The Multiple Trip Vehicle Routing Problem with Backhauls: Formulation and a Two-Level Variable Neighbourhood Search

Naveed Wassan, N.A. Wassan, G. Nagy and S. Salhi
Centre for Logistics & Heuristics Optimisation, Kent Business School, University of Kent, Canterbury, UK

Abstract:

In this paper a new VRP variant the Multiple Trip Vehicle Routing Problem with Backhauls (MT-VRPB) is investigated. The classical MT-VRP model is extended by including the backhauling aspect. An ILP formulation of the MT-VRPB is first presented and CPLEX results for small and medium size instances are reported. For large instances of the MT-VRPB a *Two-Level VNS* algorithm is developed. To gain a continuous balanced intensification and diversification during the search process VNS is embedded with the sequential VND and a multi-layer local search approach. The algorithm is tested on a set of new MT-VRPB data instances which we generated. Interesting computational results are presented. The *Two-Level VNS* produced excellent results when tested on the special variant of the VRPB.

Keywords: Routing, Multiple trips, Backhauling, VNS, Meta-heuristics

1. Introduction

We introduce a new vehicle routing problem (VRP) variant called the Multiple Trip Vehicle Routing Problem with Backhauls (MT-VRPB). The MT-VRPB combines the characteristics of the classical versions of two VRP problems studied in the literature, i.e., the MT-VRP in which a vehicle may perform several routes (trips) within a given time period; and the vehicle routing problem with backhauls (VRPB) in which a vehicle may pick up goods to bring back to the depot once the deliveries are made. Therefore in the MT-VRPB a vehicle may not only perform more than one trip in a given planning period but it can also collect goods in each trip. Since the MT-VRP and the VRPB have been studied independently in the literature, we first provide a brief description of these two routing problems.

MT-VRP: The MT-VRP model is an extension of the classical VRP in which a vehicle may perform several routes (trips) within a given time period. Along with the typical VRP constraints an additional aspect is included in the model which involves the assignment of the optimised set of routes to the available fleet (Taillard et al., 1996).

VRPB: The VRPB is also an extension to the classical VRP that involves two types of customers, deliveries (linehauls) and pickups (backhauls). Typical additional constraints include: (i) each vehicle must perform all the deliveries before making any pickups; (ii) routes with only backhauls are disallowed, but routes with only linehauls can be performed (Goetschalckx and Jacobs-Blecha, 1989).

Both the MT-VRP and the VRPB are considered to be more valuable than the classical VRP in terms of cost savings and placing fewer numbers of vehicles on the roads. These features are very important from both the managerial and the ecological perspectives. By combining the aspects of the above two models into a new model, the MT-VRPB, we achieve a more realistic model. To our knowledge, this is the first time this variant is being studied in the literature. However, there is one study that deals with time windows MT-VRPB-TW by Ong and Suprayogi (2011) where an ant colony optimization algorithm is implemented. Below we present a detailed description of our MT-VRPB model.

MT-VRPB: The MT-VRPB can be described as a VRP problem with the additional possibilities of having vehicles involved in backhauling and multiple trips in a single planning period. The objective is to minimise the total cost by reducing the total distance travelled and the number of vehicles used.

Problem characteristics:

- A given set of customers is divided into two subsets, i.e., delivery (linehaul) and pickup (backhaul).
- A homogenous fleet of vehicles.
- A vehicle may perform more than one trip in a single planning period.
- All delivery customers are served before any pickup ones.
- Vehicles are not allowed to service only backhauls on any route; however linehaul only routes are allowed.

- Vehicle capacity constraints are imposed.
- Note - The *route length* constraint is not imposed in this study, however the model is flexible to add this constraint if needed.

Figure 1 presents a graphical example of the proposed MT-VRPB with three homogeneous types of vehicles and a planning period T; *Vehicle 1* performs two trips whereas vehicles 2 & 3 perform one trip each.

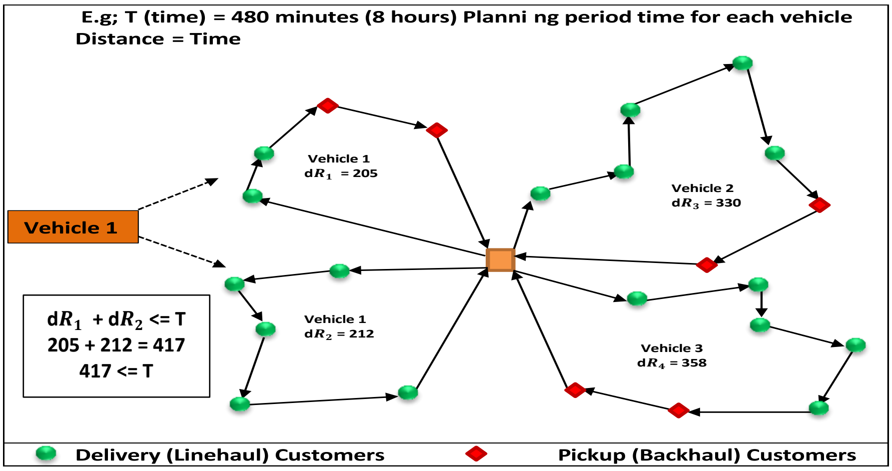

Figure 1: An example of the MT-VRPB.

The rest of the paper is structured as follows. Section 2 presents the literature review followed by a formulation of the MT-VRPB in Section 3. Section 4 explains the proposed algorithm. The computational results, including the generation of the newly created MT-VRPB data set, are presented in Section 5. Finally, a summary of the conclusions is provided in Section 6.

2. Literature review

Since there is no literature available on the MT-VRPB, we provide brief reviews for the two related routing problems namely the MT-VRP and the VRPB.

2.1 MT-VRP

The multi-trip vehicle routing was first studied in Salhi (1987) where multiple trips were conducted in the context of vehicle fleet mix. Limited to double trips, a matching algorithm is proposed to assign routes to vehicles within a refinement process. Taillard *et al*. (1996) introduced the MT-VRP model based on the classical VRP and proposed a three-phase heuristic algorithm. In the first phase, tabu search is used to generate a population of routes satisfying the capacity constraint; a set of different VRP solutions is then obtained in phase two. Routes are then assigned to the vehicles by solving the bin-packing problem (BPP) in the last phase. Moreover, a set of classical MT-VRP instances are generated in their study which are widely used in the literature as benchmarks. Brandao and Mercer (1997) studied a real world application of MT-VRP with time windows and heterogeneous fleet, and used a tabu search algorithm to solve the problem. The methodology developed in this study is adapted in Brandao and Mercer (1998) where classical MT-VRP instances were solved and compared. Petch and Salhi (2004) developed a multi-phase constructive heuristic algorithm with an objective of minimizing the overtime used in multi-trips. The algorithm obtained an MT-VRP solution by solving BPP which is improved further using the 2-Opt and 3-Opt exchange heuristic procedures. Salhi and Petch (2007) revisited their previous study described above by using a genetic algorithm which proved to be faster. Olivera and Viera (2007) studied this problem and proposed an adaptive memory programming (AMP) approach with tabu search. A set of elite routes is selected randomly from the memory and packed into vehicles solving the BPP while applying some local search refinements based on reducing the driver overtime. The AMP algorithm found feasible packing of bins (without overtime) for most of the classical benchmark instances as compared to the previous studies. Alonso, Alvarez, and Beasley (2008) studied a variant of multi-trip called site-dependent periodic MT-VRP using a tabu search algorithm. In this situation, given a planning horizon of t days, each customer gets served up to t times. Macedo et al. (2011) introduced the time windows aspect into this problem and solved the resulting model to optimality. Mingozzi, Roberti, and Toth (2013) developed an exact method based on two set-partitioning formulations to tackle the MT-VRP. A subset of 52 instances, ranging in size from 50 to 120 customers is tested and 42 are solved to optimality. For the rest, upper bounds are provided. Azi et al. (2014) recently proposed an adaptive large neighbourhood search algorithm that makes use of the ruin-and-recreate principle for the MT-VRP with the presence of service time at each node. Cattaruzza et al. (2014a) proposed a hybrid genetic algorithm for the MT-VRP that uses some adaptations from the literature. A new local search operator called the combined local search (CLS) is introduced that combines the standard VRP moves and performs the reassignments of trips to vehicles by using a swapping procedure leading to good quality results. Cattaruzza et al. (2014b) then extended the previous model to include time windows using an iterated local search methodology to solve the problem.

It is worth noting that the early studies on the MT-VRP concentrated mostly on the modelling side of the problem and the later ones on the design of powerful methods. By extending the MT-VRP model we aim to break this gap in the literature and open a new research avenue.

Finally, we note that the MT-VRP may form part of more complex logistics problems. Of particular note is the location-routing-scheduling problem, also known as the location-routing problem with multiple trips. This was introduced by Lin et al (2002), and solved using simulated annealing. Lin and Kwok (2006) extended this model to cater for multiple objectives. Recently, Macedo et al. (2015) developed a variable neighbourhood search algorithm for this problem.

2.2 VRPB

The VRPB has also attracted a good attention in the literature. Among exact approaches, Yano *et al*. (1987) developed a branch-and-bound framework based on the set covering approach for trucks in a retail chain industry. Toth and Vigo (1997) proposed a consolidated framework with both symmetric and asymmetric cost matrices. Their branch-and-bound algorithm obtains Lagrangian lower bound strengthened by adding valid inequalities in a cutting-plane fashion embedded in an integer linear programming model. Mingozzi *et al*. (1999) proposed a new set-partitioning based (0-1) integer programming model. This algorithm obtains a lower bound by blending various heuristic methods for solving the LP-relaxation of the dual problem.

The heuristics literature on the VRPB started in the early 80s but it was formally tackled by Goetschalckx and Jacobs-Blecha (1989) who developed a two-phase heuristic approach to solve a series of test instances which they generated. In their two-phase method, a space-filling approach is first used to generate an initial solution for the linehaul and the backhaul customers. The solutions are then merged in the second phase to obtain a combined LH-BH solution. Jacobs-Blecha and Goetschalckx (1993) developed a generalized assignment heuristic and produced a mathematical formulation of the problem. Toth and Vigo (1999) put forward a "cluster-first and route-second" algorithm for the VRPB. This algorithm exploits the information associated with the lower bound acquired from a Lagrangian relaxation using a new clustering method. The authors also introduced a VRPB data set based on the original VRP instances which is now commonly used for benchmarking.

The meta-heuristics are considered to be more robust methodologies to solve the VRPs. The first meta-heuristic approach to solve the VRPB was developed by Osman and Wassan (2002) who used a reactive tabu search for the VRPB. Brandao (2006) produced a multi-phase tabu search algorithm whereas Ropke and Pisinger (2006) presented a unified approach based on the concept of the large neighbourhood search for the VRPB. Further, Wassan (2007) developed a hybrid model in which reactive tabu search is blended with adaptive memory programming. Gajpal and Abad (2009) proposed a multi ant colony system in which two types of ants are exercised whereas Zachariadis and Kiranoudis (2012) used a local search heuristic that explores rich solution neighbourhoods and makes use of local search moves stored in Fibonacci Heaps. Recently, Cuervo et al. (2014) introduced an iterated local search algorithm in which an oscillating local search heuristic is used. The above methodologies have their pros and cons but appear to produce high quality results. For recent developments on the VRPB the reader may refer to Salhi *et al*. (2014).

3. MT-VRPB Formulation

The MT-VRP is modelled as an integer linear program. The following formulation is similar to the two-indexed commodity flow formulation of Nagy, Wassan and Salhi (2013). However, the MT-VRPB formulation is a three-index commodity flow formulation. In three-index formulations, variables x_{ijk} specify whether arc (i, j) is traversed by a particular vehicle k or not.

The following notations are used throughout:

Sets

{0}	the depot (single depot)
L	the set of linehaul customers
B	the set of backhaul customers
K	the set of vehicles (bins)

Input Variables

d_{ij} the distance between locations i and j ($i \in \{0\} \cup L \cup B, j \in \{0\} \cup L \cup B$)

q_i the demand of customer i (such that $i \in L$ for a delivery demand and $i \in B$ for a pickup demand)

C vehicle capacity

T Planning period (maximum driving time)

Decision Variables

$$x_{ijk} = \begin{cases} 1, & \text{if vehicle } k \text{ travels from location } i \text{ directly to location } j; \\ 0, & \text{otherwise} \end{cases}$$

R_{ij} is the amount of delivery or pickup on board on arc ij

Minimise $$Z = \sum_{i\in\{0\}\cup L\cup B} \sum_{j\in\{0\}\cup L\cup B} \sum_{k\in K} d_{ij}\, x_{ijk} \quad (1)$$

Subject to $$\sum_{j\in\{0\}\cup L\cup B} \sum_{k\in K} x_{jik} = 1 \qquad i \in L \cup B \quad (2)$$

$$\sum_{j\in\{0\}\cup L\cup B} \sum_{k\in K} x_{ijk} = 1 \qquad i \in L \cup B \quad (3)$$

$$\sum_{j\in\{0\}\cup L\cup B} x_{jik} = \sum_{j\in\{0\}\cup L\cup B} x_{ijk} \qquad i \in L \cup B,\ \forall\, k \in K \quad (4)$$

$$\sum_{i\in\{0\}\cup L} R_{ij} - q_j = \sum_{i\in\{0\}\cup L\cup B} R_{ji} \qquad j \in L \quad (5)$$

$$\sum_{i\in L\cup B} R_{ij} + q_j = \sum_{i\in\{0\}\cup B} R_{ji} \qquad j \in B \quad (6)$$

$$R_{ij} \le C \sum_{k\in K} x_{ijk} \qquad i \in L \cup B, j \in L \cup B;\ \forall\, k \in K \quad (7)$$

$$\sum_{i\in\{0\}\cup L\cup B} \sum_{j\in\{0\}\cup L\cup B} d_{ij}\, x_{ijk} \le T \qquad \forall\ k \in K \quad (8)$$

$$R_{ij} = 0 \qquad i \in L,\ j \in B \cup \{0\} \quad (9)$$

$$x_{ijk} = 0 \qquad i \in B,\ j \in L, k \in K \quad (10)$$

$$x_{0jk} = 0 \qquad j \in B, k \in K \quad (11)$$

$$R_{ij} \ge 0 \qquad i \in \{0\} \cup L \cup B,\ j \in L \cup B \quad (12)$$

$$x_{ijk} \in \{0,1\} \qquad i \in \{0\} \cup L \cup B,\ j \in \{0\} \cup L \cup B,\ k \in K \quad (13)$$

Equation (1) illustrates the objective function representing the total distance travelled. Constraints (2) and (3) ensure that every customer is served exactly once (every customer has an incoming arc and every customer has an outgoing arc). Constraint (4) states that the number of times vehicle k enters into customer i is the same as the number of times it leaves customer i. The vehicle load variation on a route is ensured by Constraints (5) and (6) for linehaul and backhaul customers, respectively. Inequalities (7) and (8) impose the maximum vehicle capacity constraint and the maximum working period constraints in which a vehicle is allowed to serve the routes, respectively. Constraints (9) forbid any load carried from a linehaul customer to either a backhaul customer or to the depot. Constraints (10) and (11) impose a restriction that a vehicle cannot travel from a backhaul to a linehaul customer and it cannot travel directly from the depot to a backhaul customer, respectively "(One may debate whether these constraints are really required in practice; we chose to include them to be in line with the subject literature). Inequality (12) sets R_{ij} as a non-negative variable. Finally, (13) refer to the binary decision variable x_{ijk}.

The above formulation may be modified as the MT-VRP by simply setting the number of backhaul customers equal to zero using equation (14).

$$\boldsymbol{B} = \emptyset \tag{14}$$

Moreover, the formulation can be extended to cater for the conditions where the number of available vehicles is no more than (or equals to), a given number K. This can be achieved by adding the following constraints (15) in the model.

$$\sum_{j \in L \cup B} x_{ijk} \leq K \qquad i \in \{0\}; \quad \forall\, (i \in L \cup B) \tag{15}$$

The MT-VRPB formulation can also be reduced to the VRPB (classical vehicle routing problem with backhauls) by adding the following constraint (16) in the model.

$$\sum_{j \in L \cup B} x_{ijk} \leq 1 \qquad i \in \{0\}; \quad \forall\, (k \in K) \tag{16}$$

Constraints (16) impose restrictions on every vehicle to be used once and therefore block the use of multiple-trips of vehicles.

4. *Two-Level VNS* Methodology

The steps of our *Two-Level VNS* methodology are presented as follows.

4.1 Initial solution

The Sweep method of Gillett and Miller (1974) is considered to be an efficient construction method for the VRPs. We have adapted a *sweep-first-assignment-second* based approach to generate an MT-VRPB initial solution. Initially two sets of open-ended routes are constructed by sweeping through LH and BH nodes separately. A distance/cost matrix for the assignment problem is created by including the distances between the end nodes of the open-ended routes. A dummy route containing the depot is also added to the matrix where a number of LH and BH routes are not equal. To produce combined LH-BH routes, the optimal matching is then obtained by solving an assignment problem using ILOG CPLEX 12.5 optimiser coded with C++ within Microsoft Visual Studio Environment.

An Illustrative example

An illustrative example of the problem instance *eil21_50* is shown in Figure 2. This instance has 21 customers consisting of 11 linehauls and 10 backhauls. A matrix containing the actual distances is shown in Figure 3. The optimal assignment matching result for the example problem is illustrated in Figure 4.

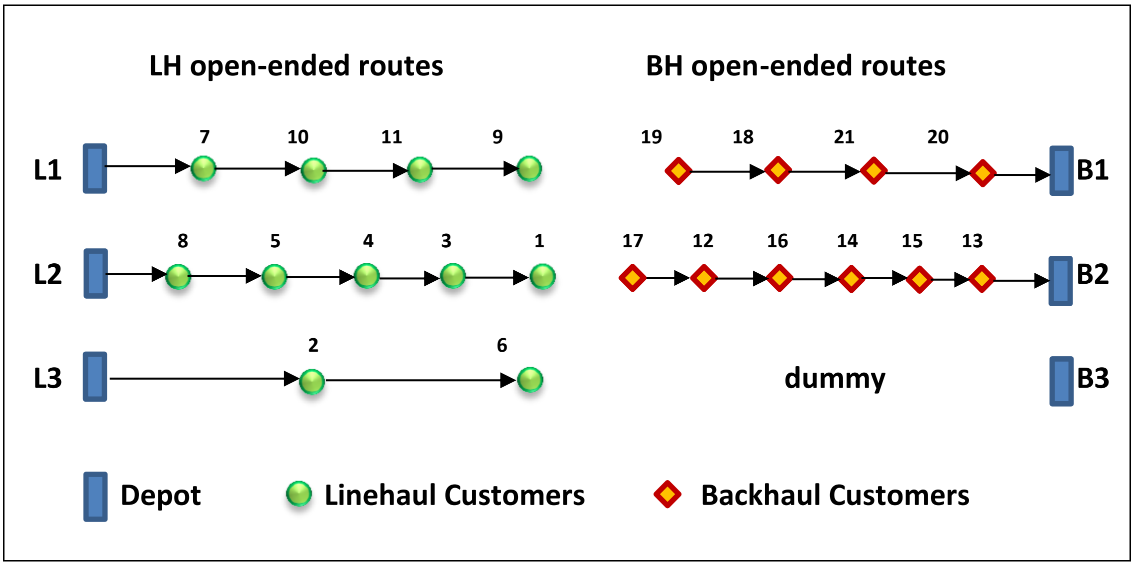

Figure 2: LH and BH open-ended routes (Problem instance *eil22_50* of *data set-2*)

	B1	B2	B3
L1	17	69	22
L2	72	9	49
L3	70	30	42

Figure 3: Distance matrix of end nodes

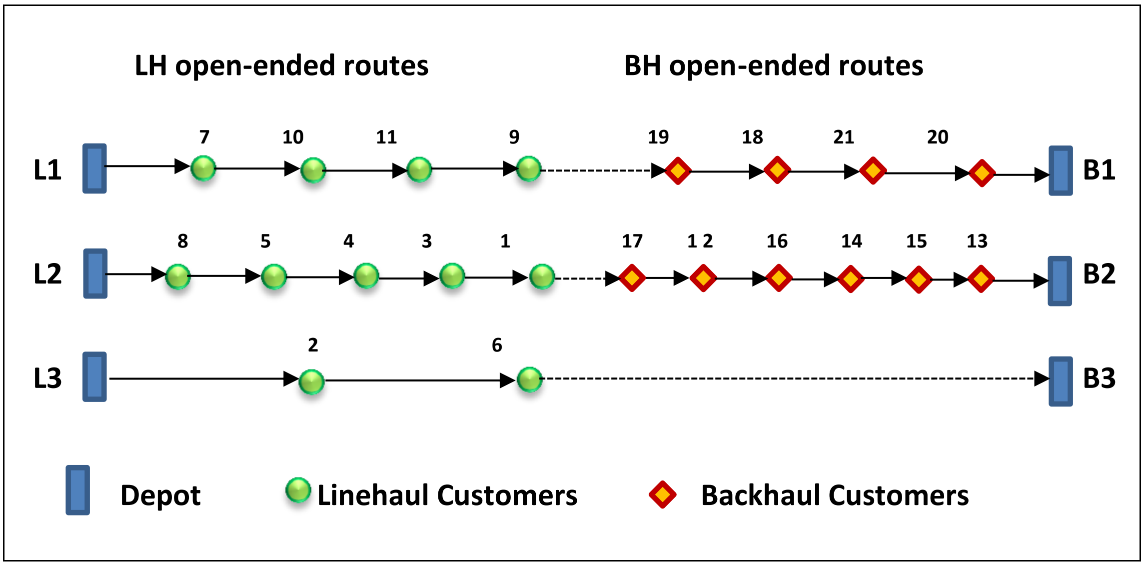

Figure 4: Combined LH+BH routes (problem instance no: eil22_50)

4.2 Two-Level VNS

The Variable Neighbourhood Search (VNS) approach (Mladenovic and Hansen, 1997) is based on the idea of a systematic change of neighbourhoods within a local search method. The concept of VNS is simple but has proved elegant and powerful in solving a variety of Combinatorial Optimization problems. Our *Two-Level VNS* is motivated by the enhanced features used in the recent paper on VNS by Mladenovic, Todosijevic and Urosevic (2014). The details of our VNS implementation are as follows.

The basic VNS concept is enriched by embedding a Sequential VND along with two shaking steps and a set of neighbourhood schemes to achieve a vigorous diversification during the search process. Moreover, a series of local search routines at two levels of the skeleton of the VNS are used to intensify the search. The merit of the two-level strategy is that it ensures a speedy and continuous balanced intensification and diversification by employing two shaking steps. The Pseudo code is presented in Figure 5.

4.2.1 An overview of the algorithm

The algorithm comprises of two levels, i.e., outer and inner. We have employed several neighbourhood structures along with associated local search refinements routines at both levels of the algorithm. For the outer-level we define N_k^O $(k = 1, \dots, k_{max})$ as a subset of neighbourhoods and LS_k^O $(k = 1, \dots, k_{max})$ as a subset of local search refinement routines; and at the inner-level N_l^I $(l = 1, \dots, l_{max})$ as a full set of neighbourhoods and LS_l^I $(l = 1, \dots, l_{max})$ as a full set of local search refinement routines. The neighbourhoods and the local search refinement routines are explained in subsections 4.2.2 and 4.2.3, respectively. Note that, the superscripts "*O*" and "*I*" refer to the neighbourhoods and local search refinement routines used at the outer and the inner levels, respectively. Moreover, a 3-dimentional data structure S_p is used to store the initial solution x as well as many other improved solutions during the search process.

At each cycle of the search process, the outer level of the algorithm generates randomly a transitory solution x' from $N_k^O(x)$. A subset LS_k^O of local search routines is utilised to improve the x'. The resulting best solution x'_{best} is then recorded and transferred to the inner level of the algorithm where a Sequential Variable Neighbourhood Descent (SeqVND) is used. At the inner level full sets of the neighbourhoods and local search refinement routines are utilised and embedded systematically within a multi-layer local search optimiser framework.

Again a transitory solution x'' is generated randomly from $N_l^I(x'_{best})$ at the inner-level transferred to LS_l^I (the multi-layer local search optimiser framework) for improvement. If the solution obtained by the multi-layer local search approach, x''_{best}, is better than the incumbent best solution x'_{best}, then it is updated as $x'_{best} = x''_{best}$ and the process cycles back to the same neighbourhood N_l^I. Moreover, if x''_{best} is found to be the same or worse compared to x'_{best}, then a new x'' is generated using the next neighbourhood $N_{l+1}^I(x'_{best})$ and the multi-level optimiser is then applied in the same manner. The process continues with the inner-level till $N^I_{l_{max}}$ is reached. At this stage, the search process shifts back to the outer-level.

```
Function Two-Level VNS (x, N^O_{k_max}, N^I_{l_max}, iter_max)
  Let: S_p = be a solution pool data structure
  S_p ← x
  iter ← 1
  while iter ≤ iter_max do
      ***start outer level***
      Let: LS^O_k = < R_3, R_4, R_5 >          [Subset of local search routines]
      Let: N^O_k = < N_4, N_5, N_6 >           [Subset of neighbourhood structures]
      k ← 1
      while k ≤ k_max do
          Select x' ∈ N^O_k(x) at random;              [shake outer level]
          x'_best ← LS^O_k(x');
                      ***start inner level***
              Let: LS^I_l = < {R_1& R_6}, {R_2& R_6}, {R_3& R_6}, {R_4& R_6}, {R_5& R_6} >
              Let: N^I_k = < N_1, ..., N_6 >                 [Full set of neighbourhood structures]
              l ← 1
              while l ≤ l_max do
                  Select x'' ∈ N^I_l(x'_best) at random;         [shake inner level]
                  x''_best ← LS^I_l (x'');                       [Multi-Layer local search framework]
                  If f(x''_best) < f(x'_best) then
                  x'_best ← x''_best; l ← 1;
                  Else l ← l + 1;
              end while
              return x'_best;
                      ***end inner level***
          If f(x'_best) < f(x) then
          x ← x'_best; S_p ← x; k ← 1;
          Else k ← k + 1;
      end while
      return x;
          ***end outer level***
  end while
```

Figure 5: Pseudo code for the *Two-Level VNS*

If x'_{best} is found to be better than the incumbent x then it is updated as $x = x'_{best}$ and the improved solution is stored $S_p = x$; hence, the process of generating a transitional solution restarts from the same neighbourhood N_k^O. But if x'_{best} is found to be the same or worse than the incumbent x, a new transitory x' is generated using the next neighbourhood in $N_{k+1}^O(x)$. Hence, the outer-level is also iterated till $N_{k_{max}}^I$ is reached. The process terminates when the maximum number of iterations $iter_{max}$ is met.

The Bin Packing Problem (BPP) is then solved for a pool of solutions stored in S_p obtained by the *Two-Level VNS* using CPLEX optimiser. Note that in the cases where a solution could not be packed due to the tight bin capacity (which equates to "maximum driving time") we use the *Bisection Method* (Petch and Salhi, 2004) to increase the bin capacity (i.e., allowing overtime) and the packed solution is reported with the corresponding overtime.

4.2.2 Neighbourhoods

The neighbourhood generation is a fundamental part in heuristic search design in general and in the VRPs in particular. Six neighbourhood schemes (*N_1,…, N_6*) are used in this study. These are briefly described as follows. *1-insertion intra-route (N_1)* relocates a customer at a non-adjacent arc within the same route; *1-insertion inter-route (N_2)* relocates a customer from one route to another; *1-1 swap (N_3)* exchanges two customers each taken from two separate routes; *2-0 shift (N_4)* relocates two consecutive customers form one route to another; *2-2 swap (N_5)* exchanges two pairs of consecutive customers taken from two separate routes; *2-1 swap (N_6)* exchanges a consecutive pair of customers from one route with a single customer from another route.

The moves in all the neighbourhood schemes are conducted according to backhauling constraints conventions described in Section 1.

4.2.3 Multi-Layer local search optimiser framework

The multi-layer local search optimiser is a combination of local search refinement routines that are employed within a local search framework as described in Subsection 4.2.1. The notion of manipulating the power of several neighbourhood structures as local searches within a local search framework was originally developed by Salhi and Sari (1997) and recently been implemented in Imran, Salhi and Wassan (2009) successfully. We have adapted this idea for our *Two-Level VNS* algorithm and used six neighbourhoods of Subsection 4.2.2 as local search refinement routines (R_1,… , R_6). The order of the local search routines in the multi-layer framework shown in Figure 6 was found empirically.

The multi-layer framework search process starts with a transitory solution x'' as explained in Subsection 4.2.1. Each local search routine is then executed in the order given in Figure 6 till a local optimum is reached whereas the post-optimiser routine *1-insertion intra-route* is then activated.

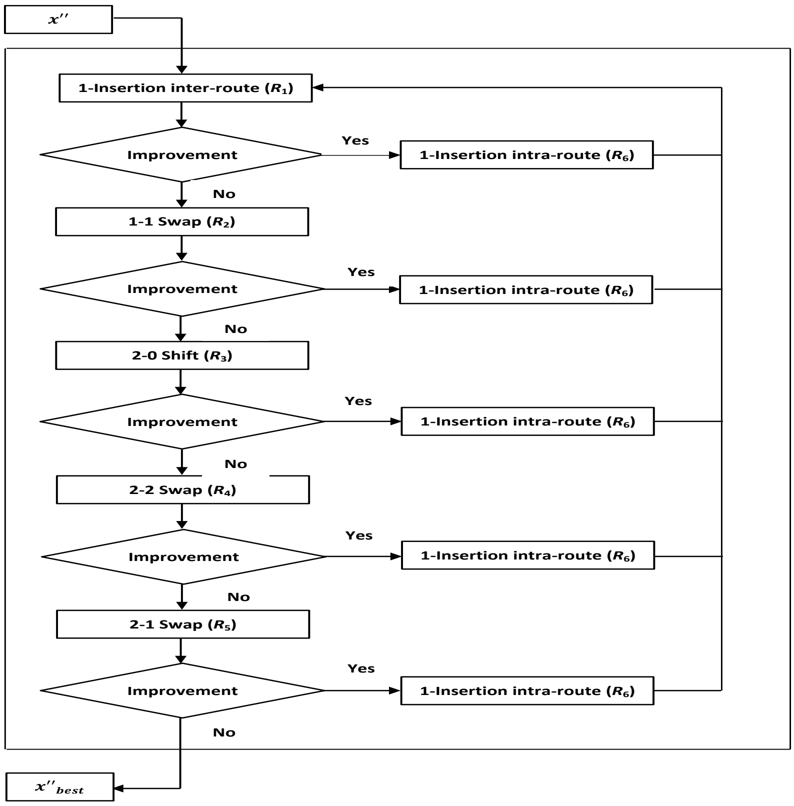

Figure 6: The multi-layer local search optimiser framework flow chart

5 Computational experience

The *Two-Level VNS* algorithm and the initial solution generation procedures are implemented in C++ programming within the Microsoft Visual Studio Environment. The experiments were executed on a PC with Intel(R) Core(TM) i7-2600 processor, CPU speed 3.40 GHz. The IBM ILOG CPLEX 12.5 is used to check the validity of our MT-VRPB formulation.

Initial Solution: The *sweep-first-assignment-second* approach is implemented, in which assignment part is solved by calling CPLEX optimiser within the Visual Studio Environment to find the optimal matching of LH-BH routes.

5.1. Data sets:

The computational experiments are reported for three data sets. Two of these (VRPB data *set-2* and *set-3,* see Toth and Vigo (1996, 1999) and Goetschalckx and Jacobs-Blecha (1989) for details) are available in the literature, and the MT-VRPB *set-1* is generated in this study.

Table 1: Details of the data *set-1*.

Problem number	Problem Name	*n*	*L*	*B*	*C*	*v*	*z**
1	eil22_50	21	11	10	6000	1,…,3	371
2	eil22_66	21	14	7	6000	1,…,3	366
3	eil22_80	21	17	4	6000	1,…,3	375
4	eil23_50	22	11	11	4500	1,…,3	677
5	eil23_66	22	15	7	4500	1,…,3	640
6	eil23_80	22	18	4	4500	1,…,2	623
7	eil30_50	29	15	14	4500	1,…,2	501
8	eil30_66	29	20	9	4500	1,…,3	537
9	eil30_80	29	24	5	4500	1,…,3	514
10	eil33_50	32	16	16	8000	1,…,3	738
11	eil33_66	32	22	10	8000	1,…,3	750
12	eil33_80	32	26	6	8000	1,…,3	736
13	eil51_50	50	25	25	160	1,…,3	559
14	eil51_66	50	34	16	160	1,…,4	548
15	eil51_80	50	40	10	160	1,…,4	565
16	eilA76_50	75	37	38	140	1,…,6	738
17	eilA76_66	75	50	25	140	1,…,7	768
18	eilA76_80	75	60	15	140	1,…,8	781
19	eilA101_50	100	50	50	200	1,…,5	827
20	eilA101_66	100	67	33	200	1,…,6	846
21	eilA101_80	100	80	20	200	1,…,7	859

n: number of customers; *C*: vehicle capacity; *v*: number of bins
*z**: free fleet VRPB solution

To test our model we have generated a set of new MT-VRPB instances, *set-1,* from 21 instances of *set-2* using the original VRPB and MT-VRP conventions established in Toth and Vigo (1996, 1999) and in Taillard *et al*. (1997), respectively. We have generated 168 problem instances by using different values of v (where v is the number of bins, (i.e., 1,…, 4), starting with an integer between one and the maximum number of bins) and T (where T is a maximum driving time for each bin). Two values of $T$are used, T_1 and T_2 for each value of v, where T_1 and T_2 are calculated as follows:

$$T_1 = [1.05\, z^*/v] \qquad\qquad T_2 = [1.1\, z^*/v]$$

The resulting values of both T_1 and T_2 are rounded up to the nearest integer, where z^* represents the VRPB solution obtained by our *Two-Level VNS* algorithm using a free vehicle fleet.

Several MT-VRPB instances are generated from each VRPB problem using T_1 and T_2 with the linehaul percentage of 50, 66, and 80%, respectively. Further details of the new MT-VRPB data *set-1* containing solutions (*z**) and free fleet (*v*) found by *Two-Level VNS* algorithm are provided in Table 1. All data sets can be downloaded from the CLHO website (CLHO, 2015).

5.2. Results and analysis:

Our *sweep-first-assignment-second* approach is very fast in producing an initial feasible solution, spending less than a second on average.

The optimal solutions and upper/lower bounds for the MT-VRPB are reported in Table 3 and Table 4 for T_1 and T_2, respectively. For each instance the CPLEX time was fixed to 2 hours. A reasonable number of optimal solutions are found for both $T_1$and T_2 groups of instances, ranging in size between 21 and 50 customers along with an instance of size 100 of T_2. Within the allocated time, CPLEX found 60 optimal solutions (i.e., T_1= 24, T_2= 36) out of all the 168 instances. The instances for which CPLEX could not find the solutions or reported as infeasible is due to either the bin(s) given time restriction and/or the instances are too large in size. We report upper bound and lower bound for those instances. CPLEX reported infeasibility in four cases where the number of bins increases and hence the given time decreases for each bin.

Insert Table 3 and Table 4 here

Table 5 and Table 6 report the detailed solutions of the *Two-Level VNS* algorithm along with the CPLEX results for the data *set-1* ($T_1$and T_2). The algorithm is run for 200 iterations and, due to the random element, best solution is reported out of 5 runs. For $T_1$the algorithm found a number of good quality (no overtime used) solutions (45 out of 84) and for the rest 39, it took less than 30 units of overtime in most cases. For T_2, 54 solutions are found without overtime and the rest (apart from a few) the algorithm did not exceed 30 units of overtime. Nonetheless, the algorithm is able to solve all the instances including 51 optimal solutions at a very low computational cost requiring on average 18 seconds per instance.

Insert Table 5 and Table 6 here

Table 7: The summary comparison of the *Two-Level VNS* and CPLEX (data *set-1*: T_1 & T_2)

	T_1		T_2	
	CPLEX	***Two-Level VNS***	**CPLEX**	***Two-Level VNS***
# of solutions found (out of 84)	24	84	36	84
# of optimal solutions found (out of 84)	24	21	36	30
Max overtime	-	58	-	52
Min overtime	-	2	-	1
Average overtime	-	10.24	-	5.33
Average CPU time (s)	5165	18	4248	17

It can be observed (see Table 5 and Table 6) that good quality solutions are found when the bin capacity is relatively larger and the number of bins is smaller. It can also be seen that with the increase in the number of bins, the likelihoods of overtime being used also increases. A further analysis of the results is provided in Table 7.

Special case – the VRPB: The *Two-Level VNS* algorithm is also tested on the VRPB where the best known results are reported. The VRPB data *set-2* and *set-3* are tested for a fixed number of iterations (400) which was deemed acceptable in terms of the solution quality and the affordable time. The algorithm produced very competitive results for both data sets. The detailed results are provided in Appendix (see Table 8 and Table 9 for data *set-2* and *set-3*, respectively). The algorithm performed extremely well when compared to the best known solution from the literature, with an overall average relative percentage deviation of 0.00 and 0.06 for *set-2* and *set-3*, respectively. In addition, all the best known solutions for *set-2* and 51 out of 62 in *set-3* are found to be the best known.

6. Conclusion

This study introduces a new VRP variant called the Multiple Trip Vehicle Routing Problem with Backhauls (MT-VRPB). An ILP mathematical formulation of the problem is produced and a new MT-VRPB data set is generated. The formulation is tested using CPLEX, and found optimal solutions for small and medium size data instances. To solve the larger instances of the problem a *Two-Level VNS* algorithm is developed that uses skeletons of the classical VNS and VND methodologies. A number of neighbourhoods and local searches are employed in a way to achieve diversification at the outer level (basic VNS) of the algorithm and intensification at the inner-level (VND with multi-layer local search framework). The algorithm found promising solutions when compared with the solutions found by CPLEX. Moreover, the algorithm is also tested on two classical VRPB instances data sets from the literature and found competitive results. It can therefore be said that this study also demonstrates the excellence and the power of VNS yet again in terms of its simplicity, flexibility, efficacy and speed.

References

Alonso F, Alvarez MJ, Beasley JE. A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions. Journal of Operational Research Society 2008; 59: 963-976.

Azi N, Gendreau M, Potvin J-Y. An adaptive large neighbourhood search for a vehicle routing problem with multiple routes. Computers & Operations Research 2014; 41: 167-173.

Brandao J. A new tabu search algorithm for the vehicle routing problem with backhauls. European Journal of Operational Research 2006; 173: 540-555.

Brandao J, Mercer A. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. European Journal of Operational Research 1997; 100: 180-191.

Brandao J, Mercer A. The multi-trip vehicle routing problem. Journal of the Operational Research Society 1998; 49: 799-805.

Cuervo DP, Goos P, Sorensen K, Arraiz E. An iterated local search algorithm for the vehicle routing problem with backhauls. European Journal of Operational Research 2014; 237: 454-464.

Cattaruzza D, Absi N, Feillet D, Vidal T. A memetic algorithm for the multi trip vehicle routing problem. European Journal of Operational Research 2014a; 236: 833-848.

Cattaruzza D, Absi N, Feillet D, Vigo D. An iterated local search for the multi-commodity multi-trip vehicle routing problem with time windows. Computers & Operations Research 2014b; 51: 257-267.

Cuervo DP, Goos P, Sorensen K, Arraiz E. An iterated local search algorithm for the vehicle routing problem with backhauls. European Journal of Operational Research 2014; 237: 454-464.

CHLO: Centre for Logistics and heuristic optimisation, 2015, Available from: http://www.kent.ac.uk/kbs/research/research-centres/clho/ [23 March 2015].

Gillet BE, Miller LR. A heuristics algorithm for the vehicle dispatch problem. Operations Research 1974; 22: 340-349.

Goetschalckx M, Jacobs-Blecha C. The vehicle routing problem with backhauls. European Journal of Operational Research 1989; 42: 39-51.

Gajpal Y, Abad PL. Multi-ant colony system (MACS) for a vehicle routing problem with backhauls. European Journal of Operational Research 2009; 196: 102-117.

Imran A, Salhi S, Wassan NA. A variable neighbourhood-based heuristic for the heterogeneous fleet vehicle routing problem. European Journal of Operational Research 2009; 197: 509-518.

Jacobs-Blecha C, Goetschalckx M. The vehicle routing problem with backhauls: Properties and solution algorithms. Georgia Institute of Technology; 1993 Technical Report MHRC-TR-88-13.

Laporte G. The vehicle routing problem: An overview of exact and approximate algorithms. European Journal of Operational Research 1992; 59: 345-358.

Lin C, Chow C, Chen C. A location-routing-loading problem for bill delivery services. Computers & Industrial Engineering 2002; 43: 5-25.

Lin C, Kwok CR. Multi-objective metaheuristics for a location-routing problem with multiple use of vehicles on real data and simulated data. European Journal of Operational Research 2006; 175: 1833-1849.

Mingozzi A, Giorgi S, Baldacci R. An exact method for vehicle routing problem with backhauls. Transportation Science 1999; 33: 315-329.

Mladenovic N, Hansen P. Variable neighbourhood search. Computers & Operations Research 1997; 24: 1097-1100.

Mladenovic N, Todosijevic R, Urosevic D. Two level general variable neighbourhood search for attractive travelling salesman problem. Computers & Operations Research 2014; 52: 341-348.

Macedo R, Alves C, Valerio de Carvalho J, Clautiaux F, Hanafi S. Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model. European Journal of Operational Research 2011; 214: 536-545.

Macedo R, Alves C, Hanafi S, Jarbouid B, Mladenovic N, Ramos B, Valerio de Carvalho JM. Skewed general variable neighborhood search for the location routing scheduling problem. Computers & Operations Research 2015; (to appear).

Osman IH, Wassan NA. A reactive tabu meta-heuristic for the vehicle routing problem with back-hauls. Journal of Scheduling 2002; 5: 263-285.

Olivera A, Viera O. Adaptive memory programming for the vehicle routing problem with multiple trips. Computers and Operations Research 2007; 34: 28-47.

Ong JO, Suprayogi. Vehicle routing problem with backhaul, multiple trips and time windows. Journal Teknik Industri 2011; 13: 1-10.

Petch RJ, Salhi S. A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. Discrete Applied Mathematics 2004; 133: 69-92.

Ropke S, Pisinger D. A unified heuristic for a large class of Vehicle Routing Problems with Backhauls. European Journal of Operational Research 2006; 171: 750-775.

Salhi S. The integration of routing into the location-allocation and vehicle composition problem. PhD Thesis. University of Lancaster, Lancaster, UK; 1987.

Salhi S, Sari M. A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. European Journal of Operational Research 1997; 103: 95-112.

Salhi S, Petch RJ. A GA based heuristic for the vehicle routing problem with multiple trips. Journal of Mathematical Modelling and Algorithms 2007; 6: 591-316.

Salhi S, Wassan N, Hajarat M. The Fleet Size and Mix Vehicle Routing Problem with Backhauls: Formulation and Set Partitioning-based Heuristics. Transportation Research Part E 2013; 56: 22-35.

Toth P, Vigo D. An Exact Algorithm for the vehicle routing problem with backhauls. Transportation Science 1997; 31: 372-385.

Toth P, Vigo D. A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls. European Journal of Operational Research 1999; 113: 528-543.

Taillard ED, Laporte G, Gendreau M. Vehicle routing with multiple use of vehicles. European Journal of Operational Research 1996; 47: 1065-1070.

Wassan N. Reactive tabu adaptive memory programming search for the vehicle routing problem with backhauls. Journal of Operational Research Society 2007; 58: 1630-1641.

Yano C, Chan L, Richter L, Cutler T, Murty K, McGettigan D. Vehicle routing at Quality Stores. Interfaces 1987; 17: 52-63.

Zachariadis E, Kiranoudis C. An effective local search approach for the vehicle routing problem with backhauls. Expert Systems with Applications 2012; 39: 3174-3184.

Glossary:

T_1: Total driving time (type one) for each bin in an instance.

T_2: Total driving time (type two) for each bin in an instance.

v: Total number of bins in each instance.

No. of Routes in each Bin: Number of routes served by each bin.

x: Infeasible.

NF: Not found.

Overtime: Overtime (equivalent to per unit distance travelled by a vehicle) allocated to bin(s) where needed to feasibly pack routes within bin(s).

Cost with overtime: Total solution cost including Overtime units.

Time(s): CPU time in seconds taken to solve each instance.

n: Total number of customers.

RPD: Relative Percentage Deviation = [(VNS Sol. - best known)/ best known * 100].

Table 3: Detailed CPLEX results for the data *set-1* (T_1)

Name	T_1	*v*	Optimal Sol.	No. Routes	No. of Routes in each Bin	Actual Time (s)	UB	LB
eil22_50	390	1	371	3	b1(3)	1.04	371.0000	367.5294
	195	2	378	3	b1(1), b2(2)	1.17	378.0000	368.0119
	130	3	x	x	x	x	x	x
eil22_66	385	1	366	3	b1(3)	1.01	366.0000	364.9640
	193	2	382	4	b1(2), b2(2)	3.02	382.0000	366.0000
	129	3	x	x	x	x	x	x
eil22_80	394	1	375	3	b1(3)	1.94	375.0000	362.1650
	197	2	378	4	b1(2), b2(2)	2.39	378.0000	364.9665
	132	3	381	3	b1(1), b2(1), b3(1)	27.13	381.0000	369.0667
eil23_50	711	1	677	3	b1(3)	0.33	677.0000	677.0000
	355	2	698	3	b1(2), b2(1)	2.36	698.0000	671.8600
	237	3	x	x	x	x	x	x
eil23_66	672	1	640	3	b1(1)	1.22	640.0000	633.1636
	336	2	640	3	b1(2), b2(1)	1.4	640.0000	635.5000
	224	3	x	x	x	x	x	x
eil23_80	654	1	623	2	b1(2)	1.44	623.0000	618.0870
	327	2	634	2	b1(1), b2(2)	1.59	634.0000	613.3380
eil30_50	526	1	501	2	b1(2)	0.44	501.0000	500.3902
	264	2	x	x		x	x	x
eil30_66	564	1	537	3	b1(3)	2.68	537.0000	511.3725
	282	2	552	3	b1(1), b2(2)	6116	552.0000	537.0000
	188	3	NF	NF	NF	7200	NF	533.7612
eil30_80	540	1	514	3	b1(3)	11.95	514.0000	474.9762
	270	2	NF	NF	NF	7200	NF	459.3289
	180	3	NF	NF	NF	7200	NF	460.3190
eil33_50	775	1	738	3	b1(3)	0.51	738.0000	738.0000
	388	2	NF	NF	NF	7200	NF	738.3900
	258	3	NF	NF	NF	7200	NF	740.7581
eil33_66	788	1	750	3	b1(3)	2.23	750.0000	732.7999
	394	2	772	3	b1(2), b2(1)	1219.03	772.0000	757.8079
	263	3	NF	NF	NF	7200	NF	746.4629
eil33_80	773	1	736	3	b1(3)	121.27	736.0000	733.8901
	387	2	NF	NF	NF	7200	NF	720.3275
	258	3	NF	NF	NF	7200	NF	690.0837
eil51_50	587	1	559	3	b1(3)	9.84	559.0000	552.1063
	294	2	NF	NF	NF	7200	NF	550.1111
	196	3	NF	NF	NF	7200	NF	553.0000
eil51_66	576	1	548	4	b1(4)	22.23	548.0000	537.7475
	288	2	NF	NF	NF	7200	NF	546.1393
	192	3	NF	NF	NF	7200	NF	542.1467
	144	4	NF	NF	NF	7200	NF	522.9460

eil51_80	594	1	565	4	b1(4)	4552.8	565.0000	553.1885
	297	2	NF	NF	NF	7200	NF	555.5726
	198	3	NF	NF	NF	7200	NF	556.1191
	149	4	NF	NF	NF	7200	NF	556.1018
eilA76_50	775	1	NF	NF	NF	7200	NF	708.2119
	388	2	NF	NF	NF	7200	NF	721.9806
	259	3	NF	NF	NF	7200	NF	721.8691
	194	4	NF	NF	NF	7202	NF	711.64.91
	155	5	NF	NF	NF	7200	NF	705.6147
	130	6	NF	NF	NF	7200	NF	708.1701
eilA76_66	807	1	NF	NF	NF	7200	NF	738.1007
	404	2	NF	NF	NF	7200	NF	737.9937
	269	3	NF	NF	NF	7200	NF	734.0403
	202	4	NF	NF	NF	7200	NF	739.9000
	162	5	NF	NF	NF	7200	NF	733.5028
	135	6	NF	NF	NF	7200	NF	739.4740
	116	7	NF	NF	NF	7200	NF	737.0274
eilA76_80	821	1	NF	NF	NF	7200	NF	739.7246
	411	2	NF	NF	NF	7200	NF	726.3083
	274	3	NF	NF	NF	7200	NF	733.6667
	206	4	NF	NF	NF	7200	NF	733.5946
	165	5	NF	NF	NF	7200	NF	732.5992
	137	6	NF	NF	NF	7200	NF	724.3518
	118	7	NF	NF	NF	7200	NF	723.4398
	103	8	NF	NF	NF	7200	NF	718.6787
eilA101_50	869	1	NF	NF	NF	7200	NF	799.5710
	435	2	NF	NF	NF	7200	NF	804.1183
	290	3	NF	NF	NF	7200	NF	802.2318
	218	4	NF	NF	NF	7200	NF	807.1541
	174	5	NF	NF	NF	7200	NF	767.5958
eilA101_66	889	1	NF	NF	NF	7200	NF	829.5004
	445	2	NF	NF	NF	7200	NF	837.3865
	297	3	NF	NF	NF	7200	NF	826.1638
	223	4	NF	NF	NF	7200	NF	815.4809
	178	5	NF	NF	NF	7200	NF	832.78.09
	149	6	NF	NF	NF	7200	NF	816.1044
eilA101_80	902	1	NF	NF	NF	7200	NF	827.3494
	451	2	NF	NF	NF	7200	NF	797.3486
	301	3	NF	NF	NF	7200	NF	790.1850
	226	4	NF	NF	NF	7200	NF	820.9844
	181	5	NF	NF	NF	7200	NF	821.9659
	151	6	NF	NF	NF	7200	NF	799.1573
	129	7	NF	NF	NF	7200	NF	825.4779

Table 4: Detailed CPLEX results for the Data set-1 (T_2)

Name	T_2	*v*	Optimal Sol.	No. Routes	No. of Routes in each Bin	Actual Time (s)	UB	LB
eil22_50	408	1	371	3	b1 (3)	0.89	371.0000	370.6087
	204	2	375	3	b1(2), b2(1)	1.67	375.0000	374.0333
	137	3	378	3	b1(1), b2(1), b3(1)	1.22	378.0000	364.4367
eil22_66	403	1	366	3	b1(3)	1.3	366.0000	364.7095
	201	2	382	4	b1(2), b2(2)	1.67	382.0000	366.0000
	134	3	366	3	b1(1), b2(1), b3(1)	0.59	366.0000	366.0000
eil22_80	413	1	375	3	b1(3)	2.72	375.0000	358.9261
	206	2	378	4	b1(2), b2(2)	8.5	378.0000	362.2288
	138	3	381	3	b1(1), b2(1), b3(1)	24.21	381.0000	364.9274
eil23_50	745	1	677	3	b1(3)	0.33	677.0000	677.0000
	372	2	689	3	b1(2), b2(1)	1.98	689.0000	680.0000
	248	3	716	3	b1(1), b2(1), b3(1)	2.46	716.0000	682.1268
eil23_66	704	1	640	3	b1(3)	0.75	640.0000	640.0000
	352	2	640	3	b1(1), b2(2)	1.23	640.0000	631.5000
	235	3	NF	NF	NF	7200	NF	662.4548
eil23_80	685	1	623	2	b1(2)	0.91	623.0000	617.8667
	343	2	631	2	b1(1), b2(1)	1.4	631.0000	614.5388
eil30_50	551	1	501	2	b1(2)	0.44	501.0000	500.3902
	276	2	501	2	b1(1), b2(1)	0.73	501.0000	501.0000
eil30_66	591	1	537	3	b1(3)	3.09	537.0000	510.3183
	296	2	552	3	b1(1), b2(2)	3451.24	552.0000	538.0355
	197	3	538	3	b1(1), b2(1), b3(1)	1.56	538.0000	534.6250
eil30_80	565	1	514	3	b1(3)	10.58	514.0000	482.8207
	283	2	535	3	b1(2), b2(1)	5519.11	535.0000	468.6333
	188	3	518	3	b1(1), b2(1), b3(1)	1426.17	518.0000	500.1891
eil33_50	812	1	738	3	b1(1)	0.44	738.0000	738.0000
	406	2	741	3	b1(2), b2(1)	2.26	741.0000	736.2820
	271	3	NF	NF	NF	7200	803.0000	658.5384
eil33_66	825	1	750	3	b1(3)	11.7	750.0000	734.5884
	413	2	767	3	b1(2), b2(1)	109.26	767.0000	764.4997
	275	3	NF	NF	NF	7200	NF	746.9500
eil33_80	810	1	736	3	b1(3)	136.31	736.0000	716.7393
	405	2	NF	NF	NF	7200	NF	723.4224
	270	3	NF	NF	NF	7200	NF	696.3739
eil51_50	615	1	559	3	b1(3)	11.23	559.0000	553.6224
	308	2	560	4	b1(2), b2(2)	67.17	560.0000	550.4380
	205	3	564	4	b1(2), b2(1), b3(1)	67.49	573.0000	559.6480
eil51_66	603	1	548	4	b1(4)	11.87	548.0000	541.1877
	302	2	548	4	b1(2), b2(2)	55.52	548.0000	546.9363
	201	3	NF	NF	NF	7200	NF	521.0965
	151	4	NF	NF	NF	7200	NF	539.9353

eil51_80	622	1	565	4	b1(4)	78.13	565.0000	562.5255
	311	2	NF	NF	NF	7200	NF	554.3046
	208	3	NF	NF	NF	7200	NF	553.8339
	156	4	NF	NF	NF	7200	NF	554.7640
eilA76_50	812	1	NF	NF	NF	7200	NF	710.0593
	406	2	NF	NF	NF	7200	NF	722.0668
	271	3	NF	NF	NF	7201	NF	720.4398
	203	4	NF	NF	NF	7202	NF	705.7348
	163	5	NF	NF	NF	7200	NF	706.7157
	136	6	NF	NF	NF	7200	NF	719.6408
eilA76_66	845	1	NF	NF	NF	7200	NF	734.9762
	423	2	NF	NF	NF	7200	NF	741.8414
	282	3	NF	NF	NF	7200	NF	734.1823
	212	4	NF	NF	NF	7200	NF	742.2662
	169	5	NF	NF	NF	7200	NF	738.0464
	141	6	NF	NF	NF	7200	NF	736.3244
	121	7	NF	NF	NF	7200	NF	733.6417
eilA76_80	860	1	NF	NF	NF	7200	NF	741.6530
	430	2	NF	NF	NF	7200	NF	732.6903
	287	3	NF	NF	NF	7200	NF	733.3761
	215	4	NF	NF	NF	7200	NF	733.4002
	172	5	NF	NF	NF	7200	NF	730.9763
	144	6	NF	NF	NF	7200	NF	731.1909
	123	7	NF	NF	NF	7200	NF	722.2782
	108	8	NF	NF	NF	7200	NF	733.8520
eilA101_50	910	1	NF	NF	NF	7200	NF	801.4182
	455	2	NF	NF	NF	7200	NF	813.7763
	304	3	NF	NF	NF	7200	NF	808.5073
	228	4	NF	NF	NF	7200	NF	803.0867
	182	5	NF	NF	NF	7200	NF	781.9759
eilA101_66	931	1	846	6	b1(6)	268.45	846.0000	840.8321
	466	2	NF	NF	NF	7200	NF	822.6394
	311	3	NF	NF	NF	7200	NF	831.4000
	233	4	NF	NF	NF	7200	NF	825.1924
	187	5	NF	NF	NF	7200	NF	814.6440
	156	6	NF	NF	NF	7200	NF	835.2673
eilA101_80	945	1	NF	NF	NF	7200	NF	828.6658
	473	2	NF	NF	NF	7200	NF	808.3282
	315	3	NF	NF	NF	7200	NF	819.9952
	237	4	NF	NF	NF	7200	NF	803.4907
	189	5	NF	NF	NF	7200	NF	817.7601
	158	6	NF	NF	NF	7200	NF	812.1149
	135	7	NF	NF	NF	7200	NF	816.7851

Table 5: Detailed comparison of the *Two-Level VNS* with CPLEX for the Data *set-1* (T_1)

Name	T_1	*v*	CPLEX			*Two-Level VNS*				
			Optimal Sol.	No. Routes	Actual Time (s)	Cost	Overtime	Cost with overtime	No. Routes	Time (s)
eil22_50	390	1	**371**	3	1.04	371	0	**371**	3	**2**
	195	2	**378**	3	1.17	378	0	**378**	3	**3**
	130	3	x	x	x	380	10	390	4	3
eil22_66	385	1	**366**	3	1.01	366	0	**366**	3	5
	193	2	**382**	4	3.02	386	10	396	4	4
	129	3	x	x	x	366	4	370	3	3
eil22_80	394	1	**375**	3	1.94	375	0	**375**	3	4
	197	2	**378**	4	2.39	378	0	**378**	4	5
	132	3	**381**	3	27.13	381	0	**381**	3	3
eil23_50	711	1	**677**	3	0.33	677	0	**677**	3	3
	355	2	**698**	3	2.36	677	34	711	3	2
	237	3	x	x	x	712	13	725	3	5
eil23_66	672	1	**640**	3	1.22	640	0	**640**	3	4
	336	2	**640**	3	1.4	640	0	**640**	3	4
	224	3	x	x	x	655	47	702	3	3
eil23_80	654	1	**623**	2	1.44	623	0	**623**	2	4
	327	2	**634**	2	1.59	634	0	**634**	2	4
eil30_50	526	1	**501**	2	0.44	501	0	**501**	2	4
	264	2	x	x	x	501	6	507	2	3
eil30_66	564	1	**537**	3	2.68	537	0	**537**	3	6
	282	2	**552**	3	6116	544	21	565	3	6
	188	3	NF	NF	7200	539	2	541	3	5
eil30_80	540	1	**514**	3	11.95	514	0	**514**	3	6
	270	2	NF	NF	7200	517	23	540	3	7
	180	3	NF	NF	7200	518	0	518	3	6
eil33_50	775	1	**738**	3	0.51	738	0	**738**	3	5
	388	2	NF	NF	7200	738	28	766	3	6
	258	3	NF	NF	7200	764	58	822	3	4
eil33_66	788	1	**750**	3	2.23	750	0	**750**	3	9
	394	2	**772**	3	1219	772	0	**772**	3	8
	263	3	NF	NF	7200	752	40	792	3	5
eil33_80	773	1	**736**	3	121.3	736	0	**736**	3	6
	387	2	NF	NF	7200	756	0	756	3	9
	258	3	NF	NF	7200	736	30	766	3	5
eil51_50	587	1	**559**	3	9.84	559	0	**559**	3	9
	294	2	NF	NF	7200	570	9	579	3	11
	196	3	NF	NF	7200	568	6	574	3	10
eil51_66	576	1	**548**	4	22.23	548	0	**548**	4	10
	288	2	NF	NF	7200	552	0	552	4	11
	192	3	NF	NF	7200	552	25	577	4	11

	144	4	NF	NF	7200	563	20	583	4	10
eil51_80	594	1	**565**	4	4553	565	0	**565**	4	13
	297	2	NF	NF	7200	565	0	565	4	12
	198	3	NF	NF	7200	580	23	603	5	11
	149	4	NF	NF	7200	581	11	592	5	11
eilA76_50	775	1	NF	NF	7200	838	0	838	6	21
	388	2	NF	NF	7200	738	0	738	6	23
	259	3	NF	NF	7200	741	0	741	6	22
	194	4	NF	NF	7202	738	49	787	6	23
	155	5	NF	NF	7200	747	36	783	6	22
	130	6	NF	NF	7200	748	31	779	6	22
eilA76_66	807	1	NF	NF	7200	768	0	768	7	23
	404	2	NF	NF	7200	768	0	768	7	21
	269	3	NF	NF	7200	772	0	772	7	23
	202	4	NF	NF	7200	784	0	784	8	21
	162	5	NF	NF	7200	781	36	817	8	23
	135	6	NF	NF	7200	783	5	788	8	23
	116	7	NF	NF	7200	771	22	793	8	22
eilA76_80	821	1	NF	NF	7200	781	0	781	8	23
	411	2	NF	NF	7200	781	0	781	8	23
	274	3	NF	NF	7200	784	0	784	8	22
	206	4	NF	NF	7200	787	0	787	8	23
	165	5	NF	NF	7200	785	3	788	8	23
	137	6	NF	NF	7200	800	7	807	9	24
	118	7	NF	NF	7200	792	24	816	8	23
	103	8	NF	NF	7200	796	38	834	8	23
eilA101_50	869	1	NF	NF	7200	827	0	827	5	39
	435	2	NF	NF	7200	835	0	835	5	42
	290	3	NF	NF	7200	847	2	849	5	42
	218	4	NF	NF	7200	849	6	855	5	42
	174	5	NF	NF	7200	833	30	863	5	41
eilA101_66	889	1	NF	NF	7200	846	0	846	6	43
	445	2	NF	NF	7200	846	0	846	6	41
	297	3	NF	NF	7200	846	0	846	6	42
	223	4	NF	NF	7200	866	0	866	6	43
	178	5	NF	NF	7200	846	28	874	6	43
	149	6	NF	NF	7200	874	32	906	7	42
eilA101_80	902	1	NF	NF	7200	859	0	859	7	42
	451	2	NF	NF	7200	859	0	859	7	45
	301	3	NF	NF	7200	859	0	859	7	45
	226	4	NF	NF	7200	770	5	775	7	42
	181	5	NF	NF	7200	869	17	886	7	43
	151	6	NF	NF	7200	863	23	886	7	42
	129	7	NF	NF	7200	859	46	905	7	44

Table 6: Detailed comparison of the *Two-Level VNS* with CPLEX for the Data *set-1* (T_2)

Name	T_2	*v*	CPLEX			*Two-Level VNS*				
			Optimal Sol.	No. Routes	Actual Time (s)	Cost	Overtime	Cost with overtime	No. Routes	Time (s)
eil22_50	408	1	**371**	3	0.89	371	0	**371**	3	3
	204	2	**375**	3	1.67	375	0	**375**	3	4
	137	3	**378**	3	1.22	380	2	382	3	3
eil22_66	403	1	**366**	3	1.3	366	0	**366**	3	2
	201	2	**382**	4	1.67	382	3	385	4	3
	134	3	**366**	3	0.59	366	1	367	3	2
eil22_80	413	1	**375**	3	2.72	375	0	**375**	3	3
	206	2	**378**	4	8.5	378	0	**378**	4	3
	138	3	**381**	3	24.21	381	0	**381**	3	4
eil23_50	745	1	**677**	3	0.33	677	0	**677**	3	4
	372	2	**689**	3	1.98	677	17	694	3	5
	248	3	**716**	3	2.46	716	0	**716**	3	4
eil23_66	704	1	**640**	3	0.75	640	0	**640**	3	4
	352	2	**640**	3	1.23	640	0	**640**	3	4
	235	3	NF	NF	7200	667	4	671	3	5
eil23_80	685	1	**623**	2	0.91	623	0	**623**	2	4
	343	2	**631**	2	1.4	631	0	**631**	2	4
eil30_50	551	1	**501**	2	0.44	501	0	**501**	2	4
	276	2	**501**	2	0.73	501	0	**501**	2	3
eil30_66	591	1	**537**	3	3.09	537	0	**537**	3	6
	296	2	**552**	3	3451.2	544	8	**552**	3	7
	197	3	**538**	3	1.56	538	0	**538**	3	5
eil30_80	565	1	**514**	3	10.58	514	0	**514**	3	6
	283	2	**535**	3	5519.1	535	0	**535**	3	7
	188	3	**518**	3	1426.2	518	0	**518**	3	5
eil33_50	812	1	**738**	3	0.44	738	0	**738**	3	4
	406	2	**741**	3	2.26	738	10	748	3	8
	271	3	NF	NF	7200	764	35	799	3	4
eil33_66	825	1	**750**	3	11.7	750	0	**750**	3	5
	413	2	**767**	3	109.26	767	0	**767**	3	9
	275	3	NF	NF	7200	754	21	775	3	5
eil33_80	810	1	**736**	3	136.31	736	0	**736**	3	8
	405	2	NF	NF	7200	756	0	756	3	6
	270	3	NF	NF	7200	736	18	754	3	6
eil51_50	615	1	**559**	3	11.23	559	0	**559**	3	10
	308	2	**560**	4	67.17	560	0	**560**	4	9
	205	3	**564**	4	67.49	568	0	568	3	11
eil51_66	603	1	**548**	4	11.87	548	0	**548**	4	10
	302	2	**548**	4	55.52	548	0	**548**	4	11
	201	3	NF	NF	7200	558	4	562	4	10

	151	4	NF	NF	7200	563	7	570	4	11
eil51_80	622	1	**565**	4	78.13	565	0	**565**	4	11
	311	2	NF	NF	7200	565	0	565	4	10
	208	3	NF	NF	7200	587	0	587	4	10
	156	4	NF	NF	7200	579	0	579	5	10
eilA76_50	812	1	NF	NF	7200	838	0	838	6	21
	406	2	NF	NF	7200	838	0	838	6	22
	271	3	NF	NF	7201	838	0	838	6	22
	203	4	NF	NF	7202	738	29	767	6	22
	163	5	NF	NF	7200	747	28	775	6	24
	136	6	NF	NF	7200	747	15	762	6	21
eilA76_66	845	1	NF	NF	7200	768	0	768	7	22
	423	2	NF	NF	7200	768	0	768	7	21
	282	3	NF	NF	7200	772	0	772	7	22
	212	4	NF	NF	7200	769	0	769	7	22
	169	5	NF	NF	7200	777	13	790	8	23
	141	6	NF	NF	7200	778	5	783	8	22
	121	7	NF	NF	7200	771	6	777	8	22
eilA76_80	860	1	NF	NF	7200	781	0	781	8	23
	430	2	NF	NF	7200	781	0	781	8	22
	287	3	NF	NF	7200	783	0	783	8	23
	215	4	NF	NF	7200	783	0	783	8	22
	172	5	NF	NF	7200	783	0	783	8	22
	144	6	NF	NF	7200	786	10	796	8	23
	123	7	NF	NF	7200	792	13	805	8	23
	108	8	NF	NF	7200	795	46	841	8	22
eilA101_50	910	1	NF	NF	7200	827	0	827	5	41
	455	2	NF	NF	7200	827	0	827	5	41
	304	3	NF	NF	7200	840	3	843	5	43
	228	4	NF	NF	7200	838	9	847	5	42
	182	5	NF	NF	7200	838	13	851	5	42
eilA101_66	931	1	**846**	6	268.45	846	0	**846**	6	43
	466	2	NF	NF	7200	846	0	846	6	42
	311	3	NF	NF	7200	846	0	846	6	43
	233	4	NF	NF	7200	853	10	863	6	42
	187	5	NF	NF	7200	848	14	862	6	43
	156	6	NF	NF	7200	852	52	904	6	44
eilA101_80	945	1	NF	NF	7200	859	0	859	7	42
	473	2	NF	NF	7200	859	0	859	7	43
	315	3	NF	NF	7200	859	0	859	7	46
	237	4	NF	NF	7200	859	0	859	7	43
	189	5	NF	NF	7200	863	15	878	7	44
	158	6	NF	NF	7200	870	13	883	7	45
	135	7	NF	NF	7200	859	24	883	7	42

Appendix

Table 8: Detailed results of the VRPB (data *set-2*)

Name	*n*	*L*	*B*	*V*	*VCap*	Best Known	*Two-Level VNS*	RPD
eil22_50	21	11	10	3	6000	371	371	0.00
eil22_66	21	14	7	3	6000	366	366	0.00
eil22_80	21	17	4	3	6000	375	375	0.00
eil23_50	22	11	11	2	4500	682	682	0.00
eil23_66	22	15	7	2	4500	649	649	0.00
eil23_80	22	18	4	2	4500	623	623	0.00
eil30_50	29	15	14	2	4500	501	501	0.00
eil30_66	29	20	9	3	4500	537	537	0.00
eil30_80	29	24	5	3	4500	514	514	0.00
eil33_50	32	16	16	3	8000	738	738	0.00
eil33_66	32	22	10	3	8000	750	750	0.00
eil33_80	32	26	6	3	8000	736	736	0.00
eil51_50	50	25	25	3	160	559	559	0.00
eil51_66	50	34	16	4	160	548	548	0.00
eil51_80	50	40	10	4	160	565	565	0.00
eilA76_50	75	37	38	6	140	739	739	0.00
eilA76_60	75	50	25	7	140	768	768	0.00
eilA76_80	75	60	15	8	140	781	781	0.00
eilB76_50	75	37	38	8	100	801	801	0.00
eilB76_66	75	50	25	10	100	873	873	0.00
eilB76_80	75	60	15	12	100	919	919	0.00
eilC76_50	75	37	38	5	180	713	713	0.00
eilC76_66	75	50	25	6	180	734	734	0.00
eilC76_80	75	60	15	7	180	733	733	0.00
eilD76_50	75	37	38	4	220	690	690	0.00
eilD76_66	75	50	25	5	220	715	715	0.00
eilD76_80	75	60	15	6	220	694	694	0.00
eilA101_50	100	50	50	4	200	831	831	0.00
eilA101_66	100	67	33	6	200	846	846	0.00
eilA101_80	100	80	20	6	200	856	856	0.00
eilB101_50	100	50	50	7	112	923	923	0.00
eilB101_66	100	67	33	9	112	983	983	0.00
eilB101_80	100	80	20	11	112	1008	1008	0.00

Name = instance name; ***n*** = number of total customers in each instance; ***L*** = number of linehaul customers; ***B*** = number of backhaul customers; ***V*** = fixed fleet; ***VCap*** = vehicle capacity; **Best Known** = best VRPB solution found in literature to date; **Two-Level VNS** = solution found by proposed algorithm; **RPD** = relative percentage deviation.

Table 9: Detailed results of the VRPB (Data *set-3*)

Name	*n*	*L*	*B*	*VCap*	*V*	Best known Solution	*Two-level VNS* Solution	RPD
A1	25	20	5	1550	8	229885.65	229885.65	0.00
A2	25	20	5	2550	5	180119.21	180119.21	0.00
A3	25	20	5	4050	4	163405.38	163405.38	0.00
A4	25	20	5	4050	3	155796.41	155796.41	0.00
B1	30	20	10	1600	7	239080.16	239080.16	0.00
B2	30	20	10	2600	5	198047.77	198047.77	0.00
B3	30	20	10	4000	3	169372.29	169372.29	0.00
C1	40	20	20	1800	7	250556.77	250556.77	0.00
C2	40	20	20	2600	5	215020.23	215020.23	0.00
C3	40	20	20	4150	5	199345.96	199345.96	0.00
C4	40	20	20	4150	4	195366.63	195366.63	0.00
D1	38	30	8	1700	12	322530.13	322530.13	0.00
D2	38	30	8	1700	11	316708.86	316708.86	0.00
D3	38	30	8	2750	7	239478.63	239478.63	0.00
D4	38	30	8	4075	5	205831.94	205831.94	0.00
E1	45	30	15	2650	7	238879.58	238879.58	0.00
E2	45	30	15	4300	4	212263.11	212263.11	0.00
E3	45	30	15	5225	4	206659.17	206659.17	0.00
F1	60	30	30	3000	6	263173.96	263173.96	0.00
F2	60	30	30	3000	7	265214.16	265214.16	0.00
F3	60	30	30	4400	5	241120.78	241120.78	0.00
F4	60	30	30	5500	4	233861.85	233861.85	0.00
G1	57	45	12	2700	10	306305.40	306305.40	0.00
G2	57	45	12	4300	6	245440.99	245440.99	0.00
G3	57	45	12	5300	5	229507.48	229507.48	0.00
G4	57	45	12	5300	6	232521.25	232521.25	0.00
G5	57	45	12	6400	5	221730.35	221730.35	0.00
G6	57	45	12	8000	4	213457.45	213457.45	0.00
H1	68	45	23	4000	6	268933.06	268933.06	0.00
H2	68	45	23	5100	5	253365.50	253365.50	0.00
H3	68	45	23	6100	4	247449.04	247449.04	0.00
H4	68	45	23	6100	5	250220.77	250220.77	0.00
H5	68	45	23	7100	4	246121.31	246121.31	0.00
H6	68	45	23	7100	5	249135.32	249135.32	0.00
I1	90	45	45	3000	10	350245.28	350245.28	0.00
I2	90	45	45	4000	7	309943.84	309943.84	0.00
I3	90	45	45	5700	5	294507.38	294507.38	0.00
I4	90	45	45	5700	6	295988.45	295988.45	0.00
I5	90	45	45	5700	7	301236.01	301236.01	0.00
J1	94	75	19	4400	10	335006.68	335006.68	0.00
J2	94	75	19	5600	8	310417.21	310417.21	0.00
J3	94	75	19	8200	6	279219.21	279219.21	0.00
J4	94	75	19	6600	7	296533.16	296533.16	0.00
K1	113	75	38	4100	10	394071.17	394375.63	0.08
K2	113	75	38	5200	8	362130.00	362130.00	0.00
K3	113	75	38	5200	9	365694.08	365694.08	0.00
K4	113	75	38	6200	7	348949.39	348949.39	0.00
L1	150	75	75	4400	10	417896.72	417943.82	0.01
L2	150	75	75	5000	8	401228.80	401228.80	0.00
L3	150	75	75	5000	9	402677.72	403639.75	0.24
L4	150	75	75	6000	7	384636.33	384636.33	0.00
L5	150	75	75	6000	8	387564.55	387564.55	0.00
M1	125	100	25	5200	11	398593.19	398869.79	0.07
M2	125	100	25	5200	10	396916.97	397786.41	0.22
M3	125	100	25	6200	9	375695.42	377315.94	0.43
M4	125	100	25	8000	7	348140.16	348140.16	0.00
N1	150	100	50	5700	11	408100.62	408100.62	0.00
N2	150	100	50	5700	10	408065.44	408111.91	0.01
N3	150	100	50	6600	9	394337.86	397621.99	0.83
N4	150	100	50	6600	10	394788.36	398330.35	0.90
N5	150	100	50	8500	7	373476.30	373723.37	0.07
N6	150	100	50	8500	8	373758.65	376200.31	0.65