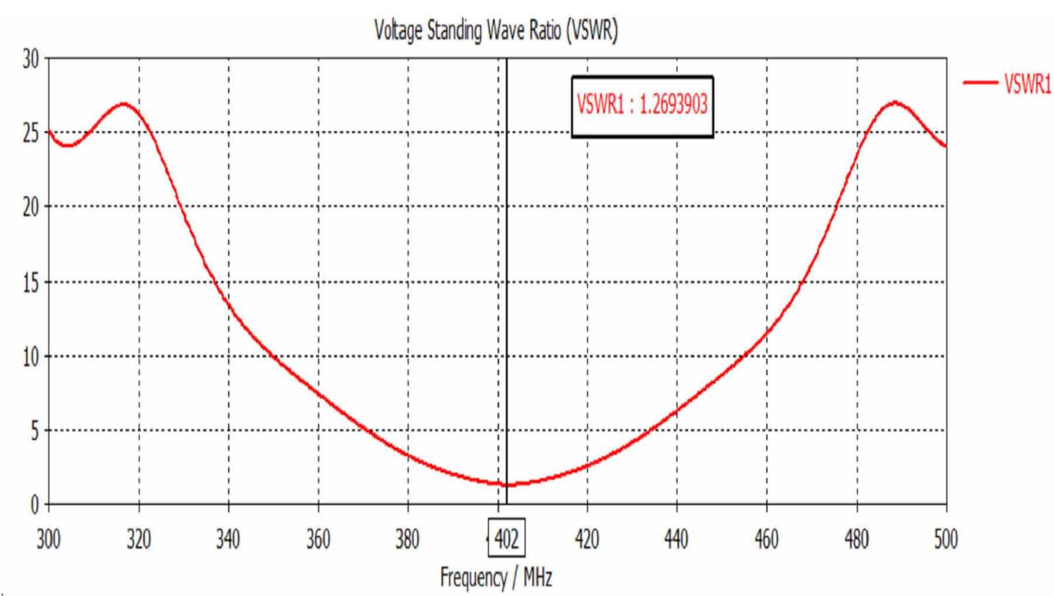
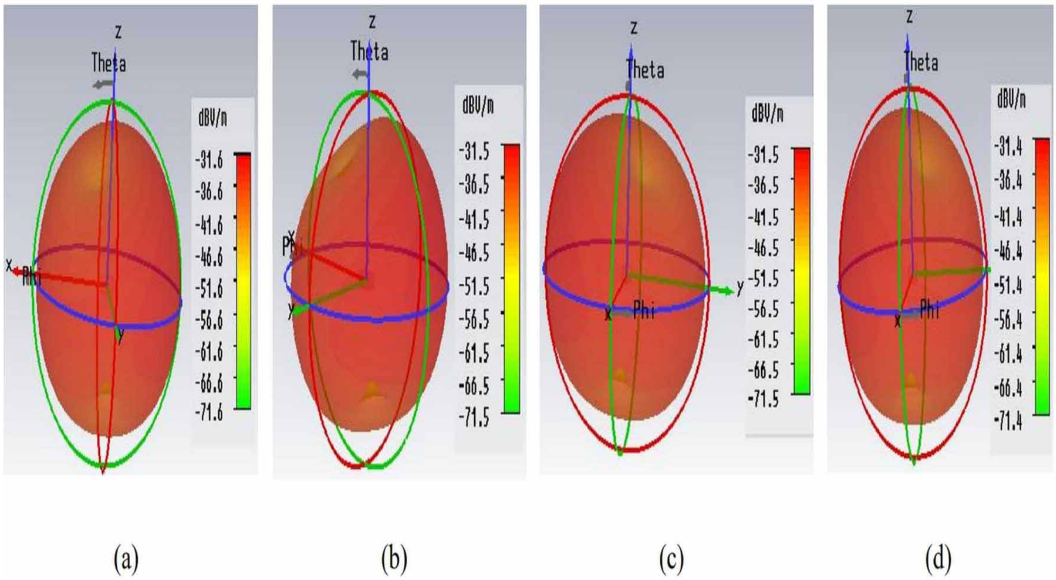


Figure 9. VSWR graph of proposed antenna



reflection coefficient results the skin model is best suited which is achieving 23MHz impedance bandwidth, directivity value 2.5 and 1.26 voltage standing wave ratio. The electrical equivalent of homogeneous tissue models is generated according to S. Gabriel literature and all parameters as directivity and conductivity is simulated in skin tissue at 402 MHz (MICS band).

Figure 10. Radiation pattern (3D) at (a)402 MHz (b) 403 MHz (c) 404 MHz (d) 405 MHz



REFERENCES

Ullah, S., Khan, P., Ullah, N., Saleem, S., Higgins, H., & Kwak, K. S. (2009). A review of wireless body area networks for medical applications. *Int J Commun Netw Syst Sci*, *2*(8), 797–803.

Bahal, I. J. (2003). *Lumped elements for RF and microwave circuits*. Boston: Artech House.

Conway, G. A., & Scanlon, W. G. (2009). Antennas for Over-Body-Surface Communication at 2.45 GHz. *IEEE Transactions on Antennas and Propagation*, *57*(4), 844-855.

Gabriel, C., Sheppard, R. J., & Grant, E. H. (1983). Dielectric properties of ocular tissues at 37 ° C. *Physics in Medicine and Biology*, *28*(1), 43–49. doi:10.1088/0031-9155/28/1/004 PMID:6828557

18. Gabriel, S., Lau, R. W., & Gabriel, C. (1996). The dielectric properties of biological tissues: II. Measurement in the frequency range 10 Hz to 20 GHz. *Physics in Medicine and Biology*, *41*(11), 2251–2269. doi:10.1088/0031-9155/41/11/002 PMID:8938025

Guo, Y.-X., & Shaoqiu, X. C. L. (2007). *A Review of Implantable Antennas for Wireless Biomedical Devices*. Forum for Electromagnetic Research Methods and Application Technologies.

Jain, L., Singh, R., Rawat, S., & Ray, K. (2015). Miniaturized Meandered and stacked MSA using Accelerated design strategy for Biomedical Applications. In *Proceedings of 5th International Conference of Soft Computing for problem solving* (pp. 725-732).

Kamarudin, M. R., Nechayev, Y. I., & Hall, P. S. (2005). Performance of antennas in the on-body environment. In *Proceedings of the 2005 IEEE Antennas and Propagation Society International Symposium* (Vol. 3, pp. 475-478). IEEE.

Kiourti, A., & Nikita, K. S. (2012). Miniature implantable antennas for biomedical telemetry: From Simulation to realization. *IEEE Transactions on Biomedical Engineering*, *59*(11), 3140–3147. http://niremf.ifac.cnr.it. doi:10.1109/TBME.2012.2202659 PMID:22692865

Liu, H.-W., Ku, C.-H., & Yang, C.-F. (2010). Novel CPW-fed planar monopole antenna for WiMAX/WLAN applications. *IEEE Antennas and Wireless Propagation Letters*, *9*, 240–243. doi:10.1109/LAWP.2010.2044860

Ouyang, Y., Karayianni, E., & Chappell, W. J. (2005). Effect of fabric patterns on electrotextile patch antennas. In *Proceedings of the 2005 IEEE Antennas and Propagation Society International Symposium* (Vol. 2, pp. 246-249). IEEE..

Salonen, P., Kim, J., & Rahmat-Samii, Y. (2005). Dual-band E-shaped patch wearable textile antenna. In *Proceedings of the 2005 IEEE Antennas and propagation society international symposium* (Vol. 1, pp. 466-469). IEEE.

Sim, C. Y. D., Cai, F. R., & Hsieh, Y. P. (2011). Multiband slot-ring antenna with single-and dual-capacitive coupled patch for wireless local area network/worldwide interoperability for microwave access operation. *IET Microwaves, Antennas & Propagation*, *5*(15), 1830–1835. doi:10.1049/iet-map.2010.0462

20. Singh, R., Rawat, S., Ray, K., & Jain, L. (2015). Performance of Wideband Falcate Implantable Patch Antenna for Biomedical telemetry. In *Proceedings of 5th International Conference of Soft Computing for problem solving* (pp. 757-765).

Tanaka, M., & Jang, J.-H. (2003). Wearable microstrip antenna. *IEEE Antennas and Propagation Int Symp*, *2*, 704–707.

Terence, S. P., & Chen, Z. N. (2005). Effects of human body on performance of wearable PIFAs and RF transmission. In *Proceedings of the 2005 IEEE Antennas and Propagation Society International Symposium* (Vol. 1, pp. 686-689). IEEE..

Wang, P., Wen, G.-J., Huang, Y.-J., & Sun, Y.-H. (2012). Compact CPW-fed planar monopole antenna with distinct triple bands for WiFi/WiMAX applications. *Electronics Letters*, *48*(7), 357–359. doi:10.1049/el.2011.3692

Yazdandoost, K. Y. (2012). UWB antenna for body implanted applications. In proceedings of 42nd European Microwave Conference, October 29- November 1, Amsterdam, The Netherlands.

Zhu, S., & Langley, R. (2007). Dual-band wearable antennas over EBG substrate. *Electronics Letters*, *43*(3), 141–142. doi:10.1049/el:20073151

Kanad Ray is a Professor and Head of Physics at the Amity School of Applied Sciences Physics Amity University Rajasthan (AUR), Jaipur, India. He has obtained MSc & PhD degrees in Physics from Calcutta University & Jadavpur University, West Bengal, India. In an academic career spanning over 22 years, he has published and presented research papers in several national and international journals and conferences in India and abroad. Some of his papers have more than 5 Impact Factor. He has authored a book on the Electromagnetic Field Theory. Prof. Ray's current research areas of interest include cognition, communication, electromagnetic field theory, antenna & wave propagation, microwave, computational biology, and applied physics. Presently he is guiding 8 PhD scholars in various interdisciplinary fields. He has been serving as Editor of different Springer Book Series. Presently he is an Associated Editor of Journal of Integrative Neuroscience published by IOS Press, Netherlands. He has travelled more than a dozen countries on different academic missions. He has established a MOU between his University and University of Montreal, Canada for various joint research activities. He has also established collaboration with National Institute for Materials Science (NIMS), Japan for joint research activities and visits MIMS as a visiting Scientist. He organizes international conference series such as SoCTA, ICOEVCI as General Chair. He is an Executive Committee Member of IEEE Rajasthan

Raghvendra Singh is pursuing a PhD from J.K. Lakshmipat University, Jaipur, India. He received Master of Technology in Electronics and Communication from MNIT, Jaipur, Rajasthan, India in 2011 and a BTech. in Electronics and Communication from UPTU, Lucknow, U.P., India in 2004. His area of expertise is Design and analysis of microstrip patch antennas for body centric wireless communication and implanted applications.

Dr. Dambarudhar Seth is an Associate Professor in Electronics and Communication Engineering. He did his M.Tech from Indian Institute of Information Technology, Allahabad with specialialization Wireless Communication& Computing and Ph.D (Engg) from Birla Institute of Technology, Mesra, Ranchi. He has more than 20 years of academic experience. Prior to joining JKLU, Jaipur he had been with KIIT University, Bhubaneswar as Associate Professor and also with Birla Institute of Technology, Mesra, Ranchi as Senior Lecturer. Ha has guided 23 M.Tech students for their thesis, specifically in the area of Cognitive Radio Network, Mobile Ad-hoc Network, Wireless Sensor Network etc. He has more than 20 publications which includes 6 publication in international journal (SCI-indexed & peer reviewed), 11 number of publications in international conferences mostly IEEE. He has also presented paper at different IEEE conferences held at IIIT Allahabad, Amity University, Noida, Park Institute of Technology Coimbatore etc. Dr. Seth is an active IEEE member and also involved in activities of IEEE Communication Society & IEEE Vehicular Technology as Member.

International Journal of Applied Evolutionary Computation
Volume 9 • Issue 1 • January-March 2018

Sentiment Analysis of Social Networking Websites using Gravitational Search Optimization Algorithm

Lavika Goel, Department of Computer Science and Information Systems, Birla Institute of Technology and Science (BITS), Pilani, Rajasthan, India

Anubhav Garg, Birla Institute of Technology and Science (BITS), Pilani, India

ABSTRACT

Analysing sentiments of various online communities have become now an interesting topic of research and industry. The behaviour of online communities resembles that of a swarm. This article presents a Gravitational Search algorithmic approach for sentiment analysis of online communities, and an optimization algorithm which is based on the mass interactions and law of gravity. In the end, the authors present comparisons with other techniques, particularly ant colony optimization and Naive Bayes classification for sentiment analysis.

KEYWORDS

Ant Colony Optimization Metaheuristic, Gravitational Search Algorithm, Sentiment Analysis, Swarm Intelligence

1. INTRODUCTION

The Internet has changed the lifestyle of many people, have disrupted many industries, have changed the way people do shopping, the way they communicate and much more. Since the evolution of the social network, people have got a lot of freedom to express their opinions about themselves as well as about other people, objects and events. Given a piece of information, people by becoming members of these social network and through its mode of communication and interaction, form up a single opinion about some posting of an event, object, etc. A topic of interest has emerged nowadays due to this social swarming behavior of people which is sentiment analysis, which means that to systematically identify, extract, quantify, and study information.

Sentiment analysis can help us to understand the behaviour of online communities. It has importance in many areas, such as society, security, travel, finance corporate, medical, entertainment and many more. For example, in this information age, a lot of internet companies use sentiment analysis to adjust their marketing strategies, develop product quality, etc. We have used Gravitational Search Algorithm in such a way so to resemble that to members of a swarm and predict their opinions. GSA is based on the mass interactions and law of gravity. In the algorithm we propose for our application, the masses interact with each other based on the Newtonian gravity and the laws of motion. After this, we have compared our results with that of swarm intelligence algorithm, namely Ant Colony Optimization and Naive Bayes classification of opinions.

DOI: 10.4018/IJAEC.2018010105

2. LITERATURE SURVEY

The study of content and analyzing it has a long history, dating back to 1966. Analysing reviews is different from that of analyzing posts as negative or positive. Tweets from Twitter has been a source of data for many researchers. In the past, sentiment analysis and opinion mining were used as synonyms. Das and Chen (2007) used the term first in the paper on market sentiment analysis. In 2002, ACL published a paper in which the term appeared.

Nasukawa et al. (2003) paper, "Sentiment analysis: Capturing favorability using natural language processing", also a paper named "Sentiment Analyzer: Extracting sentiments about a given topic using natural language processing techniques." Some researchers use the term most commonly to predict the polarity of a given piece of text, thereby referring the phrase for this task only. However, the term evolved more broadly to mean the subjectivity in the text and computational treatment of opinion, sentiment.

Go et al. (2009) used distant supervision in their paper on Twitter sentiment analysis. They have shown using Naive Bayes, Maximum entropy and SVM and achieved accuracy up to 80%.

Kun-Lin Liu et al. (2013) in their paper on Twitter sentiment analysis integrated noisy labelled data for training using emoticon smoothed language model (ESLAM) and used this for the unsupervised machine learning problem of classification.

Pak and Paroubek (2010) in their paper suggested a method for collecting corpus automatically from microblogs and build a sentiment classifier from it. In this instance, Twitter is used to gather corpus. The authors have used only English language, but their claim is that it can be used for multiple languages.

Stylios et al. (2014) first used the swarm intelligence algorithms in their work and compared the results with that of traditional methods like decision trees for evaluating the polarity sentences. Their result showed drastic improvements as compared to traditional methods.

Tumasjan et al. (2010) examined 100,000 tweets for election results. The highlight of their paper was that forty percent of the messages were posted by only four percent of the subscribers.

Goel (2011) used Ant Colony Optimization as a swarm intelligence algorithm for sentiment analysis, modelling it as a problem of social swarming resembling with that of ants.

Vanzo et al. (2014) used SVMhmm algorithm, Markovian formulation of the SVM discriminative model for sentiment polarity assessment to entire sequences.

Basari et al. (2013) published a paper in 2013 on opinion mining of movie reviews using hybrid methods of PSO and SVM. They classified the problem into two classes, positive and negative, based on SVM and then used PSO to optimize the solution. They improved the results from 71% to 77%.

Agarwal et al. (2011) in their paper on the analysis of Twitter posts used introduced new features such as POS selection, tree kernel and improved upon the already existing mechanism for the same.

3. METHODOLOGY

In this section, we describe the algorithm, its data collection and pre-processing etc.

3.1. Data Collection

The data from Twitter was collected using Twitter API. Twitter data has the following format:

"User_id", "id_str", "created_at", "favourite_count", "retweet_count", "followers_count", "text". The definitions and its meanings, as well as their values, is available on the information page of Twitter's API.

Praw API for Reddit was used to collect reddit data. Reddit data is in the form of a plain string containing the post with no user information.

3.2. Pre-Processing

Firstly, we do tokenization of the comments into words. Then we remove all the emoticons and non-Ascii characters from those words. All the citations and links are also removed because they do not add any sentiment. Stop words, which are common words like 'and' are also removed as they do not add any meaning to our evaluation.

3.3. Algorithm

We divided the dataset into 10 subsets. The algorithm is trained on 9 of these and it is tested on the tenth dataset. The method used to do this division is simple- add every tenth record to a subset and so on. In the first iteration, the first record of every subset is used for evaluation, in the second iteration the second record of all subsets and so on.

In training, we have created two mass arrays M(p) and M(n) respectively for positive mass values and negative mass values, of equal length. We could have also used a variable instead of an array, but arrays are just used for convenience. The initial value which is assigned is some constant.

First of all, we calculate the mass m of a post by counting all the positive words in it divided by the total number of words, i.e., m = positive words in post/ total words in the post. Now, we calculate the force by Newton's gravitational formula: $F = \frac{GmM}{R}$ where R has a constant value. Calculate the force due to these masses M(p) and M(n), and calculate d. Id d>0 update M(p), i.e., Mass M(p) is increased if this difference comes out to be positive else update M(n).

In testing, we don't update the values in the array but just use their values to predict the polarity of a post. If the prediction from the algorithm and the actual polarity calculated by NLP techniques do not match, then we increment the value of the incorrect variable, otherwise the correct variable. Mass values M(p) and M(n) are also decremented after the iteration to let the system come to an equilibrium. This has an analogy with the behaviour of social communities that people reads recent comments and forgets the old viewpoints and opinions. If values decrement too much as below a threshold, then we reset all the values. This does not affect the entire mechanism but just to prevent values to become very low and avoid unclarity. On the next page is the pseudo code of the algorithm just explained in Box 1 and the flowchart in Figure 1.

3.4. Results

Twitter data was collected using the Twitter API and contains post on Valentine's Day, i.e, 14th November. about 2000 records of which we used nine out of ten for training and one was used for testing.

Reddit data was collected using the Praw API from Github over a period of two days in which different genre's thread was collected. Also, nine out of ten for training and one was used for testing.

For Twitter dataset, the algorithm predicted 1868 records correctly and 130 records incorrectly leading to the accuracy of 93.5%. For Reddit, 3116 were correctly predicted while 1107 records were incorrectly predicted by the algorithm (see Table 1).

The same dataset was used for comparing with the Naive Bayes classification for sentiment analysis and also for ant colony optimization. The results of these two techniques are given in table 2 and 3 respectively.

Figure 2 shows the graph of mass values of posts calculated as explained in the algorithm versus the number of posts in 1000. Figure 3 shows the same with a number of posts with a gap of around 200. From these graphs, we can see that the frequency of polar words are less as compared to total words. This can be understood by the example that people use words such as 'fantastic' only once as the frequency will not change the polarity of the posts. Figure 4 and 5 shows the graphs for ACO algorithm.

Box 1. Pseudo code of the algorithm

Pre-processing
Tokenize all posts in words, remove all prefixes or suffixes (stemming), remove words that are duplicate, remove emoticons, acronyms etc.
TRAINING
Iterate over the records of the subsets of the dataset. Repeat-
a. Calculate mass of post, m.
b. Calculate Force due to M(p) and M(n) .
c. Evaluate difference d.
d. If d is greater than 0.
i. Update mass value M(p).
e. Else if d is less than 0.
i. Update mass value M(n).
f. Decrease mass to account for evaporation.
g. If the values too much then reset all values.
TESTING
Iterate over the records of the 1oth subset of the dataset.
Repeat-
a. Caculate d = Force(M(p)) - Force(M(n))
b. Evaluate polarity of post using NLP.
c. If post is carrying sentiment
i. Increase the value of correct if prediction matches
ii. Else increment incorrect

4. CONCLUSION AND FUTURE SCOPE

The use of optimization algorithms is a new and emerging field of research. As sentiment analysis has become a hot topic of research among the scientific community, new methods are being developed. Our algorithm gives better results as compared to previously used methods such as Naïve Bayes' classification and Ant Colony Optimization. The proposed work can be further extended with other algorithms and their hybrids and the results could possibly improve. In the future we hope to incorporate other strategies like Bat Algorithm, Rain Water Algorithm, etc., to further improve our results.

Figure 1. Pseudo Code depicted in the form of a simple flow chart

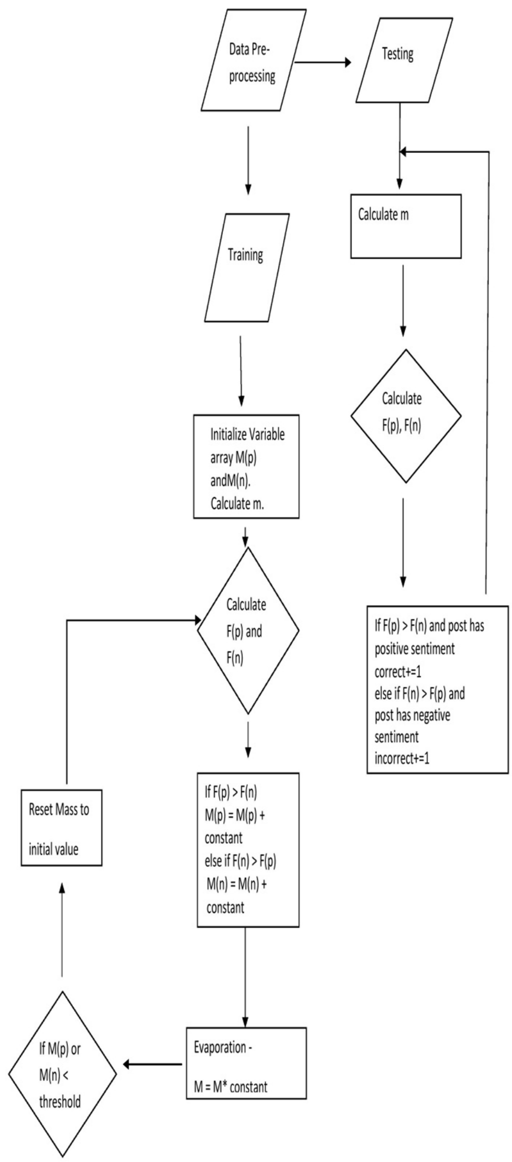


Table 1. Showing the results with GSA

Dataset	Correct Predictions	Incorrect Predictions	Accuracy
Twitter	1868	130	93.5%
Reddit	3116	1107	73.8%

Table 2. Showing the results with Naive Bayes classifier

Dataset	Correct Predictions	Incorrect Predictions	Accuracy
Twitter	1658	340	83%
Reddit	2820	1403	66.8%

Table 3. Showing the results with ACO

Dataset	Correct Predictions	Incorrect Predictions	Accuracy
Twitter	1799	199	90.04%
Reddit	3061	1162	72.6%

Figure 2. Mass m values for posts(GSA)

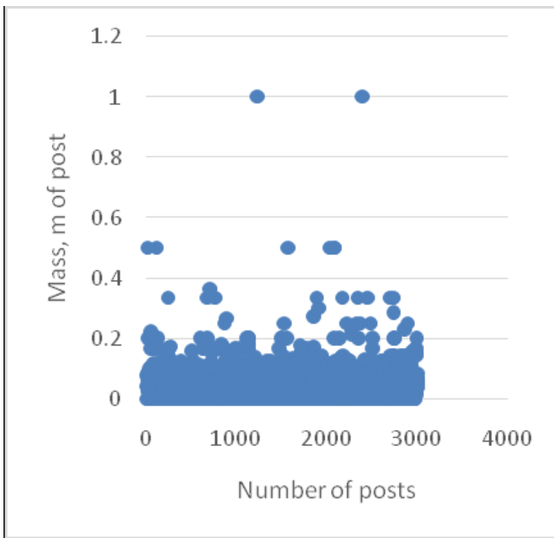


Figure 3. Mass m values for posts(GSA)

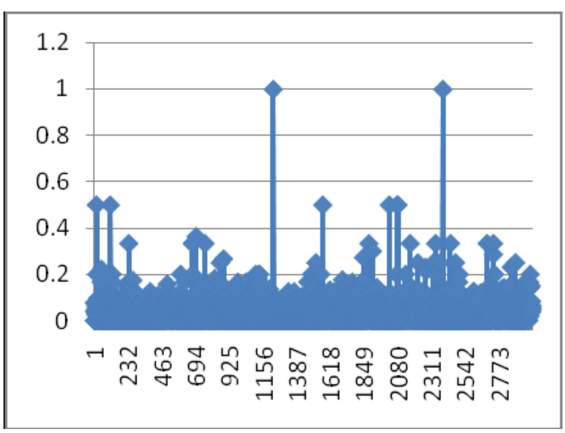


Figure 4. Positive heuristic for first few posts(ACO)

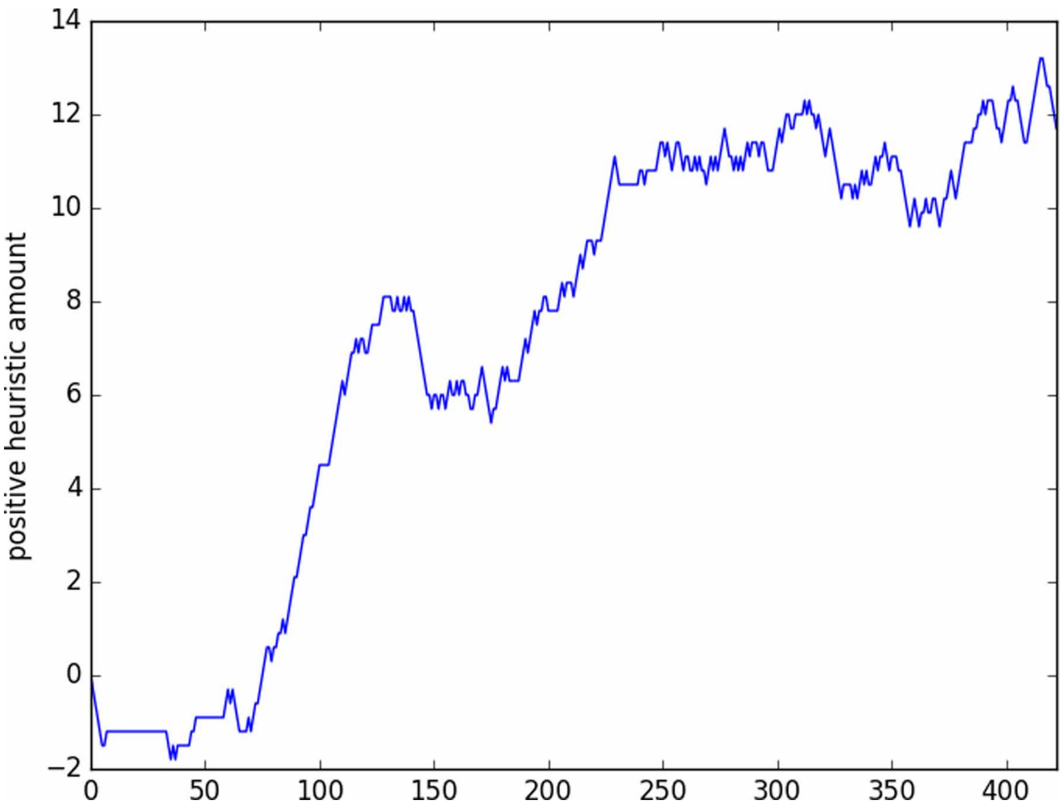
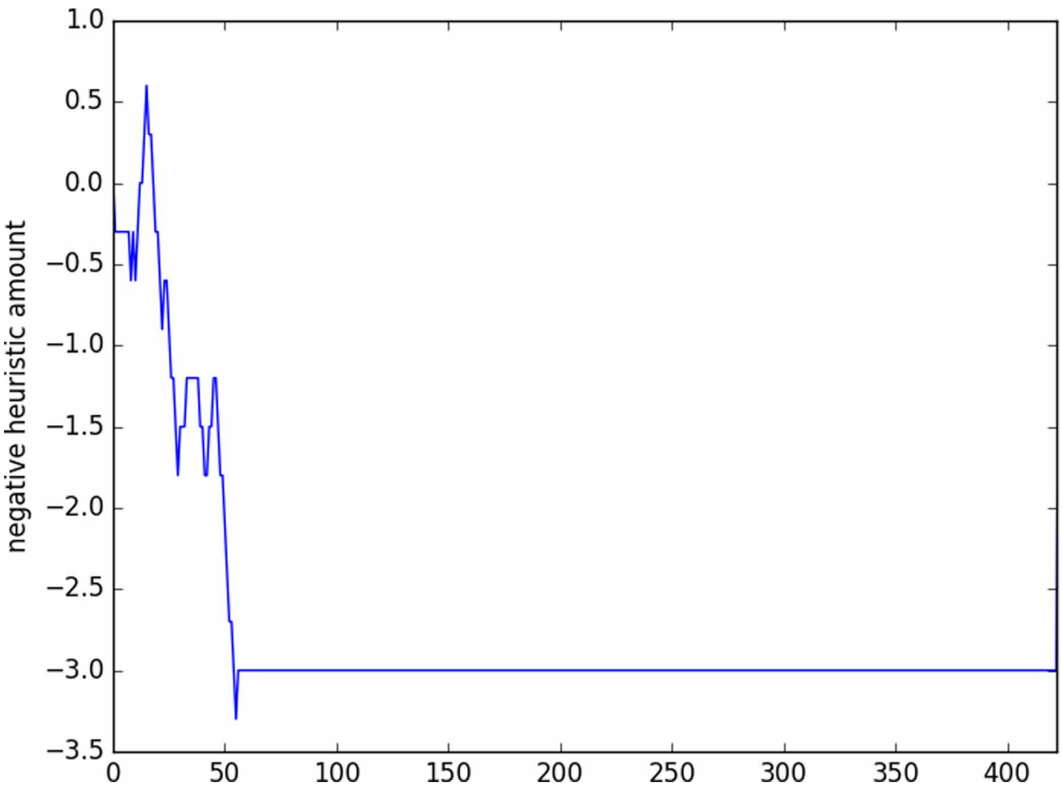


Figure 5. Negative heuristic for first few posts(ACO)



REFERENCES

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011, June). Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media* (pp. 30-38). Association for Computational Linguistics.

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011, June). Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media* (pp. 30-38). Association for Computational Linguistics.

Basari, A. S. H., Hussin, B., Ananta, I. G. P., & Zeniarja, J. (2013). Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization. *Procedia Engineering*, *53*, 453–462. doi:10.1016/j.proeng.2013.02.059

Basari, A. S. H., Hussin, B., Ananta, I. G. P., & Zeniarja, J. (2013). Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization. *Procedia Engineering*, *53*, 453–462. doi:10.1016/j.proeng.2013.02.059

Ghiassi, M., Skinner, J., & Zimbra, D. (2013). Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with Applications*, *40*(16), 6266–6282. doi:10.1016/j.eswa.2013.05.057

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision (CS224N Project Report).

Goel, L., Gupta, D., & Panchal, V. K. (2012, December). Biogeography based anticipatory computing framework for intelligent battle field planning. In *Proceedings of the 2012 12th International Conference on Hybrid Intelligent Systems (HIS)* (pp. 41-46). IEEE. 10.1109/HIS.2012.6421306

Goel, L., Gupta, D., & Panchal, V. K. (2012). Hybrid bio-inspired techniques for land cover feature extraction: A remote sensing perspective. *Applied Soft Computing*, *12*(2), 832–849. doi:10.1016/j.asoc.2011.10.006

Goel, L., Gupta, D., & Panchal, V. K. (2015). Two-phase anticipatory system design based on extended species abundance model of biogeography for intelligent battlefield preparation. *Knowledge-Based Systems*, *89*, 420–445. doi:10.1016/j.knosys.2015.07.023

Goel, L., Panchal, V. K., Gupta, D., & Bhola, R. (2010, February). Hybrid ACO-BBO Approach for predicting the deployment strategies of enemy troops in a military terrain application. In *Proceedings of the 4th International Multi-Conference on Intelligent Systems & Nanotechnology (IISN-2010)* (pp. 118-122).

Hu, X., Tang, J., Gao, H., & Liu, H. (2013, May). Unsupervised sentiment analysis with emotional signals. In *Proceedings of the 22nd international conference on World Wide Web* (pp. 607–618). ACM; . doi:10.1145/2488388.2488442

Kumar, A., & Sebastian, T. M. (2012). Sentiment analysis on twitter. *IJCSI International Journal of Computer Science Issues*, *9*(3), 372–378.

Nasukawa, T., & Yi, J. (2003, October). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture* (pp. 70–77). ACM; . doi:10.1145/945645.945658

Pak, A., & Paroubek, P. (2010, May). Twitter as a corpus for sentiment analysis and opinion mining. *LREc,* 10.

Stylios, G., Katsis, C. D., & Christodoulakis, D. (2014). Using Bio-inspired intelligence for Web opinion Mining. *International Journal of Computers and Applications*, *87*(5).

Thelwall, M., Buckley, K., & Paltoglou, G. (2011). Sentiment in Twitter events. *Journal of the Association for Information Science and Technology*, *62*(2), 406–418. doi:10.1002/asi.21462

Tumasjan, A., Sprenger, T. O., Sandner, P. G., & Welpe, I. M. (2010). Predicting elections with twitter: What 140 characters reveal about political sentiment. *ICWSM*, *10*(1), 178–185.

Vanzo, A., Castellucci, G., Croce, D., & Basili, R. (2014). A context based model for sentiment analysis in twitter for the italian language. In *Proceedings of the First Italian Conference on Computational Linguistics CLiC-it & the Fourth International Workshop EVALITA* (pp. 379-383).

Lavika Goel is currently an Assistant Professor in the Department of Computer Science & Information Systems at BITS, Pilani. She had earlier worked as an Assistant Professor at Delhi Technological University (DTU), and also holds corporate experience working at Oracle India Private Ltd. She is PhD, ME and B.Tech with distinction in Computer Science & Engineering from Delhi Technological University (formerly Delhi College of Engineering) and holds a membership of Machine Intelligence & Research (MIR) Labs. She has always been among the top rank holders of the university – 2nd ranked holder in ME, and 1st ranked holder in BTech batch. She was also the gold medalist in Mathematics (100% marks) and Chemistry in 12th class (ISC board). During her ME, she was offered six months internship from ST Microelectronics and a dual campus recruitment by ORACLE India Pvt. Ltd. as well as ARICENT Technologies. During her BTech, she was offered recruitment by HCL Technologies as a part of BTech Campus Recruitment Program and had also cleared the MICROSOFT Off-Campus Recruitment Written Examination. She secured a percentile of 98.7 and an ALL INDIA RANK of 242 in GATE, 2008. She qualified for the National Eligibility Test (NET) held by CBSE-UGC on June 28th 2015 for eligibility for Assistant Professor in central organizations. She serves as a reviewer of recognition to many reputed international journals including the Elsevier Journal of Information Sciences and Applied Soft Computing. She has published around 40 research papers in SCI indexed International conferences, journals and book chapters including Elsevier, IEEE, Taylor and Francis, Springer, etc.

Anubhav Garg is currently a 4th-year undergraduate student pursuing Dual Majors in B.E. Computer Science and M.Sc. Physics at Birla Institute of Technology and Science, Pilani, India. During his stay at BITS Pilani, he has been involved in various academic projects and internships as well as extra-curricular activities.

International Journal of Applied Evolutionary Computation

Volume 9 • Issue 1 • January-March 2018 • ISSN: 1942-3594 • eISSN: 1942-3608
An official publication of the Information Resources Management Association

MISSION

The **International Journal of Applied Evolutionary Computation (IJAEC)** presents comprehensive interdisciplinary and refereed research on the most emerging and breakthrough areas of Evolutionary Computation (EC) and applications. The journal expands existing research findings (theoretical innovations and modeling applications) to provide the highest quality original concepts, hybrid applications, innovative methodologies, and development trends studies for all audiences. The primary mission of IJAEC is to publish original/state-of-the-art/highest quality research articles in application and convergence of the areas of evolutionary computation and other similar techniques to address real world complexities. Evolutionary computation theory, practice and applications are the essential topics being covered. The journal succeeds in expanding the fields and depths of the most principal and critical concepts forming a more matured research society. The journal is targeted at both academic researchers and practicing EC professionals.

SUBSCRIPTION INFORMATION

IJAEC is published Quarterly: January-March; April-June; July-September; October-December by IGI Global. Full subscription information may be found at www.igi-global.com/IJAEC. The journal is available in print and electronic formats.
Institutions may also purchase a site license providing access to the full IGI Global journal collection featuring more than 100 topical journals in information/computer science and technology applied to business & public administration, engineering, education, medical & healthcare, and social science. For information visit www.igi-global.com/isj or contact IGI at eresources@igi-global.com.

CORRESPONDENCE AND QUESTIONS

EDITORIAL

Wei-Chiang Samuelson Hong, Editor-in-Chief • IJAEC@igi-global.com

SUBSCRIBER INFO

IGI Global • Customer Service
701 East Chocolate Avenue • Hershey PA 17033-1240, USA
Telephone: 717/533-8845 x100 • **E-Mail:** cust@igi-global.com

The *International Journal of Applied Evolutionary Computation* is indexed or listed in the following.

ACM Digital Library; Bacon's Media Directory; Cabell's Directories; DBLP; Google Scholar; JournalTOCs; MediaFinder; The Standard Periodical Directory; Ulrich's Periodicals Directory