A Generalized Island Model Based on Parallel and Cooperating Metaheuristics for Effective Large Capacitated Vehicle Routing Problem Solving

Meryem Ammi and Salim Chikhi

MISC laboratory, Computer Science Department, College of NTIC, Constantine2 University, Constantine, Algeria

Capacitated Vehicle Routing Problem (CVRP) is among transportation problems that are of the foremost concerns in logistics. Ensuring an effective product distribution over a large distribution network while reducing the required costs represents the scope of the present work. A synergic and interactive environment of parallel meta-heuristics is developed using a generalized island model to deal with large instances of CVRP. In the proposed model, cooperative meta-heuristics, namely genetic algorithms (GA) and ant colony optimization algorithms (ACO), are organized into archipelagoes. They communicate synchronously, globally and locally by exchanging solutions. In order to handle properly the migration of solutions, either between archipelagoes or between islands within the same archipelago, appropriate selection and replacement policies are adopted. Furthermore, the proposed approach uses other new features including a new binary solution representation and different optimization process (i.e GA, ACO) on each island. To prove the efficiency of the present work, tests over the well-known set of benchmarks, comparative studies and experimental analysis have been conducted.

Keywords: cooperative meta-heuristics, parallel meta-heuristics, vehicle routing problems, genetic algorithms, ant colony optimization

1. Introduction

Today, transportation activities constitute the vital link between suppliers and customers. The economical and environmental costs resulting from such activities are extremely important and need to be intelligently managed. The Capacitated Vehicle Routing Problem (CVRP) is considered as the basic variant of the Vehicle Routing Problem (VRP) problems class. As a VRP, it involves minimizing the cost of shipping goods from suppliers to customers by finding the efficient use of a fleet of vehicles used for the delivery operations [20]. In CVRP, each vehicle has limited capacity and every customer has a known demand. Having a cost to each route, the objective is to minimize the total cost of serving all customers. Its instances counting more than 100 customers are considered large and subsequently it becomes intractable and difficult to solve.

Knowing the nature of the CVRP being an NP-hard problem [11], exact methods remain an unlikely choice to deal with, particularly when manipulating its large instances. Exact methods for CVRP solving have been largely investigated in the literature, for more insight into exact methods, interested reader is referred to [42].

Recently, most approaches developed are based on heuristics. These are approximate algorithms that try to find good feasible solutions in reasonable computing time. They can be divided into two main classes; classical heuristics and meta-heuristics. In [22] there is an extensive overview of the most remarkable works proposed to solve the VRP in the literature. However, using these methods requires a reasonable computing time and guided steering search to overcome the stuck in local minima.

By another way, the availability of computing resources has made possible parallelism of meta-heuristics. Parallel meta-heuristics is a new trend that has attracted many researchers

and proved its efficiency in gaining computational time and enhancing the solutions quality [12]. The aim of the parallel strategies is mainly to speed up the search process and also to improve the quality of the obtained solutions. A survey on the different parallel meta-heuristics strategies can be found in [13]. In line with this, among the models that have exploited the concept of the parallelism as a practical option, we found Generalized Island Model (GIM). It is a new paradigm in which the optimization algorithms are distributed and cooperating over multiple islands through the migration operator [25]. This interaction leads to a remarkable gain of computing time, exploiting the search space and exploring the different regions by exchanging solutions.

The emphasis of this work is solving the Large Capacitated Vehicle Routing Problem (LCVRP) by using a generalized island model (GIM) based on genetic algorithms and ant colony optimization algorithms as cooperative meta-heuristics. Consequently, the used meta-heuristics are organized into archipelagoes where each archipelago represents an island model. Indeed, each island performs an optimization process. On one hand, the solutions selected through a selection policy are sent to the islands within the same archipelago via a synchronous inter-communication. On the other hand, the archipelagoes exchange solutions through an asynchronous intra-communication. The tests over the LCVRP instances have been carried out and the computational scores have been reported.

The remainder of this paper is organized as follows: The CVRP problem formulation and some basic concepts are described in Section 2. In Section 3, we present the related works. Continuously, in Section 4 we explain the proposed approach, named CPGA including all the used features. The sum of the findings and discussion about them are put in Section 5. Finally, the conclusion and outlooks are summarized in Section 6.

2. Capacitated Vehicle Routing Problem Formulation

Let $G = (V, A)$ be an undirected weighted graph where vertices in $V = 0 \dots n$ represent cities and edges in A represent roadways between pairs of cities. Every arc (i, j) is associated a non-negative weight $Dist_{i,j}$ that denotes the distance between the cities. Each city (customer) has a demand $d_i \geq 0$. Assuming the availability of m vehicles each of capacity Q based at the depot, solving the CVRP consists of designing a set of least-cost vehicle routes in such a way: (Figure 1 illustrates a possible solution for a CVRP problem.)

1. Each customer is visited exactly once by one route,

2. Each route starts and ends at the depot and

3. The total demand of the customers served by a route does not exceed the vehicle capacity Q
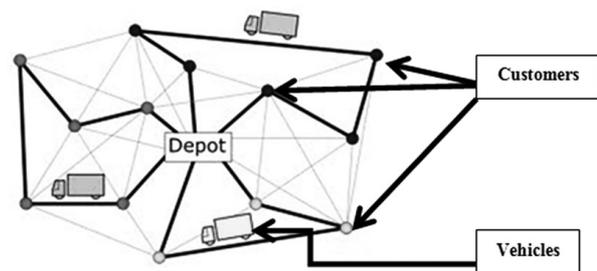
Figure 1. Illustration of a Vehicle Routing Problem.

The following notations are used to formulate the CVRP model:

- n: is the number of customers

- m: is the number of vehicles;

- d_i: the demand of the customer i;

- $Dist_{ij}$: the distance cost between customer i and customer j;

- Q: the capacity of vehicle;

- y_{ik}: binary variable: its value is 1 if the customer i is delivered by the vehicle k; otherwise it is 0;

- x_{ijk}: binary variable: its value is 1 if the vehicle k travels directly from customer i to customer j, otherwise it is 0.

The CVRP aims at minimizing the total distance required to serve all the customers. Therefore, the objective function can be expressed as:

$$min \sum_{k=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} Dist_{ij} * x_{ijk} \qquad (1)$$

Vehicle capacity constraint should be satisfied in a way:

$$\sum_{i=1}^{n} d_i * y_{ik} \leq Qk = 1..m \qquad (2)$$

Customer's demand is served by exactly one vehicle. Therefore, this can be expressed by the following constraint

$$\sum_{k=1}^{m} y_{ik} = 1, i = 1..n \qquad (3)$$

Finally, ensure that vehicles start from and return to the depot as expressed by the following equations (4) and (5)

$$\sum_{i=1}^{n} x_{ijk} = y_{ik}, j = 1..m, \forall k \qquad (4)$$

$$\sum_{j=1}^{n} x_{ijk} = y_{ik}, i = 1..n, \forall k \qquad (5)$$

3. Related Works

In this section, we propose a brief overview of the most successful works in solving the VRP problems. The problem first originated in 1954 by [14]) under a study aiming at solving the very large scale of TSP. In 1959, Dantzig et al. [15] proposed a simple matching-based heuristic for its solution [29] and the problem was designated as "vehicle routing" for the first time in 1972 [18].

Since its lifespan, literature has been full of works proposed to solve the VRPs. Based on mathematical modeling (programming formulations), a variety of exact algorithms can be found. Some formulations contain vehicle flow or commodity flow variables. [6] have proposed branch-and-cut based on the two-commodity formulation. [30] have reported a complete branch-and-cut based on the two-index formulation. The VRP can also be formulated as a set partitioning problem to which some valid inequalities are added. A successful implementation based on this methodology can be found

in [5]. A recent survey on these models can be found in [7].

The large instances of VRPs are still difficult to solve by means of exact methods. Thus, most efforts are devoted to the development of heuristic methods.

Heuristic methods fall into two main classes; Classical heuristics and Meta-heuristics. Both categories aim at generating good solutions in a reasonably short time, regardless of their limited exploration of the search space. The difference between the two categories is that heuristics are problem specific methods while Meta-heuristics are general methods applied for many problems. Heuristics can be classified into three main classes; Constructive methods, Two-phase methods and Improvement procedures. The following explains briefly the main principle for each class.

Constructive heuristics build feasible solutions partially. Many works have been proposed belonging to that class, for example: the Nearest Neighborhood Heuristic (NNH) in which additional nodes are added at each step to the first chosen node. The nodes selected to be inserted depend on their least cost [39]. In contrast, saving methods construct solutions which may be infeasible after calculating the savings generated by new route configuration by [10], which has undergone a robust enhancement by [17]. Additionally, the sweep algorithms are classified under the constructive heuristics, such as in [21]. Two-phase methods consist in dividing the subject problem into two phases: clustering and routing, such as in [37], and route-first cluster-second methods [8].

Generally, the constructive methods are post-optimized by Improvement procedures based on the exchange concept. An example of such procedures can be found in [35] where 2-opt* are exchanges applied to individual routes. A combination of 2-opt moves, vertex assignments to different routes and vertex interchanges between routes are proposed in [34]. A survey reporting the most advent heuristics can be found in [28].

By another side, meta-heuristics remain the fortunate choice for exploring widely and deeply the search space. The main principal meta-heuristics perform an optimization process in order to obtain good solutions in a reasonable

time. Among these, simulated annealing [19] and Ant Colony Optimization (ACO) [32] have been used to solve CVRP, and improvements have been suggested in [45]. Literature on the application of Genetic Algorithm (GA) for solving VRP is limited. An effective GA was presented in [4] and in [36]. Moreover, a hybrid genetic algorithm was reported by [43]. A scatter search method was applied to the vehicle routing problem with simultaneous delivery and pick up [31]. Also, [40] have proposed an enhanced version of the artificial bee colony algorithm to solve the CVRP. Another successful improved version of the artificial bee colony algorithm was proposed for tackling the CVRP with Time dependent Travel times [26]. For further information, the reader may refer to the work of [3].

Over the last years, many parallel strategies were proposed to take advantage from the advent available resources. Undoubtedly, incorporating a set of meta-heuristics (or optimization methods) that work together and complete each other is more useful than using one single search algorithm [2]. Parallel algorithms do not only contribute in speeding up the search process by exploiting the potential of computing resources, but they also enhance the quality of the obtained solutions by new exploration patterns. The work of [38] deals with a distributed heterogeneous evolutionary algorithm performing evolution on separate populations for solving the CVRP. An introduction of parallel meta-heuristics can be found in [13]. [33] have proposed a parallel evolutionary algorithm which uses a GA and scatter search as hybrid metaheuristics for solving the vehicle routing problem with heterogeneous fleet. Also, there is the work presented in [9] where a parallel hybrid genetic algorithm uses a co-evolution of two populations. The latter focuses on minimizing the total traveled distance and minimizing temporal constraint violation to generate a feasible solution respectively, for tackling the Vehicle Routing Problem with Time Windows. The utilization of some Cellular GAs including local search techniques for solving the vehicle routing problem (VRP) was proposed in [1]. Authors in [27] suggest a work that uses such principle for solving the VRP in which multiple islands using tabu search cooperate asynchronously in parallel to solve the large CVRP (LCVRP). In [44], an island model based on genetic algorithm that applies mainly an offspring selection and adaptive constraint relaxation is presented.

4. The Proposed GIM for Solving LCVRP

There exists a variety of nature inspired metaheuristics in the literature and none of them has been proven to outperform the others in all problems. Furthermore, the quality of the search process using a metaheuristic depends on the balance between its explorations and exploitation abilities, which is often influenced by the adopted settings of the algorithm's parameters. Given these facts, we propose tackling LCVPR using cooperation between different metaheuriscs with different settings. GIM provides a suitable framework for the implementation of such cooperation. Basically, the proposed GIM relies on the use of two different metaheuristics, namely genetic algorithms and ant colony optimization. Using different parameters settings, different variants of GAs and ACO algorithms are used and organized into archipelagoes as shown in Figure 2. As can be seen in this figure, the architecture of the proposed GIM is based on two archipelagoes. Each archipelago is in fact an island model that includes three variants of the same metaheuristic. Within each archipelago, the islands are onto a fully connected topology (each island sends/recieves to/from all the other islands).

B. GA Archipelago   A. ACO Archipelago
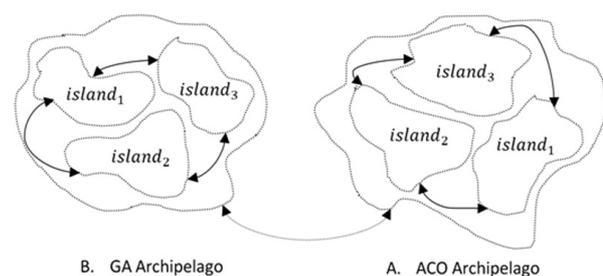
Figure 2. The architecture of the CPGA.

4.1. The CPGA Environment

The environment in which the meta-heuristics cooperate in parallel is a synergic one, ruled by the GIM settings. The synergy is created by the interaction between different islands. Indeed, and as shown in Figure 2,

within the same archipelago, a synchronous inter-communication is set after a defined number of iterations, known as the Migration interval. This latter denoted (μ) specifies the iterations of an optimization process denoted $\mathscr{A}$ before a migration occurs. While asynchronous, the interaction between the different archipelagoes is defined whenever the migration event is launched.

The core idea behind using the GIM is improving diversity and avoiding premature convergence by exchanging solutions. Therefore, two issues need to be addressed to properly handle solution migration: which solution to exchange and which one should be replaced at each migration. The selection policy $\mathscr{S}$ allows selecting individuals from the current island to be sent to other islands, as well as the introduction of foreign individuals to the local population by a replacement policy $\mathscr{R}$ [25].

4.2. The Adopted Encoding of CVRP Solutions

CVRP solving can be viewed as the process that consists in grouping the set of customers into groups such that customers within the same group are served by the same vehicle. Therefore the number of groups is given by the number of vehicles. The order of customers in a group defines the transportation plan (the tour of the corresponding vehicle). Hence, a potential CVRP solution is a partition of the set of customers. To derive such partitions, a binary matrix is used. As shown in Figure 3, this matrix has number of vehicles and number of customers as number of rows and number of columns respectively. Each customer is assigned to only one vehicle. An element of this matrix is set to 1 if the corresponding customer is assigned to the corresponding vehicle, otherwise it is set to 0. As a consequence, the sum of elements of the same column should be equal to 1. A potential solution is feasible if it satisfies the constraints given in Section 2. It is worthy to notice that after assigning the first customer randomly to one of the vehicles, the vehicle capacity Q is updated to Q'. The latter consists in subtracting the demand of the customer assigned (d_i) from the vehicle capacity according to the following equation: $Q' = Q - \sum(d_i)$. The process is repeated in a random way by taking into account the constraints till no remaining clients to be served exist after every new assignment of a vehicle to a client. Figure 3 illustrates the encoding formula and structure.
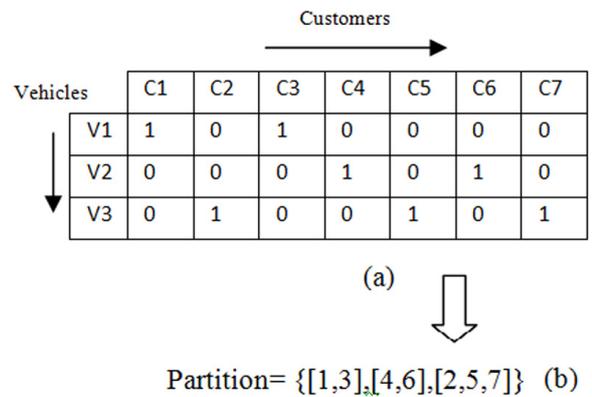

Figure 3. Solution representation: (a) binary matrix (b) Corresponding partition.

4.3. The GA Archipelago

The GA based archipelago is composed of three islands; each one performs a version of a GA for solving the LCVRP. The versions differ from each other in the rate and the probability of application of one of the genetic operators. Yet, they share the same general genetic scheme shown in the pseudo-code. The use of these different versions has allowed studying the impact of using some sensitive parameters to focus on either exploration or intensification. Because some genetic operators alter the solutions differently, some operators perform unequally well along the whole optimization process. In fact, some of these operators disturb the population to enhance the solutions quality at the beginning

Operators	Island1	Island2	Island2
Selection	10% of the population (randomly)	20% from best solutions	30% from best and randomly the rest of population
Crossover	5%	10%	15%
Mutation	5%	10%	15%

Table 1. The different probabilities of genetic operators over the GA archipelago.

of the search process whereas some are more effective when fine-tuning already good solutions [44]. The values proposed in Table 1 were empirically taken regarding that selecting the best solutions for the next generations leads to intensification, while selecting individuals randomly from the population yields to intense the diversification process.

By consequence, some islands focus in their search process on the intensification, while the others on the diversification. The pseudo-code shown in Algorithm 1 illustrates the basic functioning steps of the GA. The construction of initial solutions is achieved via the constructive heuristic proposed in [10]. Selection, crossover and mutation operators are applied according to their respective probabilities as seen in Table 1. The crossover used along the GA is a *recombination edge operator(ERX)* [1]. This operator builds an offspring by preserving edges from both parents. The mutation operator used

---

Algorithm 1. Genetic algorithm for CVRP.

Begin
   Initialize the Population by the constructive heuristic
   Fitness Evaluation of the initial population;
   Repeat
   If nbr_itrG reached
      Apply Selection operator;
      Apply Crossover operator;
      Apply mutation operator;
      Fitness Evaluation of the current population;
      Apply improvement procedure if pr <prLocal
   Until the end condition is satisfied;
   Return the best solution found;
End

Algorithm 2. Ant Colony Optimization algorithm for CVRP

Begin
   τ= intializePheromoneTrails()
   η=initializeVisibilities()
   sbest =Initialsbest;
   While notStopCriteria() do
      Pop=constructAntsSolutions(τ, η);
      Pop'=Apply 2-opt method (pop);
      τ=updatePheromones(T,pop');
      S=selecteBestOfPopulation(pop')
      if f(s)<f(sbest) then update best-so-far solution
   end While;
   return sbest;
End

---

in the algorithm consists in applying *Swap* or *Move* operators alternatively (see Section 5.2. The GA process maintains populations of good solutions that are recombined to produce new solutions. Moreover, local search methods are applied to improve the GAs. Because these improvement procedures are used in both archipelagoes, they will be explained later.

4.4. The ACO Archipelago

Since its inception [16], ACO metaheuristic has been widely used. As known, this metaheuristic has been inspired by the foraging behavior of ant colonies. The communication between ants is ensured via the pheromone trails during the search' food process to point the shortest path. The intensity of the pheromone trail increases on the shortest path because of the passing by of an increasing number of ants. Until finally, nearly all ants take this shortest path. Its successful application to solve the traveling salesman problem (TSP) has led to the emergence of many variants, ACS, MAMS, etc.

The ACO archipelago is also composed of three islands. Except for the probability of applying some local search methods, the different islands perform the same general scheme of the ACO optimization process shown in Algorithm 2. The different parameters of the ACO, related to the visibility and the attractiveness of the arcs are set as exposed in Table 2.

At the beginning, *m* ants are positioned at the depot. Each ant (k) will generate a complete tour by choosing, from the current city (i), the next city (j) to visit, in an instant (t) according to the transition rule detailed in (6). The N_i^k as the cities unvisited yet by the ant (k) at the instant (t) during the current tour.

$$P_{ij}^k(t) = \frac{\left[\tau_{ij}(t)\right]^\alpha * \left[\eta_{ij}(t)\right]^\beta}{\sum_{y \in N_i^k} \left[\tau_{iy}(t)\right]^\alpha * \left[\eta_{ij}(t)\right]^\beta}, if\, j \in N_i^k$$

(6)

This latter defines the cities containing high amount of pheromone by selecting the τ_{ij} that represents the attractiveness of the arcs, calculated according to (7, 7.a , 7.b and 8).

$$\tau_{ij} = \rho.\tau_{ij}(t) + \Delta\tau_{ij}(t) \qquad (7)$$

Parameter	Role	Recommended value	Chosen values
α	Importance related to the intensity of the pheromone trail $\tau(t)$	1	2
β	Importance related to the visibility η	$1 \leq \beta \leq 15$	7
m	Number of the ants	–	number of vehicles of the instance (m)
τ_0	Initial quantity of the pheromone	$1/n$	1/ number of customers
ρ	Amount determining the frequency of deposing the pheromone	$0 < \rho < 1$	0.99

Table 2. Parameters of the ACO.

where the parameter n in the 7.a represents the number of cities (cutomers). The (7.b) represents the amount of pheromone deposited by the ant (k) at an instant (t) on the arc (i,j)

$$\rho_t = 1 - \rho . \cos(\frac{\pi t}{3n}) \qquad (7.a)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t)$$
$$(7.b \text{ update pheromone formula})$$

By assuming that $T_k(t) = (u_{k_1}, \ldots u_{k_g})$ the tour traveled by the Kth ant during the time $[t, t+, n]$. The parameter Q is a constant value. The $cost_{k(t)}$ denotes the total length traveled so far and is obtained by analyzing the ant' memory (8).

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} Q/cost_k(t) \ (if \ (u \in T_k(t) \wedge u = (i,j)) \\ 0 \qquad\qquad\qquad\qquad\quad otherwise \end{cases}$$
$$(8)$$

while η_{ij} represents the visibility of the arcs according to (9).

$$\eta_{ij} = \frac{1}{Dist_{ij}} \qquad (9)$$

Consequently, the cities with high pheromone represent the shortest edges. The updating rule modifies the pheromone values by increasing their level on the edges which belong to the current best tour. The whole process is repeated till the stagnation of the algorithm by producing the same tours or by reaching the defined number of iterations.

5. Improving Solutions Using Local Search

The quality of the solutions obtained is enhanced by the use of some improvement procedures. These methods involve deleting some arcs and replacing them with new ones. Two types of these refinement procedures were used, namely inter-route refinement procedures and intra-route refinement procedures. The former operate between tours while the latter operate on the sub-tours within the same tour. The refinement process is repeated till no further improvements can be reached.

5.1. Inter-route Refinement Procedures

- Exchange operators: It consists in exchaging a client between two different tours(4a), called (1-1 exchange). Figure (4b) consists in removing a client from a tour to be inserted in another one, called 1-0 exchange.

- Inter-route 2-opt operator: this operator replaces one sub-tour by another one removed from another tour, Figure (4c).
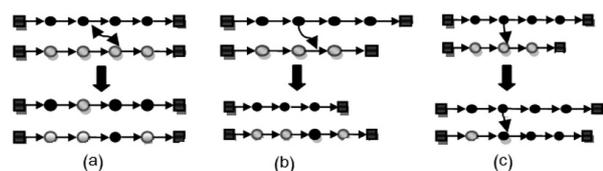

Figure 4. Inter-route operators.

5.2. Intra-route Refinement Procedures

These procedures were originally developed to TSP problem. It should be noted that in one run we apply randomly one operator to improve one route.

- Move operator: it consists in choosing a random client to be moved from its actual position to another random one as shown in Figure 5.
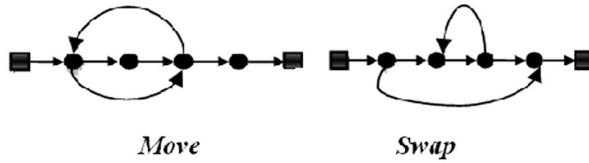
Figure 5. Move and Swap operators.

- Swap operator: the Swap operator consists in swapping two clients chosen randomly.

- Intra-route 2-opt operator: 2-opt algorithm is often used for improving a tour. This operator removes two edges and reconnects the paths created. This process is repeated till no further improvement can be found, as presented in Figure 6.
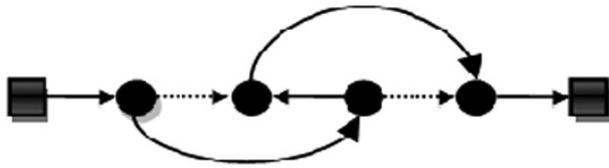

Figure 6. 2-opt mutation operator.

6. The Communication Model

Migration of solutions is a very important process that maintains the interaction between the different islands. The migration event is launched after reaching a μ (migration interval). It is worthy to assume that the algorithm used on islands should be allowed to achieve a sensible progress in optimization between two migration events. This is why the choice of the migration interval should be proceeded at least by experimental observation of the optimization progress on an isolated island [25].

This communication aims at improving the quality of the obtained solutions. Two types of communication are used in the proposed GIM: inter and intra communication. The former is related to the interaction between the different islands within the same archipelago while the intra-communication describes the interaction between the archipelagoes. In order to further explain the communication model, some notations used are presented. We define $\mathbf{A} = \langle \mathbf{I} \rangle$ the archipelago where the $\mathbf{I} = Is_1, Is_2, \ldots, Is_n$ is the set of islands. Every island Is_i is a quadruple:

$$Is_i = \langle \mathscr{A}_i, P_i, \mathscr{S}_i, \mathscr{R}_i \rangle \qquad (10)$$

where

- $\mathscr{A}_i$: the optimization process used.
- P_i: A population containing a set of individuals on each island
- $\mathscr{S}_i$: the selection policy
- $\mathscr{R}_i$: the replacement policy

The selection policy $\mathscr{S}_i$ denotes the set of solutions (individuals) $M \subseteq P$ to be sent to other islands via the migration process. We write this as $M \leftarrow \mathscr{S}_i(P)$. Also, given a deme M, the migration-replacement policy $\mathscr{R}_i$ specifies how M could be inserted into another reciever_population P. We write this as $P' \leftarrow \mathscr{R}(P, M)$. Also, since two types of optimization processes are used, $\mathscr{A}_t$ for denoting the type of the algorithm *(t= ACO or GA)* used.

6.1. Inter-communication Process (Between Islands)

Algorithm 3 describes how migration between islands within the same archipelago is performed. Indeed, after the migration interval (μ) is reached, the selection policy $\mathscr{S}_i$ consists in sending from each island the best solutions found so far $M = best_{Is}$. The numbers of solutions sent are in accordance with a defined migration rate $(\Im)$. We note this process as $best_{Is} \leftarrow \mathscr{S}_i(P)$. On the other side, the replacement policy $\mathscr{R}_i$ consists in receiving immediately the sent solutions. These received solutions will be integrated into the populations of each receiver_island, with an

---

Algorithm 3. The inter communication.

Begin
   While $i =!n(/*I$ denotes the ^{ith} island)
     Repeat
         $P' \leftarrow \mathscr{A}_i(P)$
         $M \leftarrow \mathscr{S}_i(P')$
   $*$ sending best solutions from the island (I_i) to all islands
         $P'' \leftarrow \mathscr{R}_i(P', M')$
   $*$ receiving and integration the sent solutions by the
receiver_islands
         $P \leftarrow P''$
     End_Repeat
   End_While
End

interval of migration specified by a number of iterations nbr_itr.

Algorithm 4. The intra communication.

Begin

    Set a Solution buffer

    $P_i' = \mathscr{A}_t(P_1, P_2, \ldots P_n)$

    $M' \leftarrow \mathscr{S}_j'(P_i')$

    $P_i'' \leftarrow \mathscr{R}_i'(P_i', M')$

    $P_i \leftarrow P_i''$

End

6.2. Intra-communication Process (Between Archipelagoes)

The ACO and the GA archipelagoes interact asynchronously through a buffer. The use of such asynchronous communication is supported by the conclusion that has been come up by [12]. Each archipelago sends the best solutions M found so far over all the islands, we write this as $M = \sum P_i'$. Following the selection migration policy $\mathscr{S}_j'$, the best solutions selected are in accordance with a migration rate $(\Im')$. These solutions are sorted depending on their fitness to define the best over-all $bestG_j$ which will be sent to the islands of the other archipelago. This latter will be compared to the one on the other archipelago to be updated. For the rest of the solutions, following the replacement policy $\mathscr{R}_j'$ they will be sent to the islands of the receiver archipelago. The whole process is repeated till one stopping criterion is reached.

7. Experimental Results

In this section, we describe the aspects related to the implementation of the proposed GIM including the used data sets and the obtained results. We report as well on the comparative study we conduct with regard to the best-known solutions found in the literature available online and updated as in [27] and to the results obtained by each meta-heuristic running separately. The performance measure used is the fitness function defined by equation 1.

7.1. Used Data Sets and Parameters Settings

The proposed GIM has been implemented in MATLAB8.0 on a computer that consists of 8 processors (R2013b) at 16 GHZ. Large instances of the CVRP have been used to assess the performance of the GIM. They consist of 20 benchmarks described in [23]. They include $200 - 483$ customers. The parameters of the GIM have been set as shown in Table 3. For each instance, the program has been run 10 times and statistics have been computed. In all our experiments, we are interested in the values of the objective function described in equation 1 and that gives the total cost required to serve all customers. This total cost should be minimized. $|N|$ represents the instance size.

Parameter	Value
Number of islands	3 on each archipelago
Number of archipelagoes	2
Number of iterations of each island	$1000 * \sqrt{|N|}$
Solution exchange (Migration rate $\Im$)	$1000 * \sqrt{|N|}/nbrof\ islands$
Termination condition	Maximum number of iterations

Table 3. Parameter values of the CPGA.

7.2. Computational Results and Discussion

Table 4 presents in its first and second columns respectively the best-known results and the results reported by [27] (best and average). The remaining columns report respectively: the average of the results obtained for each instance, the standard deviation (SD) and the best results recorded by CPGA. The first of the last two rows presents the average deviation, while the second shows the number of runs per instance. As it is shown in the table, the CPGA has recorded successfully almost the same best solutions known in the literature, especially in the instances ($Golden\_6, 7, 9, 10, 11, 16, 17, and 18$).

The results in bold represent the scores obtained and they are better than the best known ($Golden\_1 and 4$) while those in gray color are exactly as the best-known. The results in italic are those better compared to [27].

Instances	Previous BestKnown	Gror, C et al. [24]	Vidal et al [43]	Jin et al. [27]		CPGA		
				Best	Aver.	Aver.	SD	Best
Golden_1(240)	5623.47	5623.47	5623.47	5623.47	5623.65	5623.424	0.274	**5623.12**
Golden_2(320)	8404.61	8447.92	8404.61	8405.81	8434.78	8405.624	0.014	*8405.61*
Golden_4(480)	13592.88	13624.52	13624.52	13 590.00	13620.3	13591.24	0.101	**13591.1**
Golden_5(200)	6460.98	6460.98	6460.98	6460.98	6460.98	6460.994	0.00489898	6460.99
Golden_6(280)	8400.33	8412.90	8412.9	8400.33	8404.06	8400.388	0.0744043	8400.33
Golden_7(360)	10102.70	10195.59	10102.70	10 107.49	10134.93	10102.84	0.12	*10102.7*
Golden_8(440)	11635.30	11663.55	11635.30	11 635.34	11635.34	11635.584	0.17036432	11635.4
Golden_9(225)	579.71	579.71	579.71	579.71	580.04	579.71	0.00	579.71
Golden_10(323)	736.26	737.28	736.26	735.66	737.16	736.26	0.00	736.26
Golden_11(399)	912.84	913.35	912.84	912.03	912.72	912.846	0.008	912.84
Golden_12(483)	1102.69	1102.76	1102.69	1101.50	1103.2	1102.932	0.053066	1102.87
Golden_13(252)	857.19	857.19	857.19	857.19	858.57	857.19	0.00	857.19
Golden_14(320)	1080.55	1080.55	1080.55	1080.55	1080.55	1081.684	0.17442477	*1081.5*
Golden_15(396)	1337.92	1338.19	1337.92	1337.87	1340.13	1338.932	0.03487119	1338.9
Golden_16(480)	1612.50	1613.66	1612.50	1611.56	1614.73	1612.62	0.19390719	1612.5
Golden_17(240)	707.76	707.76	707.76	707.76	707.8	707.76	0.00	707.76
Golden_18(300)	995.13	995.13	995.13	997.58	998.9	995.13	0.00	*995.13*
Golden_19(360)	1365	1365.60	1365.60	1365.60	1366.12	1365.724	0.02244994	1365.7
Golden_20(420)	1818.25	1818.25	1818.32	1817.89	1819.76	1818.99	0.02	1818.95
Aver deviation(%)	–			0.11		0.07		
Runs per instance	–	5		10		20		

Table 4. Comparative table of the results obtained, the best-known in [27].

Clearly, in some instances a slight difference between the results (*Golden_*12, 8, 5*etc*) can be explained by fine tuning the parameters required on each archipelago. Actually, the ACO archipelago uses many parameters that need a fine tuning, such as α and β, besides the GA archipelago that uses parameters relative to the application of the recombination operators. Although different islands use different parameters in order to enhance the diversity and to raise the survival chance for different individuals with different solutions qualities, still the interplay between the different parameters require a great attention and fine tuning.

Figure 7 shows the evolution of the fitness values over the instances of Golden et al. [23] of the CPGA and the [27] after 10 runs. Besides the average deviation of the CPGA is smaller than the [27], we can deduce that the CPGA is more stable than the approach compared with.

Another set of experiments has been conducted over the Taillard instances [41]. A comparison between the rests obtained and those of [24] has been described in Table 5 the best known were taken as cited in [24].

When going through the results shown in Table 5, we conclude that the proposed approach has recorded exactly the same as the Best known, highlighted in italic (8 out of 9), whereas, the comparison of the obtained results against those gotten by [24] shows that the CPGA outperforms the [24], especially in the instance (150C). The last instance was slightly worse than the best known solution.

7.3. The Effect of the Cooperation

In the second set of experiments, and in order to prove the efficiency of the proposed approach,
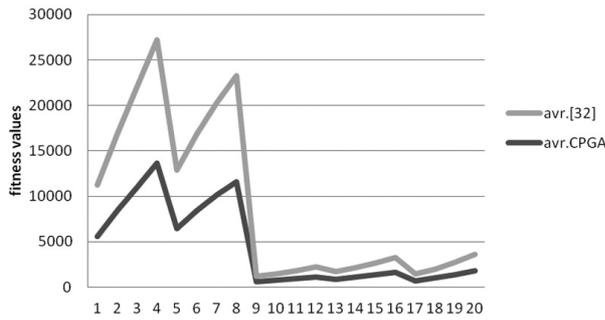
Figure 7. Comparison of the averages of the CPGA and the [27].

Instances	Previously Best knowns	Gror, C. et al. [24]	CPGA
100A	2041.34	2041.34	2041.34
100B	1939.90	1939.90	1939.90
100C	1406.2	1406.20	1406.20
100D	1580.46	1580.46	1580.46
150A	3055.23	3055.23	3055.23
150B	2727.20	2727.20	2727.20
150C	2341.84	2358.66	2341.84
150D	2645.40	2645.40	2645.40
385	24369.13	24366.69	24370.32
Runs per instance	−	5	20

Table 5. Comparison between best-known [24] and the CPGA results.

Instances	GA	ACO	CPGA
Golden_1	6247	6001	5623.12
Golden_2	9093.5	9384.5	8406.61
Golden_3	12463	12593	11036.5
Golden_4	16010	15621	13591.1
Golden_5	7589.5	7858	6460.99
Golden_6)	9746	9745.8	8401
Golden_7	11871	12110	10102.7
Golden_8	13252	13110	11635.4
Golden_9	688	673	579.71
Golden_10	879.41	861	736.26
Golden_11	1109.6	1137	912.84
Golden_12	1340.91	1287	1102.87
Golden_13	1023.6	981.6	857.19
Golden_14	1243	1267	1081.5
Golden_15	1751.8	1645	1338.9
Golden_16	1966.9	1855	1612.5
Golden_17	907.1	1001.1	707.76
Golden_18	1149.4	1098.4	995.13
Golden_19	1578	1491	1365.7
Golden_20	1994	2118.25	1818.95

Table 6. Impact of the cooperation on the performance of the CPGA

we have compared the CPGA against the GA and the ACO where each meta-heuristic is used as a separate optimization process to tackle the L-CVRP. The results obtained after 20 runs are shown in Table 6. Many observations can be drawn from the set of figures in Figure 8. In fact, the different box-plots associated to the test of Friedman have been conducted on each instance from the set of the instances [23]. The reason behind such statistical analysis is highlighting to what extent the cooperation between different islands could enhance the performance of metaheuritics operating separately to tackle a defined problem.

The box-plots of all the instances show the high stability of the CPGA, over 20 executions we may find an outlier value, but still too close to the average value. The Friedman test shows that in some cases, no significant difference between the results was reported by the GA method and the CPGA (Figures 8.1, 8.3, 8.4, 8.6, 8.7, 8.8, 8.14, 8.16, 8.18, 8.19 and 8.20) while the ACO results fall always far (see Figures 8.1 to 8.20). One explanation may be that the ACO process requires great attention to the parameters. Cooperation between the different islands increases the search effort.

8. Conclusion

The present work proposes cooperative parallel meta-heuristics to solve the large capacitated vehicle routing problem. The experimental results on a data set of large scale CVRP have reported four new best results compared with the best-known in the literature. Also, the results obtained show that the method is effective and competitive. The new features introduced in the proposed approach deal basically with the use of a new paradigm allowing the cooperation

of several meta-heuristics. The synchronous exchange of solutions and the elitist selection have contributed in increasing the performance of the optimization process. Ongoing works seek on pursuing other approaches involving other types of meta-heuristics and proposing other methods for exchanging solutions.

References

[1] E. ALBA, B. DORRONSORO, Solving the vehicle routing problem by using cellular genetic algorithms. In *Evolutionary Computation in Combinatorial Optimization* Springer, 2004, pp. 11–20.

[2] E. ALBA, M. TOMASSINI, Parallelism and evolutionary algorithms. *Evolutionary Computation, IEEE Transactions on* **6**(5) (2002), 443–462.

[3] S. ANBUUDAYASANKAR, K. GANESH, S. MOHAPATRA, Survey of methodologies for tsp and vrp. In *Models for Practical Routing Problems in Logistics.* Springer, 2014, pp. 11–42.

[4] B. M. BAKER, M. AYECHEW, A genetic algorithm for the vehicle routing problem. *Computers & Operations Research* **30**(5) (2003), 787–800.

[5] R. BALDACCI, N. CHRISTOFIDES, A. MINGOZZI, An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming* **115**(2) (2008), 351–385.

[6] R. BALDACCI, E. HADJICONSTANTINOU, A. MINGOZZI, An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations Research* **52**(5) (2004), 723–738.

[7] R. BALDACCI, A. MINGOZZI, R. ROBERTI, Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research* **218**(1) (2012), 1–6.

[8] J. E. BEASLEY, Route first–cluster second methods for vehicle routing. *Omega* **11**(4) (1983), 403–408.

[9] J. BERGER, M. BARKAOUI, A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research* **31**(12) (2004), 2037–2053.

[10] G. U. CLARKE, J. W. WRIGHT, Scheduling of vehicles from a central depot to a number of delivery points. *Operations research* **12**(4) (1964), 568–581.

[11] J.-F. CORDEAU, G. LAPORTE, M. W. SAVELSBERGH, D. VIGO, Vehicle routing. *Transportation, handbooks in operations research and management science* **14** (2006), 367–428.

[12] T. G. CRAINIC, *Parallel solution methods for vehicle routing problems*, Springer, 2008.

[13] T. G. CRAINIC, M. TOULOUSE, Parallel metaheuristics. In *Handbook of metaheuristics* Springer, 2010, pp. 497–541.

[14] G. DANTZIG, R. FULKERSON, S. JOHNSON, Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America* **2**(4) (1954), 393–410.

[15] G. B. DANTZIG, J. H. RAMSER, The truck dispatching problem. *Management science* **6**(1) (1959), 80–91.

[16] M. DORIGO, V. MANIEZZO, A. COLORNI, Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* **26**(1) (1996), 29–41.

[17] T. DOYURAN, B. ÇATAY, A robust enhancement to the clarke–wright savings algorithm. *Journal of the Operational Research Society* **62**(1) (2011), 223–231.

[18] B. EKSIOGLU, A. V. VURAL, A. REISMAN, The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering* **57**(4) (2009), 1472–1483.

[19] J. FAULIN, M. GILIBERT, A. A. JUAN, VILAJOSANA, X. R. RUIZ, Sr-1: a simulation-based algorithm for the capacitated vehicle routing problem. In *Simulation Conference, 2008. WSC 2008. Winter* (2008), IEEE, pp. 2708–2716.

[20] M. L. FISHER, R. JAIKUMAR, A generalized assignment heuristic for vehicle routing. *Networks* **11**(2) (1981), 109–124.

[21] B. E. GILLETT, L. R. MILLER, A heuristic algorithm for the vehicle-dispatch problem. *Operations research* **22**(2) (1974), 340–349.

[22] B. L. GOLDEN, S. RAGHAVAN, E. A. WASIL, *The Vehicle Routing Problem: Latest Advances and New Challenges: latest advances and new challenges*, **43**, Springer, 2008.

[23] B. L. GOLDEN, E. A. WASIL, J. P. KELLY, I.-M. CHAO, The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. In *Fleet management and logistics*, Springer, 1998, pp. 33–56.

[24] C. GROËR, B. GOLDEN, E. WASIL, A parallel algorithm for the vehicle routing problem. *INFORMS Journal on Computing* **23**(2) (2011), 315–330.

[25] D. IZZO, M. RUCIŃSKI, F. BISCANI, The generalized island model. In *Parallel Architectures and Bioinspired Algorithms* Springer, 2012, pp. 151–169.

[26] P. JI, Y. WU, An improved artificial bee colony algorithm for the capacitated vehicle routing problem with time-dependent travel times. In *Tenth* A Generalized Island Model based on Parallel and Cooperating Metaheuristics for Effective Large Capacitated Vehicle Routing Problem Solving 11 *international symposium on operations research and its applications (ISORA 2011)* (2011), pp. 75–82.

[27] J. JIN, T. G. CRAINIC, A. L/OKKETANGEN, A co-operative parallel metaheuristic for the capacitated vehicle routing problem. *Computers & Operations Research* **44**, (2014), 33–41.

[28] G. LAPORTE, M. GENDREAU, J.-Y. POTVIN, F. SEMET, Classical and modern heuristics for the vehicle routing problem. *International transactions in operational research* **7**(4-5) (2000), 285–300.

[29] G. LAPORTE, P. TOTH, D. VIGO, Vehicle routing: historical perspective and recent contributions. *EURO Journal on Transportation and Logistics* **2**(1-2) (2013), 1–4.

[30] J. LYSGAARD, A. N. LETCHFORD, R. W. EGLESE, A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming* **100**(2) (2004), 423–445.

[31] G. MAQUERA, M. LAGUNA, D. A. GANDELMAN, A. P. SANT'ANNA, Scatter search applied to the vehicle routing problem with simultaneous delivery and pickup. *International Journal of Applied Metaheuristic Computing (IJAMC)* **2**(2) (2011), 1–20.

[32] S. MAZZEO, I. LOISEAU, An ant colony algorithm for the capacitated vehicle routing. *Electronic Notes in Discrete Mathematics* **18** (2004), 181–186.

[33] L. S. OCHI, D. S. VIANNA, L. M. DRUMMOND, A. O. VICTOR, A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet. In *Parallel and Distributed Processing*, Springer, 1998, pp. 216–224.

[34] I. H. OSMAN, Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of operations research* **41**(4) (1993), 421–451.

[35] J.-Y. POTVIN, J.-M. ROUSSEAU, An exchange heuristic for routing problems with time windows. *Journal of the Operational Research Society* **46**(12) (1995), 1433–1446.

[36] C. PRINS, A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research* **31**(12) (2004), 1985–2002.

[37] C. PRINS, P. LACOMME, C. PRODHON, Order-first split-second methods for vehicle routing problems: A review. *Transportation Research Part C: Emerging Technologies* **40** (2014), 179–200.

[38] K. PULJIĆ, R. MANGER, A distributed evolutionary algorithm with a superlinear speedup for solving the vehicle routing problem. *Computing and informatics* **31**(3) (2012), 675–692.

[39] D. J. ROSENKRANTZ, R. E. STEARNS, P. M. LEWIS II, An analysis of several heuristics for the traveling salesman problem. *SIAM journal on computing* **6**(3) (1977), 563–581.

[40] W. SZETO, Y. WU, S. C. HO, An artificial bee colony algorithm for the capacitated vehicle routing problem. *European Journal of Operational Research* **215**(1) (2011), 126–135.

[41] É. TAILLARD, Parallel iterative search methods for vehicle routing problems. *Networks* **23**(8) (1993), 661žč-673.

[42] P. TOTH, D. VIGO, Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics* **123**(1) (2002), 487–512.

[43] T. VIDAL, T. G. CRAINIC, M. GENDREAU, N. LAHRICHI, W. REI, A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research* **60**(3) (2012), 611–624

[44] S. VONOLFEN, M. AFFENZELLER, A. BEHAM, S. WAGNER, Solving large-scale vehicle routing problem instances using an island-model offspring selection genetic algorithm. In *Logistics and Industrial Informatics (LINDI), 2011 3rd IEEE International Symposium on* (2011), *IEEE*, pp. 27–31.

[45] B. YU, Z.-Z. YANG, B. YAO, An improved ant colony optimization for vehicle routing problem. *European journal of operational research* **196**(1) (2009), 171–176.

Contact addresses:
Meryem Ammi
MISC laboratory
Computer Science Department
College of NTIC
Constantine2 University
Constantine 25000
Algeria
e-mail: `meryem.ammi@univ-constantine2.dz`

Salim Chikhi
MISC laboratory
Computer Science Department
College of NTIC
Constantine2 University
Constantine 25000
Algeria
e-mail: `salim.chikhi@univ-constantine2.dz`

MERYEM AMMI is a PhD student in Computer Science at the University of Abdelhamid Mehri (ex. Mentouri Constantine). She is a member of the SCAL team of MISC Laboratory. Her research areas include soft computing, artificial life techniques and their application in solving vehicle routing problems.

SALIM CHIKHI received his PhD in Computer Science from the University of Constantine, in 2005. He is currently a Professor at the University of Abdelhamid Mehri (ex. Mentouri Constantine), Algeria. He is the Leader of the SCAL team of MISC Laboratory as well as the Director of MISC laboratory (modeling and implementation of complex systems). His research areas include soft computing and artificial life techniques and their application in several domains.
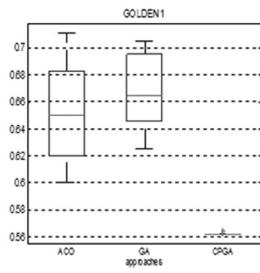
8.1.
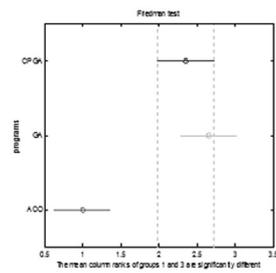
8.2.
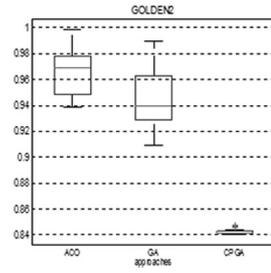
8.3.
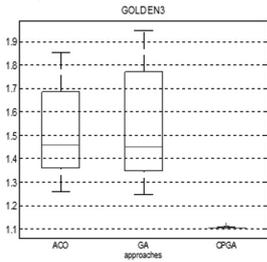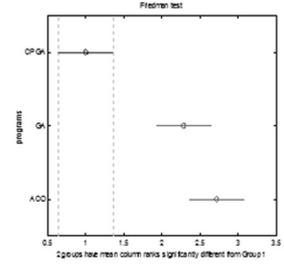
8.4.
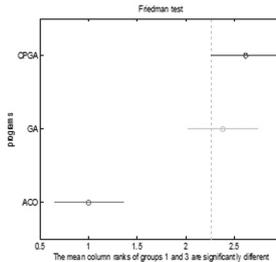
8.5.
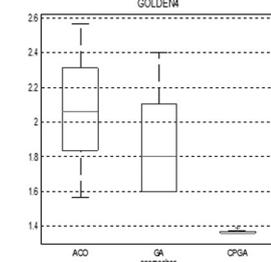
8.6.
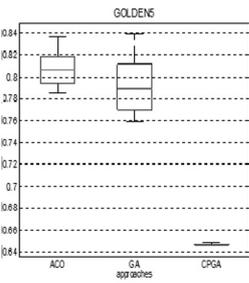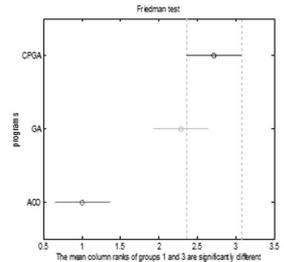
8.7.
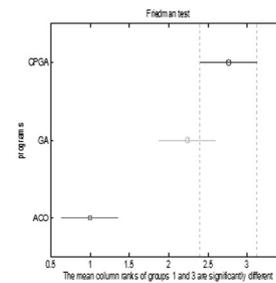
8.8.
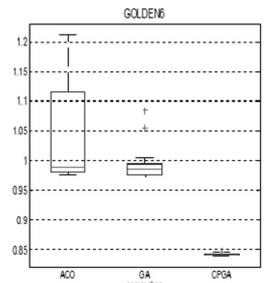
8.9.
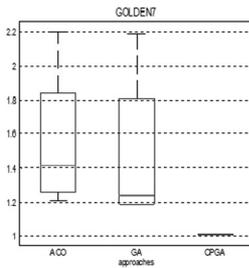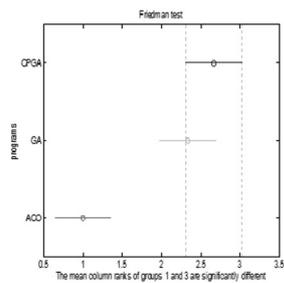
8.10.
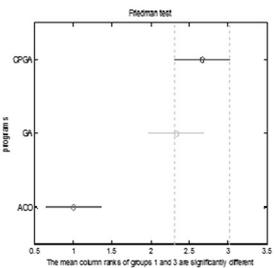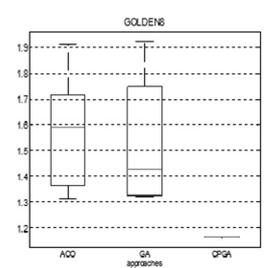
8.11.
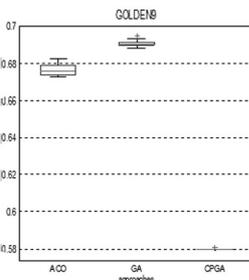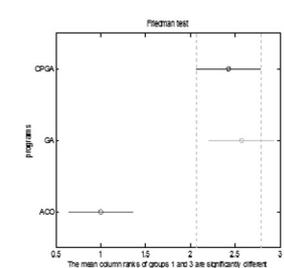
8.12.
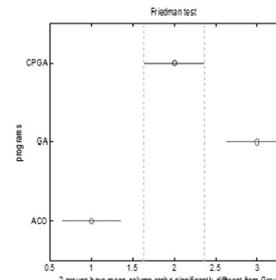
8.13.
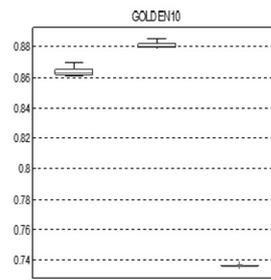
8.14.
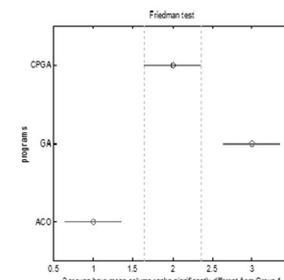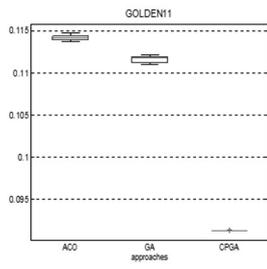
8.15.
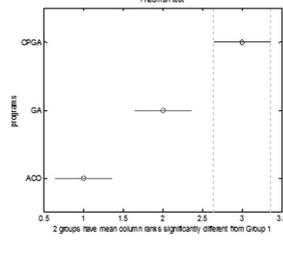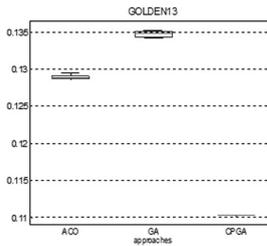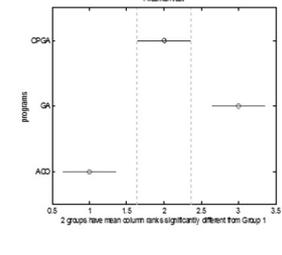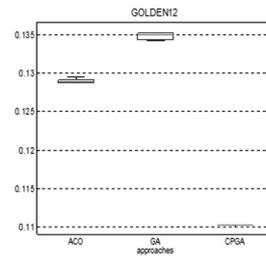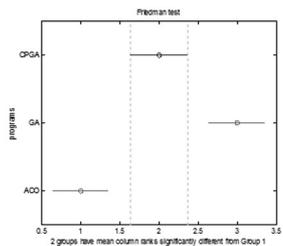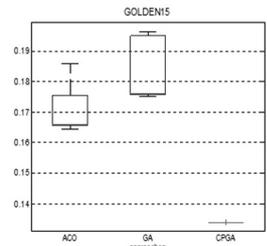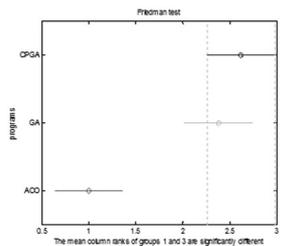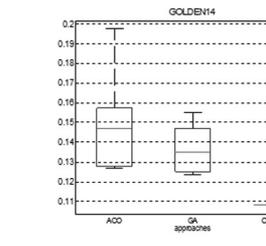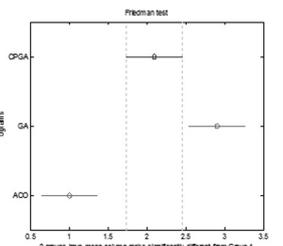
8.16.
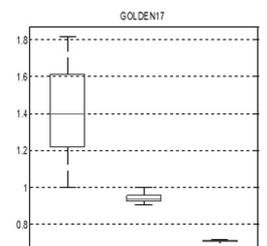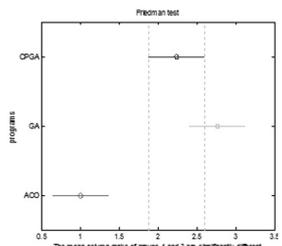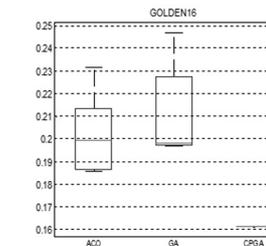
8.17.
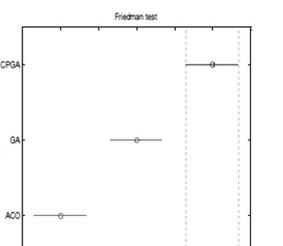
8.18.
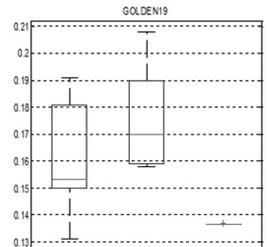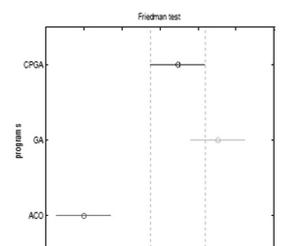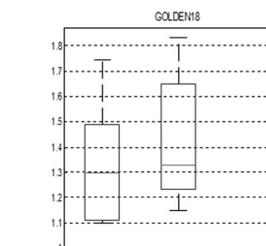
8.19.
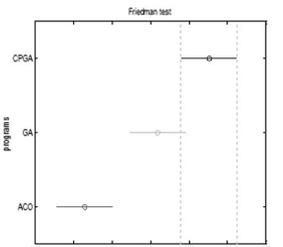
8.20.
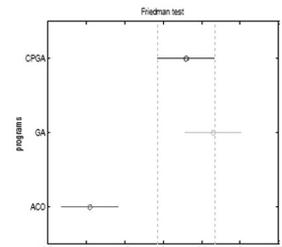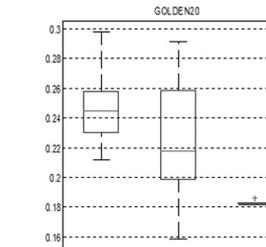
Figure 8. Box-plots of multi-comparison graph between GA, ACO and CPGA.