Hybrid Solution Methodology: Heuristic-Metaheuristic-Implicit Enumeration 1-0 for the Capacitated Vehicle Routing Problem (Cvrp)

David Escobar Vargas
Faculty of Electrical Engineering
Technological University of Pereira
Pereira, Colombia

Ramón A. Gallego Rendón
Faculty of Electrical Engineering
Technological University of Pereira
Pereira, Colombia

Antonio Escobar Zuluaga
Faculty of Electrical Engineering
Technological University of Pereira
Pereira, Colombia

***Abstract*—The capacitated vehicle routing problem (CVRP) is a difficult combinatorial optimization problem that has been intensively studied in the last few decades. We present a hybrid methodology approach to solve this problem which incorporates an improvement stage by using a 1-0 implicit enumeration technique or Balas's method. Other distinguishing features of the methodology proposed include a specially designed route-based crossover operator for solution recombination and an effective local procedure as the mutation step. Finally, the methodology is tested with instances of the specialized literature and compared with its best-known solutions for the CVRP with homogeneous fleet, to be able to identify the efficiency of the use of the Balas's methodology in routing problems.**

Keywords—1-0 implicit enumeration; CVRP; Operations research; Genetic algorithm; Chu-Beasley; Heuristics; Metaheuristics and exact methods

I. Introduction

The general vehicle routing problem (VRP) is a generic name that refers to a variety of applied problems in various areas of knowledge such as transportation, supply chain, production planning, and telecommunications. The general problem is based on a group of customers that have to be served by a fleet of vehicles. They begin their tour or path in a main depot, visiting the customers assigned to the route only once, and returning to the depot where they started. The main objective is to establish the possible routes that represent the least-cost paths that will meet the customers' demands. Customers are spread geographically and associated to one or more constraints that must be met. However, the simplicity of the description is not near to the complexity of the solution search, that is classified as an NP-hard problem and ranked as one of the most interesting optimization problems in operational research. This problem has been analysed and studied extensively since its first appearance in the literature through the formulation applied to the fuel distribution by Ramser and Dantzig in 1959 [1]. From this work, several authors have focused their efforts on finding efficient techniques and models that allow to solve this problem in items (Miller et al, 1960 [2]. Christofides et al, [3][4];. Bodin's at., 1983[5]; Fisher, 1995[6]; Desrosiers et al., 1995; Powell et al. 1995; Fukasa, 2004; Gendreau, 2005a, b; Cordeau et al. 2005, 2007; Laporte, 2009 [7]), and books (Thot and Vigo, 2002 [8]; Golden et al, 2008 [9]). Looking at the state of the problem, it is remarkable that the the large increase in algorithmic and methodical development focused on the variants of the VRP, such as the capacitated routing problem (CVRP) and the VRP with time windows (VRPTW), which highlights the techniques with better behavior for each of these variants. In this paper, the importance of the guidelines proposed in the literature were an important pillar in defining the methodology that would allow us to work on the capacitated vehicle routing problem (CVRP). One of the main objectives of this paper is seeking a technique that allows us to measure the performance of implicit enumeration 0-1 methodology or Balas's algorithm, which has not been applied to the routing problems. Therefore, our work is to define a hybid methodology that is able to solve instances proposed in specialized literature created specifically for the CVRP.

In order to solve this problem, we propose a hybrid methodology that combines the exploration of the solution space through a search based on the evolution of a specialized population of individuals, jointly with diversity techniques in the crossover, a neighborhood search as a mutation stage and a implicit enumeration 0-1 methodology as an improvement stage. The idea is to observe the performance of an exact technique (Balas´s) applied to the capacitated routing problem. The exact technique gets in charge of a sub-problem of the CVRP, as has been done in other areas [10]. The methodology called modified Genetic Algorithm of Chu-Beasley with Implicit Enumeration (GACBIE) is tested with a set of instances proposed in the specialized literature and compared to the best known solutions (BKS) for CVRP problem.

The paper is organized as follows: section II shows some related works and part of the specialized literature for the CVRP, section III shows a formal description of the capacitated vehicle routing problem followed by the methodology proposed in section IV, the pseudocode of the complete algorithm in Section V, ending with the results of tests on the proposed instances in Section VI. Finally, conclusions and recommendations from this work are illustrated.

II. Related Works and Other Literature

The CVRP plays a particular role on VRP algorithmic research, for both exact and on heuristics methods. Being the most basic variant, it is a natural testbed for trying new ideas.

Its relative simplicity allows cleaner descriptions and implementations, without the additional conceptual burden necessary to handle more complex variants. Successful ideas for the CVRP are often later extended to more complex variants. For example, the classical CVRP heuristic by Clarke and Wright [11] was adapted for the VRP for many other variants, as surveyed in Penna [12].

Laporte and Nobert [13] presented an extensive survey which was entirely devoted to exact methods for the VRP and gave a complete and detailed analysis of the state of the art up to the late 1980s. Up to the end of the last decade, the most effective exact approaches for the CVRP were mainly branch and bound algorithms using basic relaxations, as the assignment problem and the shortest spanning tree. Recently, more sophisticated bounds were proposed, as those based on Lagrangian relaxations or on the additive approach, which increased the size of the problems that can be solved to optimality by branch and bound. Moreover, following the success obtained by branch and cut methods for the TSP, encouraging results were obtained by using these algorithms for the CVRP.

Since the 1980s, almost all articles proposing new heuristic and metaheuristic methods for the CVRP reported results on a subset of the instances by Christofides and Eilon [14] and Christofides et al. [15]. This classical benchmark is also exhausted, since most recent heuristics find systematically the best-known solutions on nearly all instances.

Rochat and Taillard [16] proposed an efficient local search to solve the problem with probabilistic diversification and intensification step to solve big instances of the CVRP where they report optimal solutions for almost all the instances except on a single instance with 199 customers.

In Fukasawa *et al*. [17], a branch-cut-and-price is proposed to solve all its instances with up to 100 customers, as well as instances with 121 and 135 customers. In particular, the column and cut generation algorithms of Baldacci *et al.* [18] significantly reduced the CPU time required to solve many instances, but could not solve the some larger instances. Later, some methodologies as the ones proposed on Contardo and Martinelli [19] and Røpke [20] were capable of solving bigger instances.

In a recent work, the heuristics of Vidal et al. [21] are considered one of the best available for the CVRP basically due to their superior performance on Golden's (big) instances. In the other hand, some surveys covering exact algorithms, but often mainly devoted to heuristic methods, were presented by Christofides et al. [3], Bodin et al. [5], Laporte [22], Fisher [6], Toth and Vigo [8] and Golden et al. [9].

III. Problem Formulation

The formulation of the problem of routing vehicles can be defined as follows. Let G = (V, A) a complete graph with $V = n + 1$ vertices, divided into two groups $V = V^{DEP} \cup V^{CST}$. The unique vertex $v_0 \in V^{DEP}$ represents the depot where the products must be distributed and a fleet of m identical vehicles with limited capacity Q departs and returns. The vertices $v_i \in V^{CST}$ represent the customer indexed with i, where $i = \{1, \dots, n\}$, which require a service that is characterized by a non-negative demand associated q_i. The arcs $a_{ij} \in A$ represent all possible connections from a node v_i to another node v_j with a service cost equal to c_{ij}. The main objective of the problem is to define a set of routes that allow every set of connections to dispatch the customer demand without exceeding the maximum capacity of the vehicles so that each sum of the costs associated to the established connections represent the minimum path.

The general VRP consist of the construction of a group of maximum *m* routes that satisfy the following requirements: (i) each route must start and end at the depot; (ii) all customer demands must be met; (iii) the capacity of the vehicle must not exceed the capacity limit *Q*; (iv) each client should be visited only once by a single vehicle or route.

As a mathematical representation of the problem, a binary linear programming model is proposed for the capacitated vehicle routing problem that is presented as an optimization problem. Its objective function is associated to the minimization of the cost or the total distance traveled by all vehicles, while restrictions associated to the visit of each customer by a single vehicle per route and the vehicle's capacity constraints are met. The model presented below corresponds to a modified version of the one proposed by Toth and Vigo in [8]. In this case variables can assume the value of 1 if the arc (*i, j*) is part of the solution and 0 otherwise.

The way the model is proposed is widely used for both the problem of VRP with symmetrical distances and VRP with asymmetric distances. The formulation has the following notation:

Sets

$\boldsymbol{V}$	Set of customers indexed by i, j and an unique depot indexed by $\{0\}$
$\boldsymbol{K}$	Set of vehicles indexed by k
$\boldsymbol{S}$	Set of proper subset of all nodes V

Constants

$\boldsymbol{c_{ij}}$	The distance from node *i* to node *j*
$\boldsymbol{q_i}$	The demand at node $i \in V\backslash\{0\}$
$\boldsymbol{Q}$	The loading capacity of the vehicle

Decision Variables

$\boldsymbol{x_{ij}}$	Binary decision variable, taking a value of 1 if the vehicle goes from node *i* to node *j* and 0 otherwise
$\boldsymbol{u_i}$	The remaining load when the vehicle leaves node *i*

$$\sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (1)$$

$$\sum_{j \in V\ j \neq i} x_{ij} = 1 \qquad \forall\ i\ \in V \quad (2)$$

$$\sum_{j\ \in V\ j \neq i} x_{ij} - \sum_{j\ \in V\ j \neq i} x_{ji} = 0 \quad \forall\ i \in V \quad (3)$$

$$\sum_{j \in V} x_{0j} = 1 \quad (4)$$

$$\sum_{j \in V} x_{0j} = |K| \quad (5)$$

$$u_j \leq u_i - q_j x_{ij} + Q(1 - x_{ij}) \quad \forall\, i, j \in V \setminus \{0\}, i \neq j \quad (6)$$

$$u_0 = Q \quad (7)$$

$$u_i \geq 0 \qquad \forall\, i \in V \setminus \{0\} \quad (8)$$

$$x_{ij} \in \{0, 1\} \;\forall\, i, j \in V \quad (9)$$

The objective function (1) minimizes the total cost. This value varies depending on the distances involved in each particular problem in the connections of the customers and the depot. Equations (2) and (3) ensure that each customer is visited only once. Constraints (3) allows the continuous flow between all nodes of the problem, which is both functional connections between clients and the routes with the depot. While (4) and (5) allow the depot has output connections with customers of the problem and that each of these connections does not exceed the maximum number of vehicles k of the problem, respectively. Constraints (6) record a vehicle's remaining load level based on node sequence. If the vehicle visits node i right after node j, the first term in the right-hand-side reduces the vehicle remaining load after leaving node i based on the demand at node i (if node i is the candidate site node, the demand at node i is zero). Otherwise, constraints (6) are relaxed. Constraints (7) ensure that the remaining load of vehicle is equal to its capacity Q when it leaves the depot. The remaining load must be nonnegative in constraints (8). And the condition (9) defines the connection variables x_{ij} as binary, because it can only assume the value of 0 or 1.

On the other hand, the problem of traveling salesman (TSP), which is addressed as a main sub-problem to test the methodology of implicit enumeration 0-1 or Balas's algorithm, has a lower level of complexity compared to the classical VRP, because their characteristics are similar, but they differ in their main objective, which in the case of the TSP, is focused on the connection of every customer through a single minimum path. The formulation of the TSP is presented as follows:

$$\sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (10)$$

$$\sum_{j \in V\, j \neq i} x_{ij} = 1 \qquad \forall\, i \in V \quad (11)$$

$$\sum_{j \in V\, j \neq i} x_{ij} - \sum_{j \in V\, j \neq i} x_{ji} = 0 \quad \forall\, i \in V \quad (12)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad (13)$$

$$x_{ij} \in \{0, 1\} \;\forall\, i, j \in V \quad (14)$$

Since the sub-tours constraints (13) represent a huge amount of constraints for the implicit enumeration 0-1 method, the use of a subroutine that identifies the sub-tours within each route is proposed. If necessary, the heuristic creates and adds the specific restrictions to remove the sub-tours, so it is not necessary to consider the numerous sub-tour constraints associated to each route.

IV. Proposed Methodology

In this section, a modified genetic algorithm of Chu-Beasley [23], [24], [25] is proposed (algorithm 1), and combined with a set of heuristics and exact techniques which would enable the algorithm to find solutions to the capacitated vehicle routing problem (CVRP) using the variant that works with homogeneous fleet and an unlimited number of vehicles. For this, the genetic algorithm must start with an initial population, which is built by using different kinds of heuristics that can generate individuals with good objective function (following (1)) and high level of diversity. For the selection process, two tournaments are held between a variable number of randomly selected individuals in the population. This gives the winners a chance to continue to the crossover stage. The resulting offspring from the crossover process goes through a stage for mutation step through local searches. As a stage of improvement, the exact method of implicit enumeration 0-1 is used to repair the intra-route codification, which is achieved by solving the Travel Salesman Problem (TSP) that is part of a CVRP sub-problem. Finally, the individual is presented to the population through the diversity and quality criteria. The process is carried out iteratively until reaching the stop criterion that was defined by a maximum number of iterations.

A. Initial population: heuristic process

For the generation of the initial population of the algorithm, is necessary to use a constructive methodology. For this specific case, the constructive proposed is represented by three heuristics that solve the problem in different ways: the generation of a big tour, the multiple solutions of the CVRP starting from each customer and multiple solutions with a factor of randomness. Hence, multiple characteristics of good quality individuals may remain in the execution of the algorithm.

The set of three heuristics are executed iteratively to generate individuals to fill the portion of the initial population that corresponds to each one of them or until the maximum number of individuals that can generate the heuristic is reached. Based on this, a set of heuristics capable to solve the CVRP was developed, for this case, the nearest neighbor algorithm (NN) for the low computational cost and a modified savings algorithm (MSA) as the one proposed by Subramanian [26] and in other proposals as Shang and Bouffanais[27][28]. The MSA was modified by adding randomness management, to provide a high level of diversity to the initial population. The third heuristic corresponds to the Lin-Kernighan-Helsgaun algorithm (LKH) [29] for the generation of a minimum path configurations with good quality.

For the stage of filling the population of the genetic algorithm, the proposed strategy its based on the generation of individuals depending on the specifics characteristics of each one of the heuristics used. In the case of LKH heuristic, the logic of a big tour is applied, where the TSP is solved with all the customers and without the depot using the Lin-Kernighan-

Helsgaun algorithm. The next step is the division of the big tour with the vehicle's capacity, starting from each customer, so the resulting set of routes for each customer as a starting point represents an individual of the population.

For the nearest neighbor algorithm, the complete CVRP is solved with all the nodes of the problem and starting every iteration from each customer, taking into account its demand and the maximum capacity of the vehicle. To build the routes of each individual, it is essential to make the respective connections only with the closest customer to the customer that is being analysed. In order to find feasible solutions, a variety of conditions must be met: (i) the next connection must be made with a node that has not been visited before; (ii) connections must respect the maximum capacity of the vehicle; (iii) The algorithm continues its search of routes, whenever there are customers who have not been visited. Otherwise, a new individual is generated with the routes found, a new client is selected to be the first customer visited and the route search starts again.

On the other hand, the modified savings algorithm is focused on generating individuals through partial solutions to the CVRP. However, in order to use all advantages of this algorithm, a two-stage strategy is proposed. The first stage is defined as the generation of the first individual, which is obtained by solving the CVRP with the original distances of the problem and following the methodology proposed by Subramanian [26], where it is possible to find an individual of good quality for the population. The next step was developed to fill the portion of the population that is assigned to the heuristic or in some cases, fill the remaining portion of the initial population (small instances). To make this possible, the algorithm must work with modified distances, which are constantly varied to add randomness to individuals. For this, the distances c_{ij} between all nodes of the problem is varied by a random factor $\alpha = \{0.00, 0.05, 0.10, \dots, 1.65, 1.70\}$ that varies in each iteration. The modified distance $c'_{ij} = \alpha * c_{ij}$, and wherein each iteration is generated a set of modified distances to determine routes that represent partial solutions that will become part of the initial population.

B. Selection

For the selection of individuals from the population, two stages are responsible to define the two best individuals, the result of two tournaments which are carried out with the random selection of a variable number of parents who enter to be compared by their respective objective function [23]. The winner of the tournament will continue with its respective population's identifier to be taken to the crossover stage.

C. Crossover

Based on the fact that the individuals resulting from the selection stage may be considered as a permutation of the problem's customers, it is necessary to analyse different recombinant techniques, in order to adapt our crossover in the best way possible to take full advantage of the routes that have each parent. For that reason, recombinant techniques were reviewed, such as, the crossover with as a single point, two-point crossover and multi-point [23] crossover. They were also analysed as crossover, where specialized routes or parts of routes are exchanged between individuals. One of the main objectives of this step is to allow propagation of the attributes of good-quality parents present in the crossover step. To make this possible, it is necessary to follow a sequence of steps that are based on the random selection of one of the two parents (algorithm 2), and then continue with the logic presented below:

1) *Select a random route of the selected parent.*

2) *Pass a complete route (if possible) or any part of the parent's route that are not included in the offspring's sequence.*

3) *Select the other parent and repeat until all the offspring is complete.*

D. Mutation

The offspring obtained from the crossover sequence is subjected to a variable number of mutations, corresponding to typical movements for local search studied in literature, focused on search solution space through intra-route [30] changes. For this step, two stages allow the algorithm 3 do a search in the neighborhoods of solution space and perform a number of permutations of the offspring's configuration. In this way, it is possible to add diversity to the population and avoid being trapped in local optima prematurely. In this solution, instead of the typical movements reported by literature, a set of specialized movements is employed. Every move is performed only if the change leads to an improvement in the objective function. The mentioned movements are listed below:

- **Shift (1, 0):** A customer is transferred from its original route to another location in a different route.
- **Shift (2, 0):** Two consecutive clients are transferred from their current positions inside the route to other positions in different routes, keeping the intermediate arc that connect them.
- **Swap (1, 1):** Two different customers of different routes exchange their position.
- **Swap (2, 1):** The movement of an unbalanced exchange, obtaining a change of a pair of consecutive customers with one customer of a different route, where the intermediate arc in the pair of consecutive customers is preserved.
- **Swap (2, 2):** This movement follows the logic applied in the Swap (1, 1) resulting in the exchange of customers between consecutive pairs and two different routes, where the arc that connects each pair of customers is preserved.
- **Cross:** The exchange between two route segments.
- **Multi-Shift (1, 0):** A perturbation process is performed, inserting (insertions such as, Shift (1, 0)) random customers from their original routes to different routes, consecutively.
- **Multi-Swap (1, 1):** Two random customers are exchanged from their original routes a random number of times.

E. Improvement Stage: Methodological enumeration 0-1 or implicit algorithm Balas

The main reason for using specialized 0-1 implicit enumeration algorithms, which usually combined the technique of nested Benders decomposition to solve problems with integer and continuous variables, is that, unlike what happens with linear programming (LP), there is no integer programming algorithm that allows to solve all types of problems, because the efficiency of a LP algorithm depends largely on the particular characteristics of the problem. Therefore, there is no LP algorithm that is better than the other to solve all problems, or computer programs available with such features.

In this context, one of the first stages of a research project is to select a method to solve the problems resulting LP, and this selection should be made considering the different methods that exist in the literature such as: cutting-plane method, Branch & Bound, implicit enumeration, suboptimal methods, etc. At a later stage, a specialized algorithm should be developed that exploits the specific characteristics of the problem (algorithm 4).

Notation:

$+j\ indicates\ that\ the\ variable\ x_j\ =\ 1$

$-j\ indicates\ that\ the\ variable\ x_j\ =\ 0$

An underscore as $\underline{j}, x_j$ element indicates that the variable to the alternative $x_j = 0$ and was explored and probed.

Partial solution (J): It is an ordered set which values are defined as binary to a subset $j \subseteq N$. For example:

$$J = \{6, -2, \underline{-4}, 5\}$$

Free variables ($N - J$): They are variables that have not yet assigned a binary value to a partial solution and, therefore, is available to assume a value 0-1.

Complement of J: The set of solutions obtained from J assigning all variables that are still free binary values 0-1.

Partial pruned solution: A partial solution and can be pruned if all complement of J can be discarded for not being interesting.

Glover's implicit enumeration scheme: *In the algorithm of Balas, the best possible solution is stored. The list of 2^n possible solutions is analyzed implicitly or explicitly, the last best feasible solution found, named incumbent, is the optimal solution.*

1) Pruning tests

The pruning tests are designed to exclude the maximum possible complements (i.e., derived solutions) of a partial solution for neither improving the objective function nor the feasibility (not interesting solutions). These tests are mainly of heuristic type therefore they may be so weak that allow explicit enumeration of almost all 2^n feasible solutions or so powerful that exclude virtually all solutions not interesting.

At iteration t, J_t is the partial solution, then we have:

$$S_i^t = b_i - \sum_{j \in J_t; j>0} a_{ij}\,; i \in M \qquad (15)$$

$$z^t = \sum_{j \in J_t; j>0} c_j \qquad (16)$$

where S_i^t defines the value of the slack variables and z^t the objective function. Be z_{min} the best feasible solution found, so-called incumbent. The idea of probing tests, in an attempt to exclude a set of possible solutions because they are not considered interesting, can be based on two basic considerations. For a partial solution defined by J_t assume that $S_i^t < 0$ for at least one $i \in M$.

Test 1 (Balas):

It defines:

$$A_t = \{j \in (N - J_t) \mid a_{ij} \geq 0, \forall\, i \mid S_i^t < 0\} \qquad (17)$$

The elements in A_t are those free variables that being raised to 1 do not improve the infeasibility of the current partial solution.

Be:

$$N_t^1 = (N - J_t) - A_t$$

If $N_t^1 = \emptyset$ it means that no free variables can be raised to 1, then J_t is pruned by infeasibility and a backward movement must be performed.

Test 2 (Balas):

It is a test for exclusion of variables, in the vector of free variables, using an optimization criterion, where:

$$B_t = \{ j \in N_t^1 \mid (z^t + c_j) \geq z_{min} \} \qquad (18)$$

The elements B_t free variables are those that improve the infeasibility of the problem. Each variable carries an objective function greater than z_{min} incumbent and this is a worst quality objective function. Thus, $x_j, j \in B_t$ are excluded as candidates to assume a value of 1 because they are not interesting for optimality.

Be:

$$N_t^2 = N_t^2 - B_t$$

If $N_t^2 = \emptyset$, J_t is pruned because it has no better complement feasible and must performed a backward movement.

Test 3 (Balas):

It is a pruning test, where:

$$C_t = \left\{ i \in M \;\middle|\; S_i^t < 0; \sum_{j \in N_t^2} a_{ij}^- > S_i^t \right\} \qquad (19)$$

$$a_{ij}^- = min(0, a_{ij})$$

If $C_t \neq \emptyset$, at least one restriction will remain infeasible,

then J_t is pruned, because it has no feasible complement. It must be performed a backward movement. If $C_t = \emptyset$, the pruning test continue.

Test 3' (Glover-Zionts):

This test assesses whether each variable $x_j \in N_t^2$ that is promoted to 1 will cause an increase in the objective function beyond the relationship permissible for the restriction violated [31].

For each $S_i^t < 0$ calculate:

$$r_i = min_{j \in N_t^2} \left\{ \left(\frac{S_i^t}{a_{ij}^-} \right) C_j ;\ a_{ij}^- < 0 \right\} \qquad (20)$$

Where $\left(\frac{S_i^t}{a_{ij}^-} \right) C_j$ is the relative cost of each variable x_j, regarding the degree of reducing the infeasibility of that variable in the violated constraint i. If $r_i \geq (z_{min} - z^t)$ then J_t is pruned.

2) Forward movements

When pruning tests fail, the number of variables in J_t should be increased. That is, some free variables must take defined values and should be included in the set J_t assuming specific values either 0 or 1.

Test of Geoffrion:

This expansion test attempts to set values of the free variables into 0 or 1 in order to ensure feasibility, identifying essential variables for the problem [32]. However, this does not guarantee that the movement promote variables to the partial solution (may fail). The test is formulated as follows:

For each *i* such that $S_i^t < 0$ and every $j \in N_t^2$, if:

$$\left[S_i^t - \sum_{j \in N_t^2} min(0, a_{ij}) - |a_{ij}| \right] < 0 \qquad (21)$$

Then:

$$x_j = 0 \text{ if } a_{ij} > 0 \text{ or } x_j = 1 \text{ if } a_{ij} < 0$$

Test of Glover-Zionts:

This test allows a free variable set at 0 through

a very simple test [33], [31] formulated as follows:

For each *i* and $p \in N_t^2$ such that $a_{ip} > S_i^t$, calculate,

$$c_h = min\{c_j \mid j \in N_t^2 - \{p\}, a_{ij} < 0\} \qquad (22)$$

If $c_h + c_p \geq z_{min} - z^t$ then $x_p = 0$

Test of Balas:

It selects a variable x_{j^*}, $x_{j^*} \in N_t^2$ to take the value of 1 and add $j^*$to J_t. This variable is selected by the relationship:

$$v_{j^*} = max_{j \in N_t^2}\{v_j\} \qquad (23)$$

Where:

$$v_j = \sum_{i \in M} min(0, S_i^t - a_{ij})\ ;\ j \in N_t^2 \qquad (24)$$

If $v_{j^*} = 0$, there is no infeasibility and so J_{t+1} is feasible and this new solution should lead to better value z_{min}, the incumbent. So J_{t+1} is pruned and the incumbent must be updated.

3) Backward move

If a partial solution has been pruned, then a backward move must be made in the scheme of Balas. This means that the value of one of the variables of J_t must be modified. In the normal scheme of Balas, this process is performed with the LIFO (last-in, first-out) rule. With LIFO, the last variable to enter the list is the first considered for further exploration. For example, let J_t.

$$J_t = \{2, \underline{-5}, 3, 6, 1, \underline{4}\}$$

If J_t is pruned, the new partial solution is defined for the relation:

$$J_t = \{2, \underline{-5}, 3, 6, \underline{-1}\}$$

Evidently, the order in which variables are analysed alters the process enumeration. This fact was noted by Tuan who showed it is not necessary to strictly follow the LIFO [34] rule implicitly embedded in the numbering scheme Glover, to ensure proper development of numbering scheme Glover. Suggests Tuan determining a subset J'' consists of variables that can be selected for the development of future scans, this that is, those variables that can be underlined. If j_1 is the element J_t would be selected by the rule LIFO, then the elements of J'' are those elements that are located J_t from j_1 even to find the first highlighted in a section of element right to left in the elements of Jt.

F. Methodology for sub-tours identification

The methodology of implicit enumeration has a set of pruning and expansion movements, which allow it to realize a reduction of the search space of the problem. However, to to solve the TSP is necessary to use a model full of the problem with all sub-tour constraints that prevent isolated customer groups in the final solution.

This group of constraints also represents a huge burden on the constraint matrix of the method. Therefore, the determination of a subroutine responsible for identifying isolated customer groups was necessary.

As a result, it was found that the implicit enumeration algorithm ensured a global optimum if all sub-tour constraints are included for small and medium instances. Nevertheless, at a high computational cost. Hence, a modified and relaxed model without sub-tours constraints was implemented instead of using the extensive one. To make that possible, a subroutine to identify sub-tours that would reduce the number of sub-tour restrictions that were strictly necessary was added. This achieves a good compromise between processing time and quality responses.

V. Pseudocode of the Proposed Methodology

Algorithm 1. GACBIE

```
procedure GACBIE(Instance)
    LKH ← LinKernighanTSP(Customers)
    NN ← NearestNeighborTSP(Customers)
    MS ← ModifiedSavingsTSP(Customers)
    Population ← GeneratePopulation(LKH, NN, MS)
    P* ← worstCromosome(Population)
    F* ← f(P*)
    *P ← bestCromosome(Population)
    *f ← f(*P)
    P ← ∅
    f ← ∞
    for i ← 0 to MaxIter do
```

```
        P ← Crossover(Population, seed)
        P ← Mutation(P, seed)
        SubTour ← ∅
        while isFeasible(P) ≠ TRUE do
            Model ← GenerateReducedModel(P, SubTour)
            P ← BalasAlgorithm(Model, P)
            SubTour ← findSubTour(P)
        end while
        if diversityCriteria(P) = TRUE then
            if ( f(P) < F*)
                AddToPopulation(Population, P, P*)
                P* ← worstCromosome(Population)
                F* ← f(P*)
            end if
        else
            if ( f(P) < *f)
                AddToPopulation(Population, P, *P)
                *P ← P
                *f ← f(*P)
            end if
        end if
        end for
    bestP ← findBest(Population)
    return bestP
end procedure GACBIE
```

```
Algorithm 2. Crossover
procedure Crossover(Population, seed)
    ParentSet_1 ← SelectRandomParents(Population, seed)
    ParentSet_2 ← SelectRandomParents(Population, seed)
    Paren_1 ← FindBestObjetiveFunction(ParentSet_1)
    Paren_2 ← FindBestObjetiveFunction(ParentSet_2)
    while AllCustomersIncluded(P) ≠ TRUE do
        R ← SelectRandomRoute(Parent_1, Parent_2)
        P ← P ∪ R - (P ∩ R)
    end while
    return P
end procedure Crossover
```

```
Algorithm 3. Mutation
procedure Mutation(s, seed)
    s' ← s
    iter ← 0
    s* ← s
    f* ← f(s)
    while iter to MaxMutIter do
        s ← LocalSearch(s)
        if f(s) < f(s') then
            s' ← s
            iter ← 0
        end if
        iter ← iter + 1
        s ← Perturbation(s, seed)
    end while
    if f(s') < f* then
        s* ← s'
        f* ← f(s')
    end if
    return s*
end procedure Mutation
```

```
Algorithm 4. BalasAlgorithm
procedure BalasAlgorithm (Model, Path)
    M ← Model
    J_t ← InitialSolution(M, Path)
    NodeStack ← ∅
    NodeStack ← J_t
    while NodeStack ≠ ∅ do
        if isFeasible(J_t, M) then
            if f(J_t)< f* then
                f* ← f(J_t)
                J* ← J_t
                J_t ← backwardMove(J_t, M, NodeStack)
        else
            if PruningTest(J_t) = TRUE then
                J_t ← backwardMove(J_t,M, NodeStack)
            else
                J_t ← forwardMove(J_t, M, NodeStack)
            end if
        end if
    return J*
end procedure BalasAlgorithm
```

VI. COMPUTATIONAL EXPERIMENTS

As a final step, computational tests are done to analyse the performance, behavior and contribution of the GACBIE methodology. Results are compared to BKS (Best Known Solution).

The implementation of the proposed methodology was done in C ++ language under the G++ compiler that is part of a set of free license compilers from the collection of GNU compilers (GCC). The operating system used was Ubuntu 14.04 with kernel version 3.14, on an Intel Core i5 3.2 GHz and 8 GB RAM machine. Below there is a comparison between the best solutions obtained in the literature, for instances of CVRP, as also implementation of the various proposals for implicit enumeration 0-1 method.

A. Parameter setting

The correct behavior of the proposed algorithm is directly dependent on the parameters used, which is the reason to make a parameter assignment, so that in this way it is possible to determine a range in which the methodology can vary with a semi-random form.

In the development process of the algorithm, some parameters come to dominate the behavior of the algorithm. This is because some stages as mutation and improvement processes are highly structured, while the selection step depends on the number of individuals into competition. In this regard, some authors like [36] warn that a large number of individuals can make the algorithm too elitist. So the values assigned are intended to allow the range to preserve the diversity of algorithm individuals avoiding taking very good quality repetitively.

In addition, portions of the initial population assigned to the heuristics are presented in an equitable way for the LKH and NN, because they are algorithms that makes low use of computing resources.

While the modified savings algorithm, has a much smaller portion of the population since the algorithm can take much longer to generate individuals, given their division in stages, there is a very high probability that the best individual who can be generate by the algorithm, is present in the generation of the first stage.

TABLE I. PARAMETER SETTING

Description	Range/Value
Population size	60
LKH	40%
NN	40%
MSA	20%
Selection	2-3
MaxIter	100-1000

Table I shows, respectively, the values for: the population size, the corresponding portions of the Lin-population Kernigan heuristics (LKH), nearest neighbor (NN) and the modified savings algorithm (MSA), the selection range (the number of individuals entering the tournament) and MaxIter range (iterations number range), which may be increased depending on the size of each instance.

B. Computational results

1) GACBIE vs GACB and GACBC

Table II summarizes the results for three algorithms: the GACBIE (with implicit enumeration method as step improvement), the genetic algorithm with CPLEX® as improvement step (GACBC) and the same genetic algorithm without implicit enumeration but heuristics (nearest neighbor and simple inter-routes exchange) as improvement stage (GACB). All the algorithms were tested with 20 instances proposed by Augerat *et al.* (1995) (available at [35]) and compared with BKS to measure performance and effectiveness in the search for the optimal response. In the first two columns of Table 2, the instance name used followed by the number of customers (n) followed by the results of tests with the GACBIE, GACBC and GACB, accompanied by the GAP (denoted by equation (25)) between the BKS and the best solution reached each methodology, the average responses and their execution time. Finally The last column represents the number of iterations for execution.

TABLE II. RESULTS OF THE TEST ON INSTANCES OF AUGERAT ET AL

Instance	n	BKS	GACBIE				GACBC				GACB				Iterations
			Best	*Avg.*	*GAP(%)*	*Time (min)*	*Best*	*Avg.*	*GAP(%)*	*Time (min)*	*Best*	*Avg.*	*GAP(%)*	*Time (min)*	
A-n32-k5	32	784	784	784.00	0.0	6.12	784	784.00	0.0	1.12	784	784	0.0	0.11	300
A-n33-k5	33	661	661	662.00	0.0	6.40	661	662.00	0.0	1.40	662	663.89	0.15	0.09	300
A-n33-k6	33	742	742	742.00	0.0	5.58	742	742.00	0.0	2.58	742	742.78	0.0	0.08	300
A-n34-k5	34	778	778	778.08	0.0	5.81	778	778.00	0.0	1.81	778	779.05	0.0	0.08	300
A-n36-k5	36	799	799	810.15	0.0	6.41	799	816.06	0.0	2.41	817	822.47	2.25	0.13	300
A-n37-k5	37	669	669	674.07	0.0	8.66	669	674.71	0.0	3.66	677	681.26	1.196	0.15	300
A-n37-k6	37	949	949	956.26	0.0	9.71	949	952.00	0.0	6.71	960	963.02	1.159	0.16	600
A-n38-k5	38	730	730	745.56	0.0	10.32	730	747.00	0.0	8.32	756	759.67	3.562	0.17	600
A-n39-k5	39	822	822	833.60	0.0	18.84	822	834.00	0.0	9.84	842	843.11	2.43	0.24	600
A-n39-k6	39	831	831	834.39	0.0	18.36	831	835.97	0.0	10.36	850	852.01	2.28	0.22	600
A-n44-k6	44	937	946	955.09	1.07	20.26	946	953.34	1.07	12.26	952	954.86	1.60	0.34	600
P-n16-k8	16	450	450	450.00	0.0	1.13	450	450.00	0.0	0.12	450	452.42	0.0	0.02	200
P-n19-k2	19	212	212	212.00	0.0	1.30	212	212.00	0.0	0.10	212	212	0.0	0.04	200
P-n20-k2	20	216	216	216.00	0.0	2.25	216	216.00	0.0	0.13	216	220.01	0.0	0.04	200
P-n21-k2	21	211	211	211.00	0.0	2.05	211	211.00	0.0	0.20	211	215.62	0.0	0.05	200
P-n22-k2	22	216	216	218.35	0.0	2.58	216	218.35	0.0	0.60	216	225.36	0.0	0.05	200
P-n22-k8	22	590	590	593.84	0.0	2.23	590	593.14	0.0	0.23	590	595.12	0.0	0.05	200
P-n23-k8	23	529	529	529	0.0	2.80	529	529	0.0	0.40	529	529.87	0.0	0.06	200
P-n40-k5	40	458	462	462	0.8	18.18	462	462	0.8	10.11	462	504.9	0.04	0.30	600
P-n45-k5	45	510	523	525.65	2.5	26.8	523	520.06	2.5	9.06	523	613.5	0.15	0.37	600
				Avg. GAP	**0.2185**			**Avg. GAP**	**0.2185**			**Avg. GAP**	**0.74085**		

$$gap = \frac{Best_{Sol} - BKS}{BKS} * 100 \qquad (25)$$

2) NIC vs NIRSD

The strategy to reduce sub-tours constraints for routes of VRP that are repaired by the methodology of implicit enumeration, proves to be an interesting idea to reduce computational time for large problems. As observed in Table III, the computational time reduction that occurs with the increase of size of the problems is due to the implicit enumeration methodology is responsible for testing each of the possible connections possible that cannot be pruned for their selection strategies variables.

To revise the advantage of using a strategy of identifying sub-tours, tests were conducted where it was possible to make a comparison of computation times between the method of implicit enumeration with the full model including restrictions on sub-tours (NIC), the same method of implicit enumeration constrained dynamic sub-tours (NIRSD) and the speed up gap (denoted by equation (26)), wherein the steps of perturbation and crossover alter the continuity of each of the routes of the customer involved.

$$gap = \frac{NIC\_time}{NIRSD\_time} \qquad (26)$$

TABLE III. NIC VS NIRSD RESULTS

Test	n	NIC(s)	NIRSD(s)	GAP(times)
1	5	2.02	5.5	-0.36
2	6	5.76	8.12	-0.7
3	7	8.04	10.28	0.78
4	8	18.22	14.97	1.21
5	9	37.19	28.43	1.31

VII. CONCLUSIONS

At the end of the proposed project, a series of results found through the tests conducted in the modified genetic algorithm method relied on a technique of implicit enumeration stage 1-0 improvement implemented for the routing problem of trained vehicles with homogeneous fleet with diverse neighborhoods and a set of perturbation techniques. Results could be obtained the following observations:

- Replacing the typical crossover combinations by a crossover based on route exchange proved to be a very efficient strategy to keep good characteristics of the routes obtained in the population.
- By using the implicit enumeration technique the use of commercial solvers is avoid, which is an advantage in terms of cost for the implementation.
- Compared to those techniques that have to solve matrixes (Simplex, for instance) the Balas's technique is less complex since it has to realize simple algebraic operations.
- Computational experiments show that the algorithm proposed is able to obtain high quality solutions within reasonable computing times.
- Adding all restrictions on sub-tours is prohibitive for the slave problem as shown in the table III. It is more convenient to add only sub-tours constraints necessary.
- The results suggest that the proposed algorithm can be applied to other variants of problem routing vehicle, considering the inclusion of multiple depots, and use fleets heterogeneous vehicles, and the possible addition of techniques they can further narrow is the solution space.
- The use of genetic algorithm with CPLEX® compared with the methodology of implicit enumeration, shows equivalent results in reasonable computer times, considering that the solver CPLEX® has strong pruning techniques, allowing the solver to reduce the dimension of the solution space in a quickly and efficiently way. Compared with the proposed methodology, it shows a very good performance since the technique achieves global optimal in traveling salesman problems assigned. Furthermore, the use of the genetic algorithm with simple heuristics movements as improvement stage demonstrate the need for a strong methodology to repair the connections that can be generated at the stage of mutation and crossover.

REFERENCES

[1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," Management Science, vol. 6, pp. 80–91, 10 1959. 195

[2] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," J. ACM, vol. 7, no. 4, pp. 326–329, Oct. 1960. [Online]. Available: http://doi.acm.org/10.1145/321043.321046 195

[3] Christofides, N., M;ingozzi A. Toth P. (1981) "Exact algorithms for the Vehicle Routing Problem,based on Spanning Trees and Shortest Path Relaxations". Mathematical Programming 20.Pp 255-282.

[4] Christofides, N., M;ingozzi A. Toth P. (1981) "Exact algorithms for the Vehicle Routing Problem,based on Spanning Trees and Shortest Path Relaxations". Mathematical Programming 20.Pp 255-282.

[5] Bodin L., Golden B, Assad and M. Ball, (1983) "Routing and scheduling of vehicles and crews: The state of the art", Computers and OperationsResearch, 10(2),, 63-211.

[6] Fisher, M. and Jaikumar, R. (1981).A generalized assignment heuristic for vehicle-routing.Networks, 11(2):109–124.

[7] G. Laporte, H. Mercure, and Y. Nobert, "An exact algorithm for the asymmetrical capacitated vehicle routing problem," Networks, vol. 16, no. 1, pp. 33–46, 1986. [Online]. Available: http://dx.doi.org/10.1002/net.3230160104 195

[8] P. Toth and D. Vigo, Eds., The Vehicle Routing Problem. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001. 195, 197

[9] Golden L., T. Wong. (1981) "Capacitated arc routing problems." Networks, 11:305-315.

[10] R. Romero and A. Monticelli, "A zero-one implicit enumeration method for optimizing investments in transmission expansion planning," Power Systems, IEEE Transactions on, vol. 9, no. 3, pp. 1385–1391, Aug 1994. 196, 199

[11] Clarke, G., Wright, J.,. "Scheduling of vehicles from a central depot to a number of delivery points". Operations research, 1964 12 (4), 568–581.[online] available: http://pubsonline.informs.org/doi/abs/10.1287/opre.12.4.568

[12] A. Pessoa, M. P. de Aragão, and E. Uchoa, "Robust branch-cutand-price algorithms for vehicle routing problems," in The Vehicle Routing Problem: Latest Advances and New Challenges, ser. Operations Research/Computer Science Interfaces, B. Golden, S. Raghavan, and E. Wasil, Eds. Springer US, 2008, vol. 43, pp. 297–325. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-77778-8_14 195

[13] G. Laporte, Y. Nobert, S. Taillefer, "A branch-and-bound algorithm for the asymmetrical distance-constrained vehicle routing problem, Mathematical Modelling", Volume 9, Issue 12, 1987, Pages 857-868, [Online]. Available: http://dx.doi.org/10.1016/0270-0255(87)90004-2.

[14] Christofides, N., and S. Eilon. "An Algorithm for the Vehicle-dispatching Problem". *OR* 20.3 1969: 309–318. [online] available: http://www.jstor.org/stable/3008733

[15] Christofides, N., Mingozzi, A., Toth, P., 1979. The vehicle routing problem. In: Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (Eds.), Combinatorial Optimization. Vol. 1. Wiley Interscience, pp. 315–338.

[16] Rochat, Y., Taillard, E. D.,. "Probabilistic diversification and intensification in local search for vehicle routing". Journal of heuristics 1, 1995 (1), 147–167. [online] available: http://dx.doi.org/10.1007/BF02430370

[17] Fukasawa, R., Longo, H., Lysgaard, J., Poggi de Aragao, M., Reis, M., Uchoa, E., Werneck, R.. Robust branch-and-cut-and-price for the capacitated vehicle routing problem, Mathematical programming, 2006, vol. 106, 3, pp. 491–511.

[18] Baldacci, R., Christofides, N., Mingozzi, A., 2008. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. Mathematical Programming 115 (2), 351–385

[19] Contardo, C., Martinelli, R., 2014. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. Discrete Optimization 12, 129–146.

[20] Røpke, S., 2012. Branching decisions in branch-and-cut-and-price algorithms for vehicle routing problems. Presentation in Column Generation 2012.

[21] Vidal, T., Crainic, T. G., Gendreau, M., Prins, C., 2014. A unified solution framework for multi-attribute vehicle routing problems. European Journal of Operational Research 234 (3), 658 – 673.

[22] Laporte, G. (2007) What you should know about the vehicle routing problem. NavalResearch Logistics 54, 8, 811-819

[23] P. C. Chu and J. E. Beasley, "A genetic algorithm for the generalized assignment problem," Comput. Oper. Res., vol. 24, no. 1, pp. 17–23, Jan. 1997. [Online]. Available: http://dx.doi.org/10.1016/S0305-548(96)00032-9 199, 201

[24] J. H. Holland, Adaptation in Natural and Artificial Systems. Cambridge, MA, USA: MIT Press, 1992. 199

[25] G. Syswerda, "Schedule optimization using genetic algorithms," in Handbook of Genetic Algorithms, L. Davis, Ed. New York, NY: Van Nostrand Reinhold, 1991. 199

[26] A. Subramanian, P. H. V. Penna, E. Uchoa, and L. S. Ochi, "A hybrid algorithm for the heterogeneous fleet vehicle routing problem," European Journal of Operational Research, vol. 221, no. 2, pp. 285 – 295, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221712002093 200

[27] Y. Shang and R. Bouffanais, "Consensus reaching in swarms ruled by a hybrid metric-topological distance," *Eur. Phys. J. B*, vol. 87, no. 12, pp. 1–8, 2014.

[28] Y. Shang and R. Bouffanais, "Influence of the number of topologically interacting neighbors on swarm dynamics.," *Sci. Rep.*, vol. 4, p. 4184, 2014.

[29] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," Operations Research, vol. 21, no. 2, pp. 498–516, 1973. [Online]. Available: http://dx.doi.org/10.1287/opre.21.2.498 200

[30] P. Penna, A. Subramanian, and L. Ochi, "An iterated local search heuristic for the heterogeneous fleet vehicle routing problem," Journal of Heuristics, vol. 19, no. 2, pp. 201–232, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10732-011-9186-y 202

[31] F. Glover and S. Zionts, "A note on the additive algorithm of balas," Operations Research, vol. 13, no. 4, pp. 546–549, 1965. [Online]. Available: http://dx.doi.org/10.1287/opre.13.4.546 206, 207

[32] A. M. Geoffrion, "Integer programming by implicit enumeration and balas' method," SIAM Review, vol. 9, no. 2, pp. 178–190, 1967. [Online]. Available: http://www.jstor.org/stable/2027441 206

[33] F. Glover, "A multiphase-dual algorithm for the zero-one integer programming problem," Operations Research, vol. 13, no. 6, pp. 879–919, 1965. [Online]. Available: http://dx.doi.org/10.1287/opre.13.6.879 207

[34] N. P. Tuan, "A flexible tree-search method for integer programming problems," Operations Research, vol. 19, no. 1, pp. pp. 115–119, 1971.[Online]. Available: http://www.jstor.org/stable/168870 208

[35] "NEO Research Group," http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/, accessed: 2014-01-12. 211, 213

[36] Rendon, R.A.G. and Zuluaga, A.H.E. and Ocampo, Eds., Tecnicas metaheuristicas de optimizacion, Universidad Tecnologica de Pereira, 2008. 100, 150, 9789587220070.