Available online at www.sciencedirect.com
ScienceDirect
Omega 37 (2009) 138 – 154
omega
The International Journal of Management Science
www.elsevier.com/locate/omega

A visual interactive approach to classical and mixed vehicle routing problems with backhauls[☆]

G. Yazgı Tütüncü[a,*], Carlos A.C. Carreto[b], Barrie M. Baker[c]

[a]Department of Mathematics, Faculty of Science and Literature, İzmir University of Economics, Sakarya Caddesi, 35330 İzmir, Turkey
[b]Deparmento de Informatica, Instituto Politecnico da Guarda, Av.Dr. Francisco Sa Carneiro, 506301-559 Guarda, Portugal
[c]Faculty of Engineering and Computing, Decision Support (Statistics & Operations Research) Group, Coventry University, Armstrong Siddeley Building, Priory Street, CV1 5FB Coventry, UK

Received 19 October 2005; accepted 6 November 2006
Available online 2 January 2007

Abstract

In this paper a new visual interactive approach for the classical vehicle routing problem with backhauls (VRPB) and its extensions is presented. The classical VRPB is the problem of designing minimum cost routes from a single depot to two type customers that are known as *Backhaul* (pickup) and *Linehaul* (delivery) customers where deliveries after pickups are not allowed. The mixed VRPB is an extension of the classical VRPB where deliveries after pickups are allowed.

A decision support system (DSS) is developed in order to solve the classical VRPB, mixed VRPB and the restricted VRPB, which is a compromise problem between the classical VRPB, and the mixed VRPB. And a new criterion, which considers the remaining capacity of the vehicles, is proposed for producing solutions for mixed and restricted VRPB. The visual interactive approach that is based on Greedy Randomised Adaptive Memory Programming Search (GRAMPS) is described, and experimental results for the VRPB benchmark test problems are presented and analysed. The computational results on VRPB benchmark test problems indicated that the new criterion and the proposed visual interactive approach are effective towards finding a compromise between the mixed or restricted and the classical VRPB problems. The developed DSS is used by 18 students and reported to be capable of producing high quality solutions for the VRPB.

Keywords: Vehicle routing; Backhauls; Heuristics; Decision support systems

1. Introduction

Transportation is generally considered as being a significant factor of economic activities in any company. The problems of appropriate usage of the vehicle fleet appear as a matter of restricted resources of the company and expectations of customers and they are known as vehicle routing problems (VRP). The classical vehicle routing problem with backhauls (VRPB) is an extension of the VRP where two types of customers are served from a single depot by a fleet of vehicles. The first type of customers is known as “linehaul” customers who require delivery of goods to their specified location and the second type is known as “backhaul” customers who require pickups from their specified locations. The importance of the VRPB was addressed in an early study of Casco et al. [1] and an example in the grocery industry where the supermarkets are linehaul customers

☆ This manuscript was processed by Associate Editor Prof. B. Lev.

* Corresponding author. Tel.: +90 232 488 83 11; fax: +90 232 488 84 75.

E-mail addresses: yazgi.tutuncu@ieu.edu.tr (G. Yazgı Tütüncü), ccarreto@ipg.pt (C.A.C. Carreto), b.baker@coventry.ac.uk (B.M. Baker).

0305-0483/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.omega.2006.11.001

and the grocery suppliers are backhaul customers is given. The financial prudence of serving backhaul and linehaul customers on the same routes for the grocery industry in the USA was highlighted. In recent years, it became more obvious that in real-world applications allowing vehicles, which are returning from linehaul customers, to visit backhaul customers leads to significant saving in the distribution cost. Therefore the classical VRPB and its variants have attracted the attention of researchers.

The objective of the classical VRPB is to find a set of routes with the following features that minimises the total distance travelled by the vehicle fleet:

- All customers must be visited and each customer must be visited by only one vehicle.
- Each vehicle is restricted by capacity constraints and the combined loads associated with linehaul and backhaul customers of each vehicle must not violate the given capacity constraints.
- Vehicles may or may not be restricted by distance constraints.
- Each vehicle has to serve backhaul customers, if any, after all linehaul customers are served.
- The vehicle fleet size is fixed and vehicles are homogenous.

The fourth feature given in the above list defines the fact that in the classical VRPB deliveries after pickups are not allowed. In theory this restriction reduces the complexity of the problem and in practice it avoids the problems that may rise because of rearranging goods on the vehicle and supports the fact that linehaul customers have priority over backhaul customers. However, it can be easily proposed that ignoring this restriction may reduce the total travelling cost. Therefore, the mixed VRPB is defined as an extension of the classical VRPB where the constraints and the objective are the same as in the classical VRPB but deliveries after pickups are allowed. This difference makes the mixed VRPB more difficult to solve than the classical VRPB. The main reason behind this difficulty is the need to check the capacity constraints for possible violation for every arc of each route before inserting a customer into a new position on any route. In the classical VRPB it is enough to check capacity constraints violations in the corresponding part of the route (backhaul or linehaul parts) while in the mixed VRPB these capacity constraints have to be checked for every link between the customers. Therefore, although the capacity constraints seem to be similar for both problem types they became more restrictive in the mixed VRPB. On the other hand, Wade and Salhi [2] proposed a new version of the mixed VRPB to produce a practical compromise between the classical VRPB and the mixed VRPB. In this new problem, mixed linehaul and backhaul customers are permitted but the position along the routes where the first backhaul may be serviced is restricted. They refer to this type of VRPB as the restricted VRPB, where the constraints prevent the inclusion of backhaul customers until a given percentage of the total linehaul load has been delivered.

The classical VRPB, the mixed VRPB and the restricted VRPB are the special case versions of the basic VRP where two different types of customers are served. Therefore they are NP-hard and the computational effort required to solve these problem increases exponentially with the problem size. This implies that real-life problems are much larger and thus heuristics have been used to solve the classical VRP and its extensions in practice. However, most of these heuristic methods have been proposed as part of theoretical research not for real-life applications and are not necessarily suitable to implement within a computerised routing software system. Moreover they do not give users (decision makers) any opportunity to control the solution process using their judgment and they require a lot of computational time which may not allow any visual or manual improvement process. But in real-world distribution management systems, decision makers may wish to have a simple and intuitive solution method that is easy to understand and use with a nonexpert person, requiring short computational time to produce good quality solutions for their company's distribution problems.

In this paper we propose an inexpensive decision support system (DSS) based on a new Greedy Randomised Adaptive Memory Programming Search (GRAMPS) algorithm to solve the classical VRPB, the mixed VRPB and the restricted VRPB in a visual interactive environment. Furthermore a new criterion for creating good solutions for the mixed and the restricted VRPB is proposed based on the vehicle's free capacity when accepting backhauls before all the deliveries have been made, to help the user produce routes that are workable in practice. In the remainder of this paper, we review some of the best-known works in the literature. Then, the proposed method and the DSS are discussed. The paper concludes with some test results for the VRPB benchmarks given in the literature.

2. Literature review

Several exact and heuristic algorithms for the solution of the classical VRPB and its extensions have been proposed that are variations on algorithms for the VRP. In this section brief descriptions of two early methods and some recent heuristics for the VRPB and mixed VRPB are presented.

2.1. Heuristics for the classical VRPB

The classical VRPB has been studied by several researchers for more than 20 years and because of its complexity heuristics are commonly used to deal with it. In the early 1980's Deif and Bodin [3] proposed the first constructive method for the classical VRPB, which is an extension of Clarke and Wright's [4] savings algorithm. Goetschalckx and Jacobs-Blecha [5] proposed a two-phase method that is in the cluster-first route-second category for the classical VRPB with Euclidean cost matrix. Their approach is based on space-filling curves; in this implementation the space-filling transformation was used to assign a number between 0 and 1 for each customer according to its relative position on a line. This transformation was applied to the backhaul and linehaul customers separately and these two different kinds of sequence of customers were clustered in order to build feasible routes. Goetschalckx and Jacobs-Blecha applied a greedy clustering method in which linehaul and backhaul routes are built separately. After all feasible backhaul and linehaul routes were constructed each linehaul route was merged with a backhaul route that is the closest to the linehaul route according to their space-filling transformation values. They proposed a set of Euclidean VRPB test problems with 25–150 customers where 25–50% of them are backhaul. The method was tested on these problems and the results showed that although the solutions were not better than the results given in [3] the solution times were shorter especially for the large test problems.

In their second paper [6] Goetschalckx and Jacobs-Blecha used a clustering method, which is based on the generalised assignment approach proposed by Fisher and Jaikumar [7] where the number of routes K that will be constructed in the heuristic was specified in advance. The linehaul customers are sorted according to their increasing distance from the depot and the backhaul customers are sorted according to their decreasing distance to the depot. Both customer sequences are divided into K clusters by solving generalised assignment heuristics. After clustering backhaul and linehaul customers separately, linehaul and backhaul routes are merged according to the best combination of connections that has the smallest distance whilst not allowing any backhaul customer before a linehaul customer. This method was tested on the same test problems given in [5] and the results showed that the method obtained better results than the results found with the first approach given in [5].

Another two-phase method for the classical VRPB was proposed by Toth and Vigo [8]. Their approach is a cluster-first route-second heuristic and solves both symmetric and asymmetric VRPB problems in which visiting backhaul customers before linehaul customers and routes containing only backhaul customers are not allowed and the vehicles are without distance restrictions.

In the first phase of the heuristic linehaul and backhaul customers are clustered separately by a Lagrangian relaxation method that is based on finding a set of capacitated shortest paths leaving the depot and visiting all the linehaul customers, a set of capacitated shortest paths leaving the depot and visiting all the backhaul customers and a set of disjoint arcs which join the linehaul and backhaul customers. It was noted that at the end of this phase some of the clusters can be infeasible with respect to the capacity constraints but generally the clusters obtained in this phase were found to be very similar to the final solution. Toth and Vigo tested their approach on three different sets of VRPB instances. The results obtained for the first set of problems that was given in [5] were compared with the optimal solutions or the Lagrangian lower bounds and the approaches proposed by Deif and Bodin [3] and Goetschalckx and Jacobs-Blecha [5]. Toth and Vigo's approach achieved 52 best solutions for 62 test problems with longer but reasonable computational times. The results obtained for the second set of problems that was proposed by Toth and Vigo [9] were compared with the approaches proposed by Deif and Bodin [3] and Goetschalckx and Jacobs-Blecha [5] and the results showed that Toth and Vigo's approach performed better than the other two approaches. Finally, results obtained for the third set of problems that was proposed by Toth and Vigo [9] for the asymptotic VRPB problems were compared with the modified approach of Deif and Bodin [3] and also in this case Toth and Vigo's approach obtained very good solutions. The reader is referred to the book of Toth and Vigo [10] for more heuristics and detailed explanations for the VRPB.

2.2. Heuristics for the mixed VRPB

The first study for the mixed VRPB was presented by Deif and Bodin [3]. In this study they applied a saving heuristic that is based on the Clarke and Wright's [4] savings algorithm. The proposed algorithm starts with a solution in which all customers are served using different vehicles. Then these routes are combined according to the highest savings s_{ij} calculated with a condition that backhaul customers can be served after linehaul customers. But the results obtained using this algorithm showed that this condition reduces the number of feasible mergings. Therefore, a modified savings formula, called *backhaul savings* was proposed for both classical and mixed VRPB problems and it was given by

$$s_{ij}^{B}=\begin{cases} s_{ij}-p\hat{s} & \text{if } i\in L,\ \ j\notin L \text{ or vice versa},\\ s_{ij} & \text{otherwise,}\end{cases} \tag{1}$$

where $\hat{s}$ is the estimated maximum saving, L is the set of linehaul customers and $p \in [0, 1]$ is the penalty value. Deif and Bodin tested their algorithm for several values of p on randomly generated problem instances with 100–300 customers that include 10% and 50% backhaul customers. The results showed that values of p between 0.05 and 0.20 produced the best solutions.

Casco et al. [1] presented a load-based backhaul insertion algorithm that solves the mixed VRPB. In their algorithm first they generated a feasible solution using Clarke and Wright savings algorithm, which is actually a feasible classical VRPB solution, then they modified this initial feasible solution with a penalty function that controls the delivery load of each vehicle after pickups. This procedure takes the linehaul part of the feasible solution and uses the 1-insertion procedure to add backhaul customers in between linehaul customers taking into account the delivery load remaining after servicing each customer. They reported that their procedure is effective at reducing the total distance travelled by allowing a mixture of customers on a route.

Salhi and Nagy [11] proposed a number of insertion-based heuristics. The first heuristic they proposed was an extension of Casco et al. [1]'s 1-insertion procedure, the second one was a 2-insertion heuristic and the last one was a cluster insertion heuristic. In the cluster insertion heuristic they combined the 1-insertion procedure and 2-insertion procedure for creating all possible clusters of backhauls. In their algorithm first a single backhaul customer with the least insertion cost (1-insertion procedure) is found and following this a pair of backhaul customers with least insertion cost (2-insertion procedure) is found. Then, a cluster with a least insertion cost is built by considering all the results of 2-insertion procedure. After finding the single backhaul customer, a pair of customers and a cluster with the least insertion costs one of the best is inserted in the corresponding route. This procedure was repeated until all backhaul customers are inserted. These procedures were tested on benchmarks and the authors suggested that the cluster insertion heuristic could be used to create starting solutions for more complex heuristics.

Wade and Salhi [2] proposed a new version of the mixed VRPB to produce a practical compromise between the classical VRPB and the mixed VRPB. In this new problem, mixed linehaul and backhaul customers are permitted but the position along the routes where the first backhaul may be serviced is restricted. They refer to this type of VRPB as the restricted VRPB (R-VRPB), where the constraints prevent the inclusion of backhaul customers until a given percentage of the total linehaul load has been delivered. In order to solve this new backhaul problem Wade and Salhi presented a heuristic where the user is asked to set a restriction percentage (RP) on the insertion of backhaul customers. Thereupon, they defined the insertion-based procedures based on the proposed algorithm in [11] to solve R-VRPB in two steps. In the first step the linehaul customers are routed. In the second step, the backhaul customers are inserted into the route and the RP is used to control these insertions.

Wade and Salhi tested their approach on two classes of VRPB instances. The first class of the VRPB instances was taken from Goetschalckx and Jacobs-Blecha [5] and the second class of the VRPB instances was generated by Toth and Vigo [9] where each VRP instance corresponded to three VRPB instances with linehaul percentage of 50%, 66% and 80%. Each class of VRPB instances was tested for five different RPs of 0%, 25%, 50%, 75% and 100% where 0% corresponds to the fully mixed VRPB and 100% is correspondent to the classical VRPB. These values were chosen in order to show the decrease in the route travelling cost when the mix of backhaul and linehaul customers is increased. The results showed that for the first class of VRPB instances total travelling cost of vehicles can be reduced by 7% and for the second class of VRPB instances total travelling cost of vehicles can be reduced by 6% when the RP is 75% compared to their results with RP set to 100% (although the latter results are not given or compared to the best-known solutions). Wade and Salhi [2] said that they would expect similar improvements if a RP was allowed with other, more powerful, heuristics.

3. The visual interactive approach

The approach and the DSS, we propose in this paper can be described as a further development for the restricted VRPB that Wade and Salhi introduced in [2]. However, the main contribution of this paper is to propose a DSS based on a visual interactive approach for solving the classical, mixed and restricted VRPB at the same time. There are some early studies in the literature which addressed interactive solution methods for the basic VRP, such as the man–machine approach of Krolak et al. [12] and the interactive approach with pricing mechanism proposed by Cullen et al. [13]. However, as far as we are aware to date other than the one by Carreto and Baker [14] for the classical VRPB no visual interactive approach has been proposed for solving the classical VRPB and its extensions at the same time. So, apart from being a further development for an earlier study, the proposed approach is the first visual interactive approach for both mixed and restricted VRP.

The proposed visual interactive approach is based on the procedure that was developed by Fisher and Jaikumar [7] and extended by Baker [15]. In this method, the customer locations are first displayed on the graphical-user-interface and then the user can select a single seed customer to form an outline route for each vehicle [13]. Alternatively, two or more customers can be selected in the order that the user would expect them to be visited by the same vehicle, to form an outline route [15]. Baker and Carreto, extended this approach and solved the basic VRP [16] and the classical VRPB [14]. In this study we improved the approach of Baker and Carreto [14,16] and implemented it in a superior decision system environment. The new visual interactive approach implemented in the proposed decision system, namely Computerised Routing Using Interactive Seeds Entry version 2. (CRUISE2), contains three stages: *initial solution stage*, *problem modification stage* and *final solution stage*. These stages are described in the next subsections.

3.1. Initial solution stage

The initial solution stage consists of two phases; seeds selection phase and solution proposition phase. Seeds selection phase is the first phase of this stage in which the user selects the seeds for each vehicle. The seeds are the customers that for some reason the user chooses to be serviced by specific vehicles i.e. the driver of that vehicle may be more familiar with this customer than the other drivers.

In the proposed DSS the customers are represented on a visual interactive display by triangles with areas proportional to the customer demands. On screen the linehaul customers are displayed as red triangles and backhaul customers are displayed as blue triangles. In all black and white printed figures, linehaul customers appear as black and backhaul customers appear as grey triangles. The seed selection can be done in two ways: manual or automatic. In both ways throughout the seeds selection, the computer checks the capacity, the distance and the precedence constraint violations. Precedence constraints are the ones that restrict the allocation of backhaul customers before serving all the linehaul customers, i.e. linehaul and backhaul customers can be selected for the same seed but if the user attempts to select a linehaul customer after a selection of a backhaul customer, the user is warned by the computer. The seed selection procedure considers that selection of linehaul customers after a backhaul customer is not valid (precedence constraint) for the first phase of our solution method because the aim of the first stage of our method is to find a classical VRPB solution.

In manual selection the user selects the seeds for each vehicle by pointing at the customers on the screen using the mouse in accordance with his local knowledge and experience of a real-life problem, or by identifying certain patterns like clusters of customers, customers with very high demands or isolated customers. Selection of seeds continues as many times as the number of vehicles that are available. In the seeds selection step the user is helped by the system so as not to violate the capacity distance and precedence constraints defined in the beginning of the process.

In automatic seed selection, the user points at the related tool in the menu of the user interface and an algorithm selects one customer at a time as seed customers. For selecting one customer at a time we proposed a new seed selection algorithm which is based on Gillett and Miller's [17] sweep algorithm which we called Division Algorithm. A division is a portion of a circle where the depot is the centre and the distance of the farthest customer from the depot is the radius.

Let Q be the maximum capacity of a vehicle defined by the user. The number of divisions in a problem is the minimum number of the vehicles needed to serve all the linehaul customers, calculated as follows. Let T be the total linehaul demand of all the customers in a problem, the number of divisions $D^{\#}$ is calculated as follows.

$$D^{\#} = \left\| \frac{T}{Q} \right\| + 1.$$

The algorithm defines a reference point on the horizontal line (0°) that passes through the depot (centre) and selects as the first seed the linehaul customer that is in the anticlockwise direction and has the smallest angle with this reference point. Then starting from this point it moves to the next linehaul customer which has the second smallest angle between the reference point and adds the demand of the customer to the demand already allocated to the corresponding division. This continues until the total demand is more than the capacity of the vehicle. When the search reaches the customer which exceeds the vehicle capacity, it is selected as the seed for the next division.

This procedure continues until all the linehaul customers are visited. Once the user selects the automated seed selection tool from the menu, he/she will be able to select different seeds with different divisions by setting the initial angle between the horizontal line and the reference point to different values. Obviously this selection of the initial angle yields different divisions and different combinations of automated seeds. The pseudocode of the Division Algorithm is given in Algorithm 1.

Algorithm 1. Division Algorithm

```
Begin
      Initialise vehicle capacity Q;
      Calculate number of divisions D#;
      Initialise division demand = Q;
      Initialise division counter = 0;
      Initialise a reference point;
      Set the search position on the reference point;
   Repeat
      Find the closest customer j to the set position in anticlockwise direction;
      division demand = division demand + customer's demand;
      if ( division demand > Q )
      {
         select an unused vehicle r ;
         allocate customer j to vehicle r;
         division demand = 0;
         division counter = division counter + 1;
      }
      Set the position on customer j ;
   until ( division counter = D#)
end
```

The selection of the seeds can be seen as a critical phase since the seeds are the starting point for the solution method. If they are not well chosen, the solution will be poor or infeasible. It is also the seed selection that determines the number of vehicles used. Although being a critical phase, the interaction makes it easy for the user to try different seeds in order to obtain a satisfactory solution. The user can do this by selecting completely new seeds or making minor adjustments during the solution process.

The solution proposition phase is the second phase of the initial solution stage, which includes a new metaheuristic that employs a restart procedure GRAMPS. In general, GRAMPS belongs to the class of guided construction metaheuristic, which is a combination of the Greedy Randomised Adaptive Search Procedure and the Adaptive Memory Programming. GRAMPS was first proposed by Ahmadi and Osman [18] for the capacitated clustering problem and as far as we are aware up to now no researcher has attempted to develop a memory-based GRASP algorithm for solving vehicle routing problems. Each iteration of Ahmadi and Osman's [18] GRAMPS algorithm consists of three processes: construction, learning and local search. In the construction process of their proposed work they used the density search construction method, which was developed for the capacitated clustering problem, and they used a learning process, which is similar to the AMP type in Fleurent and Glover [19]. They implemented GRAMPS with a local search descent based on a restricted k-interchange neighbourhood.

The metaheuristic used in our GRAMPS employs a combination of GRASP and AMP that is different from the one proposed by Ahmadi and Osman [18] in the learning strategy. Since the algorithms proposed in the GRAMPS metaheuristic can be described as a version of GRASP it is a multi-start metaheuristic and each iteration consists of construction and local search phase but additionally continues with a learning phase.

In the construction phases, a construction heuristic allocates each customer to the position in a route where the insertion cost is minimised. The insertion cost and insertion position are calculated for each unallocated customer using the approximation for the insertion costs, which was proposed by Baker [15]. More specifically this approximate insertion cost is calculated as follows: let c_{rj} be the minimum insertion cost of customer j in route r and d_{ij} be the distance between customer i and customer j. The insertion cost is

$$c_{rj} = \min_{0 \leqslant k \leqslant K} \{d_{i_k,j} + d_{i_{k+1},j} - d_{i_k,j_{k+1}}\},$$

where i_0 and i_{K+1} represent the depot. This calculation is performed for each un-routed customer and for each route defined in the seed selection stage of the visual interactive solution method. Throughout these calculations the construction heuristic checks for any violation of the problem constraints and at the same time it records the insertion position corresponding to this minimum insertion cost of customer j in route r, denoted by p_{rj}.

After calculating the minimum insertion costs for all unallocated customers, these unallocated customers are allocated in routes one at a time in a sequential order. The order of allocation of customers is ruled by giving allocation priority to customers with a more obvious insertion route, and to customers for which the number of routes they can go on is smallest. According to this rule, customers that can be inserted in only one route are allocated first and then the customers that can be inserted in more than one route are selected from a restricted candidate list (RCL) randomly. In general the RCL consists of the best elements whose incorporation to the partial solution results in the smallest incremental costs in the final solution. In the proposed construction heuristic the RCL is formed by the first five unallocated customers with the biggest difference between the smallest and the second smallest insertion costs. The random selection of customers from RCL is done with a probability proportional to customers' demand.

Let w_j be the demand of customer j, the probability of selecting customer j as the next customer to be inserted on a route is given by

$$P(j = \text{selected}) = \frac{w_j}{\sum_{\forall i \in \text{RCL}} w_i}.$$

This selection forms the randomised aspect of our GRAMPS algorithm. At each iteration, if there is at least one customer j that cannot be allocated to any route then there is no feasible solution. However, the construction heuristic continues as the best infeasible solution achieved may be needed in the next step. If there is more than one customer that can go to one route the heuristic selects the one with the biggest difference between the smallest and the second smallest insertion cost.

Restrictions are applied to the construction heuristic in order to solve the classical VRPB, specifically the precedence constraints, that disable the insertion of a linehaul customer after a backhaul customer. For example, if the seed of a route includes a mix of linehaul and backhaul customers the construction heuristic does not consider the insertion of any unallocated linehaul customer after a backhaul customer. Similar logic is used for the 3-opt heuristic that is performed at the end of the construction heuristic. The 3-opt heuristic is proposed in order to perform exchange procedures without violating the precedence constraints. The heuristic may find a feasible solution where all the customers have been allocated to the vehicles; or the construction may result in an infeasible solution, because there was a violation of a capacity or distance constraint.

In the local search phases, a restricted variable neighbourhood search procedure, called RVNS, is applied for finding a local minimum solution starting from a feasible or an infeasible solution obtained in the construction phase. Similar to the construction heuristic, RVNS is proposed as a general-purpose heuristic for the last two phases of our GRAMPS.

The RVNS heuristic consists of two different heuristics. The first heuristic is proposed for finding a local minimum solution starting from a feasible solution obtained in the construction phase and the second heuristic is proposed for finding a local minimum solution starting from an infeasible solution obtained in the construction phase. In both of the heuristics, the well-known λ-interchange procedures proposed by Osman [20] are applied for $\lambda = 1$ and 2. The procedures are applied in sequence, first the 1-interchange and next the 2-interchange. Interchanges can be restricted to p-neighbourhoods, in which each customer j can be moved only to routes that contain at least one of the p nearest customers to customer j. In this context, the depot location is included along with the customer locations and, if the depot location is one of the p nearest locations to customer j, then j can be moved to any other route. First, each customer is considered in turn and tried in its best position in each route that is eligible. Then all pairs of routes are considered, and all customers that are eligible to be moved between these two routes are tried in pair-wise swaps between the two chosen routes. But in the one-node interchange and the two-node interchange procedures while reallocating the customers the preference constraints must be taken into account.

Two different strategies are considered for selecting the next move in each procedure: *best-improve* and

Algorithm 2. GRAMPS Approach of CRUISE2

```
Begin
  User selects initial seeds;
  Best feasible cost = ∞;
  Best infeasible cost = ∞;
      Repeat
      Start with the initial seeds;
      Construct greedy randomised solution;
      Improve the solution using RVNS heuristic;
      Keep the best_solution;
      Calculate the average_feasible_cost;
      until (Max number of feasible and infeasible solutions is met)
      Repeat
      Start with the best feasible solution found in the first loop;
      Change the seeds with maximising g;
            Repeat
            Start with the changed seeds;
            Construct a greedy randomised solution;
                  Improve the solution using RVNS heuristic;
                  Keep the best_solution_changedseeds;
                  Calculate the average_feasible_changedcost;
                  If (average_feasible_changedcost > average_feasible_cost);
                  Count the iterations with above average_feasible_cost;
            until ( Max number of above average iterations is met )
      until ( Max number of seed changes is met )
end
```

first-improve strategies, as described by many other authors (such as [20], for example). The user can choose between these two strategies before starting the solution process. In addition, the user can choose between two metrics for determining the p-neighbourhoods: the *circular* metric that uses the Euclidean distance between each pair of customers, and the *generalised savings* metric proposed by Yellow [21].

The last phase of the GRAMPS consists of a learning process. A learning process is normally designed to exploit information to guide the constructive process and it includes a memory, which collects information on good elements in a set of solutions. As mentioned before the seeds are the starting point for our solution method and if they are not well chosen, the solution will be poor or infeasible. Since the selection of the seeds can be seen as a critical phase adaptive memory plays an integral part of our GRAMPS. At the end of 100 iterations of the algorithm the best solution found is saved in the memory and at the beginning of the following iteration a new set of seeds that is obtained from this solution is used as initial seeds. For obtaining this new set of seeds first of all the route with maximum distance is determined and then an adjacent route is chosen. These are the routes whose seeds will be changed, and a pair of customers from these routes, are selected as new seeds to maximise the function given in

$$g_{ij} = \frac{\theta_{ij}}{\max \theta_{ij}} + \frac{(d_{0i} + d_{0j})}{2 \max d_{ij}}. \tag{2}$$

This function is used to define a new set of seeds in order to search different parts of the solution space but also remain committed to the user's selection. In this equation θ_{ij} represents the angle between customer i and customer j and d_{ij} represents the distance between these customers. Maximising this function helps the heuristic to define new seeds that are chosen far from the depot and each other and this structure increase the diversity of the solutions. The pseudocode given in Algorithm 2 illustrates the general structure of the new visual interactive approach and the GRAMPS method that is used in the construction and improvement phase of the approach.

Throughout the GRAMPS iterations, the display shows the best solution obtained so far (best feasible or best infeasible), and the user can terminate the

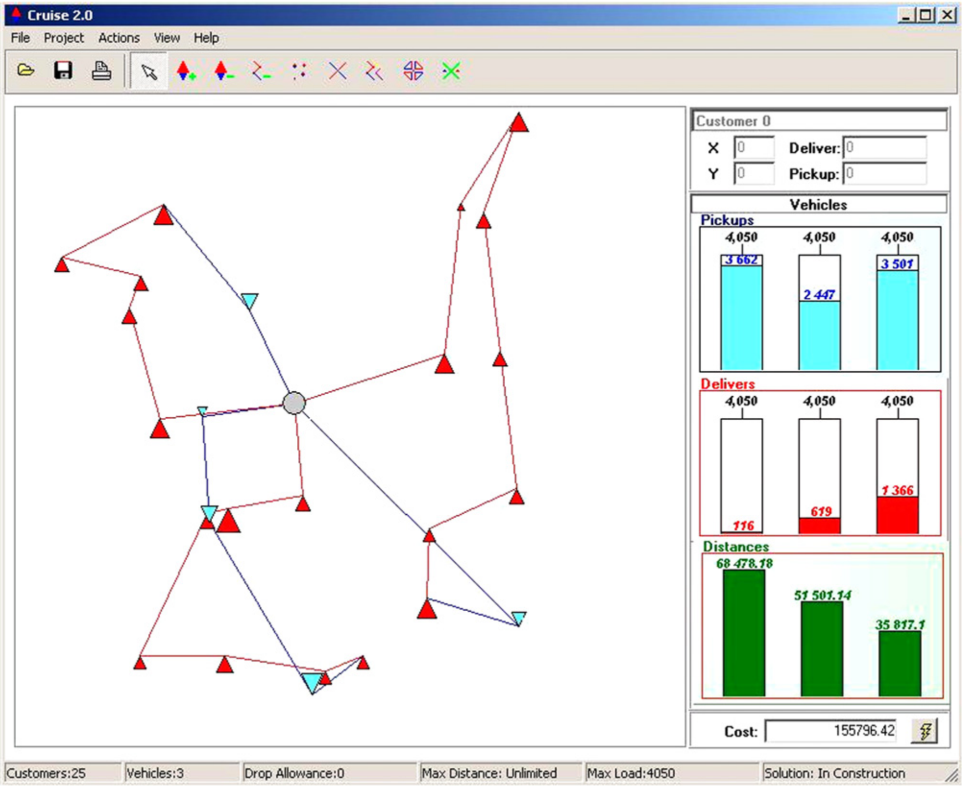

Fig. 1. CRUISE2's graphical user interface for the classical VRPB.

procedure whenever they wish to make a manual intervention in the solution process. In this context, the best infeasible solution is considered to be one for which the unallocated customers have minimal total delivery requirements, since this offers good scope for finding a feasible solution following a manual intervention. At the end of the initial solution stage, the classical VRPB solution achieved by implementing the algorithm is presented in a form designed to provide visual cues to the user. The user is allowed to try to refine the solution in order to improve or modify it regarding his\her judgements and insights, by using interactive tools. With no formal rules to follow, the CRUISE2 interactive environment is sufficiently flexible to allow creativity and imagination from the user, and different refinement techniques are possible for using these tools. The simplest example consists of the removal of some customers from their current routes, followed by a new call to the construction and improvement heuristics.

An example of a solution obtained for a classical VRPB problem instance at the end of the initial solution stage is given in Fig. 1. In this figure as we mentioned before the linehaul customers are displayed by red triangles and backhaul customers are displayed by blue triangles on the user interface but on the grey scale the backhaul customers are coloured in grey and linehaul customers are coloured in black. The information about the remaining pickup, delivery capacities and distance travelled by the vehicles are given in the user interface.

3.2. Problem modification stage

After solving the classical VRPB in the initial solution stage of our solution method, in the problem modification stage, the user is optionally allowed to construct a mixed or a restricted VRPB. For this reason the interactive tools of CRUISE2 are developed in order to allow the user to insert backhaul customers in front of linehaul customers and to guide the user by some useful information that helps the user at the time he/she tries to insert a backhaul customer in front of a linehaul customer or customers.

This information consists of two values that are called (RP) and remaining capacity (RC). RP, given in Eq. (3), is a criterion which was proposed by Wade and Salhi

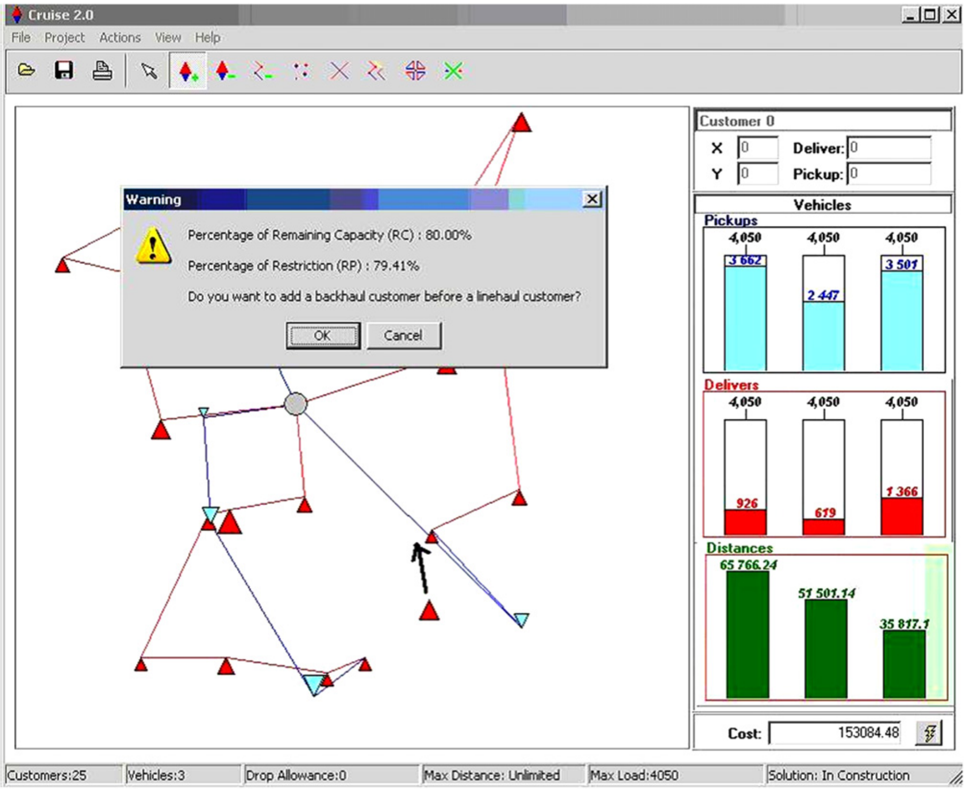

Fig. 2. Insertion of a linehaul customer after a backhaul customer in CRUISE2.

[2] for the restricted VRPB and it gives the percentage of the total linehaul load that has been delivered before the first backhaul customer on a specific route. The RP is calculated when a backhaul customer is inserted between two linehaul customers for each route. Let the total linehaul demand for route r be D^r and the linehaul load of vehicle r after serving customer i be L^r_{ij}, where i represents a linehaul customer and j represents the first backhaul customer served after this linehaul customer with vehicle r. The formula to calculate the RP on route r is denoted by $\mathrm{RP}_r(i, j)$ in Eq. (3) [2]. It can be seen that, in the case of classical VRPB the value of L^r_{ij} will be zero making the RP equal to 100%:

$$\mathrm{RP}_r(i, j) = \left(1 - \frac{L^r_{ij}}{D^r}\right).100. \tag{3}$$

RC is a new criterion that we defined besides RP, which gives the RC of a vehicle when the vehicle serves a backhaul customer before serving all the linehaul customers. Letting Q^r be the maximum capacity of vehicle r and the RC on route r is denoted by $\mathrm{RC}_r(i, j)$. The formula to calculate the RC is given as

$$\mathrm{RC}_r(i, j) = \left(1 - \frac{L^r_{ij}}{Q^r}\right).100. \tag{4}$$

The main reasoning behind giving a new criterion is that the problem associated with the mixed VRPB is the difficulty of arranging goods on the vehicle. Therefore, it is appropriate that the restriction applied with respect to backhaul customers should be based on the RC of the vehicle rather than the actual load of the vehicle. If we consider the example given in [2] from a different point of view, let us assume not just one vehicle, as assumed in [2], now assume two vehicles of maximum capacity 100 units set off from the depot, one filled to capacity and the other carrying only 70 units. If the RP is set at 80% say, the first vehicle can accept a backhaul when there are still 20 units on the vehicle, whereas the second vehicle must wait till it has 14 units or less before accepting a backhaul. As it can be seen from this example in such cases using the RP criterion alone does not make sense if the difficulty in practice

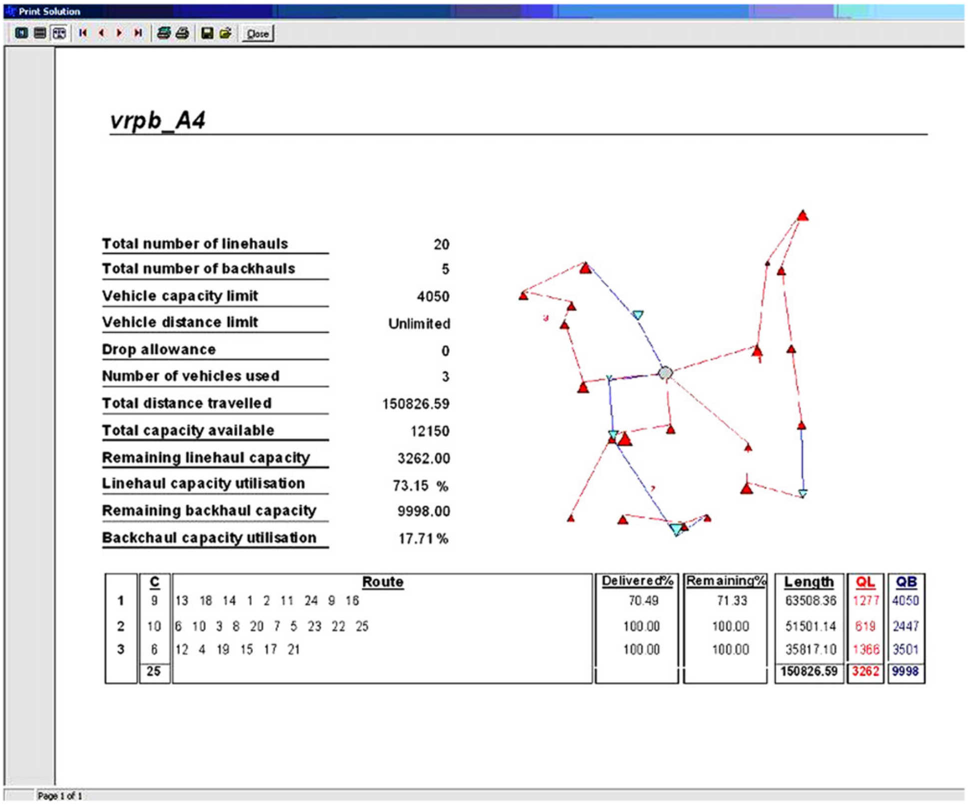

Fig. 3. Report of the solution of a restricted VRPB instance.

is re-arranging the vehicle contents. In summary, our new criterion plays an important role in making intelligent decisions and gives more sophisticated information about the problem to the decision maker in order to cope with the difficulties in practice. In CRUISE2 the user is informed by a popup window about the RP and RC just before the insertion of a backhaul customer in front of a linehaul customer as well as the insertion of a linehaul customer after a backhaul customer.

As seen in Fig. 2, the classical VRPB solution is achieved for a VRPB instance with 25 customers and three vehicles. The user is guided by the information on the popup window about the RP and RC just before the insertion of a customer, which has been identified by a black arrow in Fig. 2. The interface is flexible and sufficient for the user to control and restrict the mix of backhaul and linehaul customers to the required part of any route in order to obtain a practical and better solution.

The problem modification stage of the solution method ends with creation of a reconstructed VRPB that can be called mixed or restricted VRPB where the names depend on the specifications of the user. If the user constructs a problem in which at least one vehicle route is defined with a RP of 0%, this problem is a mixed VRPB.

3.3. Final solution stage

After reconstruction of the solution in restricted VRPB or mixed VRPB form, the user is allowed to improve the solution by a new call to solver to apply the GRAMPS algorithm as seen in Fig. 3. By allowing a backhaul customer in front of a linehaul customer, the user has changed the structure of the problem and now by calling the solver, the process is guided to search for a better solution using the GRAMPS algorithm.

After inserting a backhaul customer between two linehaul customers as seen in Fig. 2 the routing cost of the solution for the test problem is equal to 154 870 units. After running GRAMPS, the new routing cost was reduced to 150 826 units (see Fig. 3). It can be concluded that the GRAMPS algorithm generally finds better solutions with respect to the user's specifications and the technique described here is found to be very effective at improving solutions through the insights of the user.

Moreover, at the end of the decision process, the user has the opportunity to see the report of the solution, which includes information about RC and RP of the vehicles. An example of the final report of the solution is given in Fig. 3.

In this final report the last five columns include the basic information about the solution. The column 'Delivered%' gives the RP of the insertion of backhaul customers and the column 'Remaining%' gives the percentage of RC on the vehicle after the first backhaul customer. In column Length the distance travelled by the corresponding vehicle is given and in columns QL and QB, the remaining linehaul and backhaul capacity of the corresponding vehicle are given. For each benchmark problem the smallest values that are given in the column 'Delivered%' are used in the analysis of the results.

4. Computational results

The proposed procedure has been tested on two sets of benchmark problems. The first set consists of 14 benchmark problems selected from 62 benchmark problems given by Goetschalckx and Jacobs-Blecha [5], so far called Set 1. These 62 problems are divided into 14 groups and problems in each group are exactly same except their vehicle capacities and distance limits. We selected one problem from each group which includes the fewest vehicles. The selected problems include between 25 and 150 customers with backhauls ranging between 25% and 50%. The second set consists of nine benchmark problems selected from 30 benchmark problems given by Toth and Vigo [9], so far called Set 2. These instances include between 50 and 100 customers and each instance has linehaul percentage of 50%, 66% and 80%, respectively. In the literature two different methods of calculation are used in order to find Euclidean distances between pairs of customers for the problem instances. The first way is to have a real-valued matrix with Euclidean distances and calculate the final solution by rounding to the nearest integer. The second way is to have an integer-valued cost matrix with the Euclidean distances multiplied by 10 and rounded to the nearest integer and the final solution is calculating by dividing these values by 10 and rounding to the nearest integer. In this paper we used the first way for calculating distances. These Set 1 and Set 2 problem instances are constructed without any particular criteria except the ones described above.

First of all in order to show the simplicity and the usefulness of the proposed approach, the DSS was applied by 18 undergraduate students at Instituto Politecnico da Guarda to Set 1 and Set 2 problem instances for the classical VRPB problems. These students were not told that these were benchmark problems and asked to spend maximum 15 min on each problem instances. Tables 1 and 2 summarise the results obtained by 18 students. For Set 1, the best know solution of two problem instances achieved by at least one student and for Set 2 one student achieved a new best-known solution for problem instance eil51_80 (in bold). The best values referenced by Osman and Wassan [22] are included in Tables 1 and 2 for comparison.

Table 1
Solutions for Set 1 benchmark problems obtained by the 18 students

Name	No. of customers	No. of backhauls	No. of vehicles	Capacity	Best known	Students' Best	%Gap
vrpb_A4	20	5	3	4050	155 796	155 796	0.00
vrpb_B3	20	10	3	4000	169 372	169 372	0.00
vrpb_C4	20	20	4	4150	195 366	198 943	1.83
vrpb_D4	30	8	5	4075	205 832	212 628	3.30
vrpb_E3	30	15	4	5225	206 659	208 837	1.05
vrpb_F4	30	30	4	5500	233 861	235 175	0.56
vrpb_G6	45	12	4	8000	213 457	219 740	2.94
vrpb_H5	45	23	4	7100	246 121	246 604	0.20
vrpb_I4	45	45	6	5700	295 988	299 522	1.19
vrpb_J3	75	19	6	8200	279 219	282 668	1.24
vrpb_K4	75	38	7	6200	349 039	351 306	0.65
vrpb_L4	75	75	7	6000	384 844	394 424	2.49
vrpb_M4	100	25	7	8000	348 624	359 904	3.24
vrpb_N6	100	50	8	8500	377 665	388 451	2.86
Average							1.54
Standard deviation							1.18

Table 2
Solutions for Set 2 benchmark problems obtained by the 18 students

Name	No. of customers	No. of backhauls	No. of vehicles	Capacity	Best known	Students' Best	%Gap
eil51_50	50	25	3	160	559	564	0.89
eil51_66	50	16	4	160	548	560	2.19
eil51_80	50	10	4	160	565	**564**	−0.10
eilA76_50	75	38	6	140	739	755	2.15
eilA76_66	75	25	7	140	768	775	0.90
eilA76_80	75	15	8	140	781	815	4.35
eilA101_50	100	50	4	200	842	863	2.52
eilA101_66	100	33	6	200	846	878	3.84
eilA101_80	100	20	6	200	875	894	2.15
Average							2.10
Standard deviation							1.33

In the last column of both Tables 1 and 2 the percentages above best-known solutions of students' best results are given. In the last two rows of the tables the average percentage and the standard deviation of these percentages above best-known solutions are calculated. It should be noted that these students had the disadvantage of using the DSS with no experience of vehicle routing. The average percentages and the standard deviations show that the proposed approach within the DSS is able to produce high quality solutions even if inexperienced users apply them.

For testing the proposed approach for restricted and mixed VRPB, five different values of restricted percentage RP are considered, namely 100% (the classical VRPB), 90%, 80%, 70% and 0% (the fully mixed VRPB). These values are chosen in order to demonstrate the reduction of the route cost that can be achieved as the constraints on backhaul and linehaul customers are progressively relaxed.

Although there is only one published result for the efficiency of the proposed heuristic for solving the restricted VRPB in [2], there are no results given for the travelling costs so it is not suitable for comparison. In fact we compare the results obtained for the classical VRPB in the initial solution stage with the results obtained for the restricted VRPB by using CRUISE2's interactive tools and GRAMPS applied in the problem modification stage. On the other hand we gave the best solutions published in [22] to show the performance of our GRAMPS heuristic and the reported results for the classical VRPB are obtained from a single run of the GRAMPS algorithm following a single interactive session.

From Table 3 it can be seen that our decision system with the interactive GRAMPS heuristic is successful at producing solutions close to the best-known solutions given in the literature [22], moreover 4 out of 14 solutions for the classical VRPB instances are equal to the best published ones.

The results illustrate that an average improvement of 0.73–2.90% could be generated when RP is set between 90% and 70%, which can be practically implemented and these improvements can be achieved with very high RC values which means the solutions are practically applicable. Moreover, it is illustrated that a maximum improvement of 10.06% has been generated in the case of the fully mixed VRPB and although the RC value seems quite small in our DSSs environment the decision maker has the opportunity to analyse the solutions and judge whether they are reasonable for application.

Table 4 reports the results of Set 2 problem instances using real-valued cost matrices with rounding the final result to the nearest integer. In Table 2 it can be seen that our GRAMPS heuristic produced a new best solution for the benchmark problem that is called eil51_80 and very close solutions to the best-known solutions given in the literature. For Set 2 test problems our heuristic achieved more successful improvements than Set 1 test problems in the average. The results showed that an average improvement of 1.32–4.56% could be generated when RP is set between 90% and 70%, which can be practically implemented and these improvements can be achieved with very high RC values. Again that means the solutions produced using CRUISE2 DSS are practically applicable.

However, an interesting feature is that some problems in these two sets of problem instances have little or no potential for improvement, and the degree of relaxation required to achieve a significant improvement also varies. Therefore, it is not appropriate to focus on average improvements. The important point is

Table 3
The classical, restricted and mixed VRPB solutions for Set 1 benchmark problems

Name	No. of customers	No. of backhauls	No. of vehicles	Capacity	Best known	Our best	Cost			Cost			Cost			Cost		
							RP 90%	RC (%)	Improvement (%)	RP 80%	RC (%)	Improvement (%)	RP 70%	RC (%)	Improvement (%)	RP 0%	RC (%)	Improvement (%)
vrpb_A4	20	5	3	4050	155 796	155 796	154 523	91.33	0.82	153 896	80.10	1.22	150 826	80.10	3.19	142 874	33.73	8.29
vrpb_B3	20	10	3	4000	169 372	169 372	169 190	97.91	0.11	167 629	83.52	1.03	157 638	71.81	6.93	152 340	3.90	10.06
vrpb_C4	20	20	4	4150	195 366	195 366	189 432	91.42	3.04	189 432	91.47	3.04	185 480	82.36	5.06	186 901	64.49	4.33
vrpb_D4	30	8	5	4075	205 831	205 831	204 409	95.45	0.69	203 976	89.61	0.90	202 178	71.46	1.77	192 520	32.96	6.47
vrpb_E3	30	15	4	5225	206 659	208 836	208 703	97.46	0.06	207 924	94.99	0.44	204 920	81.11	1.88	203 188	81.11	2.70
vrpb_F4	30	30	4	5500	233 861	235 175	234 564	97.31	0.26	233 369	81.41	0.77	229 693	86.63	2.33	227 394	55.93	3.31
vrpb_G6	45	12	4	8000	213 457	219 740	218 468	94.65	0.58	215 676	89.07	1.85	214 831	83.73	2.23	205 222	65.82	6.61
vrpb_H5	45	23	4	7100	246 121	246 604	246 166	93.62	0.18	243 594	89.62	1.22	238 746	78.42	3.19	234 205	20.89	5.03
vrpb_I4	45	45	6	5700	295 999	299 522	297 837	93.28	0.56	295 110	83.60	1.47	292 454	83.60	2.36	284 203	27.14	5.11
vrpb_J3	75	19	6	8200	279 219	282 668	279 223	92.63	1.22	274 190	87.27	3.00	274 190	87.21	3.00	278 420	29.98	1.50
vrpb_K4	75	38	7	6200	349 038	351 306	347 948	92.32	0.96	341 265	82.71	2.86	341 265	82.74	2.86	340 133	54.24	3.18
vrpb_L4	75	75	7	6000	384 844	394 937	394 937	95.42	0.00	394 937	95.42	0.00	387 037	78.18	2.00	381 286	41.13	3.46
vrpb_M4	100	25	7	8000	348 624	359 904	356 135	81.30	1.05	355 931	81.30	1.10	350 848	75.27	2.52	347 503	26.19	3.45
vrpb_N6	100	50	8	8500	377 665	388 451	385 353	91.69	0.80	382 995	86.81	1.40	382 995	81.95	1.40	377 384	30.21	2.85
Average									0.74			1.45			2.91			4.74
Standard deviation									0.74			0.90			1.41			2.30

Table 4
The classical, restricted and mixed VRPB solutions for Set 2 benchmark problems

Name	No. of customers	No. of backhauls	No. of vehicles	Capacity	Best known	Our best	Cost RP 90%	RC (%)	Improvement (%)	Cost RP 80%	RC (%)	Improvement (%)	Cost RP 70%	RC (%)	Improvement (%)	Cost RP 0%	RC (%)	Improvement (%)
eil51_50	50	25	3	160	559	559	558	96.88	0.18	555	90.00	0.72	551	82.50	1.43	501	43.75	10.38
eil51_66	50	16	4	160	548	551	544	94.38	1.27	546	99.68	0.91	535	77.60	2.90	524	48.76	4.90
eil51_80	50	10	4	160	565	**564**	551	93.75	2.30	547	84.38	3.01	539	80.00	4.43	497	35.63	11.88
eilA76_50	75	38	6	140	739	751	746	93.57	0.67	731	89.29	2.66	729	77.90	2.93	720	51.43	4.13
eilA76_66	75	25	7	140	768	775	762	91.43	1.68	745	82.86	3.87	738	73.52	4.77	728	15.858	6.06
eilA76_80	75	15	8	140	781	815	811	90.00	0.49	778	80.00	4.54	777	72.40	4.66	759	45.00	6.87
eilA101_50	100	50	4	200	842	858	848	95.50	1.17	816	82.00	4.90	800	73.50	6.76	784	1.00	8.62
eilA101_66	100	33	6	200	846	860	841	95.00	2.21	829	85.50	3.60	812	81.00	5.58	778	41.00	9.53
eilA101_80	100	20	6	200	875	889	872	92.50	1.91	840	87.50	5.51	822	75.00	7.54	812	7.00	8.66
Average									1.32			3.30			4.56			7.89
Standard deviation									0.77			1.67			1.94			2.58

that the interactive approach allows us to identify those problems where a relaxation will give rise to significant improvement, and to explore the degree of relaxation required to achieve this improvement. Users can then exercise their judgement and experience to decide whether any particular solution will be acceptable in practice.

The other point that we would like to highlight about the performance of the proposed DSS is that the interactive session times were short. All the experiments were performed on a 2.53 GHz Pentium IV PC, running Windows XP. The average solution time with a reasonable amount of interactive session that a user with an average knowledge of the problem structure has to spend for producing a practically applicable solution varies between 10 and 15 min depending on the complexity of the problem. It is very difficult to compare our proposed approach regarding the computational time, as it is not only the speed of the algorithm that indicates the performance. The knowledge of the user, the time spent by the user and the size of problem also play an important role. In our experiments the average computational time of GRAMPS algorithm that solves the classical VRPB problem in the initial solution stage of the visual interactive approach was about 40 s for both Set 1 and Set 2 problem instances. The portion of time that was used by running the algorithm within the visual interactive approach (in the initial solution stage and in the final solution stage) averaged 1 min. The benchmark results given by Dongarra [23] for the Mflop/s are used to help compare the computational requirements of our algorithm and the ones in the literature. The heuristic algorithm that was proposed by Osman and Wassan [22] has average 2267.25 CPU seconds for the best solutions of Set 1 problem instances and the CPU times for their heuristics were on Sun-Sparc Server 1000 with 50 MHz processor (10 Mflop/s). The CPU times for our GRAMPS algorithm are on a Pentium IV PC with 2.53 GHz (1190 Mflop/s). At a simplistic level, we might divide the average CPU for Osman and Wassan by 119, giving 19 s, to allow for the difference in performance of the computers, but it should be noted that Osman and Wassan used 8 runs in each of two versions of their heuristic to obtain their best solutions, compared to a single interactive session to obtain our results. These results indicate that the proposed GRAMPS algorithm and the visual interactive approach are competitive with the other exact and approximate algorithms in the literature [22] regarding the solution process time and the quality of solutions, whilst offering significant improvements in flexibility for practical implementations.

5. Conclusion

In this study the restricted vehicle routing problems with backhauls where the backhauls are not necessarily visited after all linehaul customers are serviced and the mixed vehicle routing problems with backhauls where backhaul and linehaul customers are fully mixed are described. A visual interactive DSS that uses a GRAMPS procedure is proposed to solve these problems. A flexible interface is developed for the user to control the restriction of backhaul and linehaul customers to the required parts of the routes in order to obtain a practical and better solution. The proposed DSS gives users the opportunity to incorporate their knowledge, understanding and experience of particular situations into the solution process. It allows users to investigate different solutions by observing the solution obtained and making changes. It is obvious that human intelligence is better than a computer system in evaluating vague situations. With the proposed interactive DSS the heuristics that are used to solve the VRPB problems are under the control of the user and the user is able to guide the search towards more promising parts of the problem's solution space by selecting the parameters and algorithms involved in the solution process. Furthermore, solutions that are obtained using the CRUISE2 DSS are more acceptable for a decision maker, because the user, being more involved in the solution process, can interact with an algorithmic procedure and consequently affect the final solutions to reflect his/her subjective point of view.

The GRAMPS algorithm within the proposed visual interactive approach was found to be very competitive regarding the computational time and the solution quality with the best algorithms in the literature and the overall visual interactive approach was able to find a new solution for one of the instances and several practical solutions for both Set 1 and Set 2 problem instances. The computational results on data Set 1 and Set 2 indicated that our visual interactive approach is capable of producing high quality solutions for the classical, mixed and restricted vehicle routing problem with backhauls within a reasonable computation time. The new criterion, percentage of RC, is found to be effective along side the restricted percentage criterion towards finding a compromise between the mixed or restricted and the classical VRPB problems. Hence, the new visual interactive approach and the new criterion can be recommended for real-life vehicle routing problems with backhauls. Moreover, when the results obtained by students who are not "experts" were compared to the results obtained by us it can be conclude that good results for the VRPB applications may likely be achieved by an average truck dispatcher using the proposed DSS.

Possible research directions include the study of implementing the circuity factors proposed by Ballou et al. [24] in order to approximate actual travel distances for solving real-life inter-city vehicle routing problems, and modifying the proposed DSS in order to solve more realistic variants of VRPB such as VRPB with heterogeneous vehicles. Although, the proposed DSS was used by 18 undergraduate students in order to simulate the advantages of the DSS utilisation another possible research direction includes performing the similar experiment in a real dispatching company with real truck dispatchers.

Acknowledgements

The authors would like to thank the anonymous referees for their comments that helped to improve the content of the paper. The authors must also thank the 18 undergraduates at Instituto Politecnico da Guarda who, unknowingly, provided some results for this paper.

References

[1] Casco D, Golden BL, Wasil E. Vehicle routing with backhauls: models algorithms and case studies. In: Golden BL, Assad AA, editors. Vehicle routing: methods and studies. 1988. p. 127–47.
[2] Wade A, Salhi S. An investigation into a new class of vehicle routing problem with backhauls. Omega 2002;30:479–87.
[3] Deif I, Bodin L. Extension of the Clarke and Wright algorithm for solving the vehicle routing problem with backhauling. In: Kidder A, editor. Proceedings of the Babson conference on software uses in transportation and logistic management, Babson Park, 1984. p. 75–96.
[4] Clarke G, Wright J. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 1964;12:568–81.
[5] Goetschalckx M, Jacobs-Blecha C. The vehicle routing problem with backhauls. European Journal of Operational Research 1989;42:39–51.
[6] Goetschalckx M, Jacobs-Blecha C. The vehicle routing problem with backhauls: properties and solution algorithms. Technical report, Georgia Institute of Technology, School of Industrial and Systems Engineering, 1993.
[7] Fisher ML, Jaikumar R. A generalized assignment heuristic for vehicle routing. Networks 1981;11:109–24.
[8] Toth P, Vigo D. A heuristic algorithm for the symmetric and asymmetric vehicle routing problem with backhauls. European Journal of Operational Research 1999;113:528–43.
[9] Toth P, Vigo D. An exact algorithm for the vehicle routing problem with backhauls. Transportation Science 1997;31(4): 372–85.
[10] Toth P, Vigo D. The vehicle routing problem. SIAM monographs on discrete mathematics and applications. Philadelphia, PA: SIAM; 2002 [Chapter 8].

[11] Salhi S, Nagy G. A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. Journal of the Operational Research Society 1999;50:1034–42.
[12] Krolak P, Felts W, Nelson J. A man–machine approach toward solving the generalized truck-dispatching problem. Transportation Science 197; 6(2):149–70.
[13] Cullen F, Jarvis J, Ratliff H. Set portioning based heuristic for interactive routing. Networks 1981;11:125–43.
[14] Carreto AC, Baker BM. A GRASP interactive approach to the vehicle routing problem with backhauls. In: Ribeiro CC, Hansen P, editors. Essays and surveys on metaheuristics. Dordrecht: Kluwer Academic Publishers; 2002. p. 185–200.
[15] Baker BM. Further improvements to vehicle routing heuristics. Journal of the Operational Research Society 1992;43:1009–12.
[16] Baker BM, Carreto AC. A visual interactive approach to vehicle routing. Computers and Operations Research 2003;30:321–37.
[17] Gillett BE, Miller LR. A heuristic algorithm for the vehicle dispatch problem. Operations Research 1974;22:240–349.
[18] Ahmadi S, Osman IH. Greedy random adaptive memory programming search for the capacitated clustering problem clustering problem. European Journal of Operational Research 2005;162(1):30–44.
[19] Fleurent C, Glover F. Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory. INFORMS Journal on Computing 1999;11:198–204.
[20] Osman IH. Metastrategy simulated annealing and Tsabu search algorithms for the vehicle routing problem. Annals of Operations Research 1993;41:421–51.
[21] Yellow P. A computational modification to the savings method of vehicle scheduling. Operational Research Quarterly 1970;21:281–3.
[22] Osman IH, Wassan NA. A reactive Tabu search metaheuristic for the vehicle routing problem with backhauls. Journal of Scheduling 2002;5(4):263–85.
[23] Dongarra JJ. Performance of various computers using standard linear equations software. Technical report, University of Tennessee and Oak Ridge National Laboratory, 2005. ⟨http://www.netlib.org/benchmark/performance.ps⟩.
[24] Ballou R, Rahardja H, Sakai N. Selected country circuity factors for road travel distance estimation. Transportation Research Part A 2002;36:843–8.