The Selective Traveling Salesman Problem with Draught Limits

Shahin Gelareh[a,*], Bernard Gendron[b], Saïd Hanafi[c], Rahimeh Neamatian Monemi[d,e], Raca Todosijević[f]

[a]Département de Réseaux et Télécommunications, IUT de Bethune, Université Artois, F-62400, Béthune, France
[b]CIRRELT, Université de Montréal C.P. 6128, Succursale Centre-Ville Montréal, Québec, Canada H3C 3J7
[c]LAMIH UMR CNRS 8201 - Université de Valenciennes, 59313 Valenciennes Cedex 9, France
[d]Southampton Business School, University of Southampton, Southampton, SO17 1BJ, United Kingdom
[e]Centre for Operational Research, Management Science and Information Systems (CORMSIS), University of Southampton, Southampton, SO17 1BJ, United Kingdom
[f]INRIA Lille-Nord Europe, 40 Avenue Halley 59650 Villeneuve dAscq, France

Abstract

This paper introduces the Selective Traveling Salesman Problem with Draught Limits, an extension of Traveling Salesman Problem with Draught Limits, wherein the goal is to design maximal profit tour respecting draught limit constraints of the visited ports. We propose a mixed integer linear programming formulation for this problem. The proposed mixed integer program is used to solve –to optimality– small size instances and also to asses the quality of heuristic solutions found in the paper. The heuristic solutions are obtained using a general variable neighborhood search heuristic that explores several neighborhood structures. Our extensive computational experiments confirm the efficiency of the method and the quality of the reported solutions.

1. Introduction

Recently, a new variant of the Traveling Salesman Problem (TSP) in the context of maritime transportation has been proposed in Rakke et al. (2012), which is referred to as Traveling Salesman problem with Draught Limits (TSPDL). The TSPDL consists of visiting and delivering goods to a set of ports using a ship located initially at a depot. Since each port has a demand known in advance, the ship starts its tour with a load equivalent to the total demand, visits each port exactly once and returns to the depot performing the lowest cost (length) tour. However, to each port a draught limit is assigned, which represents the maximal allowed load on the ship upon entering some port.

The existence of a feasible solution to such a problem is not guaranteed. Consider the following instance with $n = 4$ nodes (node 0 being the depot): each port has a demand of 1 unit of flow and each unit of flow imposes one unit of draught. Moreover, let the ports with draught of 2 units be located on vertices of a square starting from the depot to node 1, node 2 and node 3. This example has no feasible solution if all ports have to be visited. In reality, we are ready to reject customer demands but reducing the volume of on-board that guarantees the existence of a feasible solution. Here, dripping any of these ports (except depot) from the tour results guarantees existence of a feasible solution.

In industrial shipping, while production at the production plants follow a seasonal pattern, bulkers often do not have a published itinerary. In these cases, usually the volume of demand (the replenishment orders) at the destinations is not known sufficiently in advance (do not always follow a predictable pattern). Therefore, once the demands are known, the tramp bulkers decide on the subset of ports to be served and the order in which those ports shall be called. This is expected to be in such a way that port draught limits be respected, given the volume/weight on board of a vessels when entering a given port.

*Corresponding author, shahin.gelareh@{univ-artois.fr;gmail.com}

This work has been motivated by a real case wherein a construction company that is constructing tourism facilities in a regional archipelago outsources its requirement from some other subcontractors based at the mainland. In the mainland, a subcontractor receives orders on a weekly basis from almost all the construction sites and acquires a long list of requirements from raw materials (such as sand, cements etc.) and semi-manufacturer products (e.g. rolling mill products such as pillar, trestle, truss etc.) to the heavy construction equipment or their parts (such as boom, dipper, bucket etc.). As often the draught limits are very restrictive in such waters, the sequence of visits plays a major role in maximizing the revenue as a result of visiting more ports or delivering demands with higher revenue.

In this paper, we propose a variant of the TSPDL called the Selective Traveling Salesman Problem with Draught Limit (STSPDL). The differences between the STSPDL and the TSPDL are the following: i) visiting a port in the STSPDL results in collecting a ceratin amount of the profit assigned to that port; ii) unlike the TSPDL, in the STSPDL it is not required to visit all ports; and iii) the objective function of the STSPDL seeks to maximize the total profit captured on a tour, which equals the sum of profits of the visited ports minus the tour length.

Formally, the problem may be stated as follows: *Given an undirected graph $G = (V, E)$ where $V = \{0, 1, ..., n\}$ represents the set of ports including the starting port, i.e., the depot denoted by 0, while $E = \{(i, j)|i, j \in V, i \neq j\}$ represents the edge set where each edge (i, j) from the set E has the associated cost c_{ij}. For each port i, with the exception of the depot, a draught limit, l_i, a demand, d_i, and a profit p_i are given. Additionally, we denote by L_i the load on the ship upon entering port i, calculated relative to a given tour T. The tour T will be called a feasible tour if $L_i \leq l_i$ for all $i \in T$, otherwise it will be called an infeasible tour. The goal of the STSPDL is to design a (hopefully non-empty) maximal profit tour respecting draught limit constraint of each visited port. The total profit captured on a feasible tour T, will be denoted by $v(T)$. It is assumed that the vessel capacity is sufficient for delivering demand of all the ports.*

To the best of our knowledge, the STSPDL has never been studied previously. The problem closest to the STSPDL is the TSPDL. The TSPDL was introduced in Rakke et al. (2012) where the authors proposed two formulations of the TSPDL as well as valid inequalities and strengthened bounds. In order to tackle instances up to 48 nodes, they developed a branch-and-cut algorithm. Due to the intrinsic complexity of this problem, there are still instances within the test-bed in Rakke et al. (2012) that could not be solved to optimality using the proposed branch-and-cut. Later, Battarra et al. (2014) proposed three new formulations of the TSPDL, which enabled them to solve instances with up to 48 nodes to optimality using exact algorithms. In Todosijević et al. (2014), the authors proposed two variable neighborhood search based heuristics for solving TSPDL as well as the set of large size instances. The proposed heuristics turn out to be very efficient in solving known instances of TSPDL and represent the state-of-the-art heuristics.

Another class of problems closely related to the STSPDL includes Traveling Salesman Problems with Profits (TSPP). In Feillet et al. (2005), the Traveling Salesman Problems with Profits are classified relative to the way the objectives of minimizing distance and maximizing profit are handled. The authors distinguished the following three major classes of TSPP:

- **First class** - Problems in which both objectives are combined in a single objective function (see e.g., Dell'Amico et al. (1995)).

- **Second class** - Problems in which the travel cost is a constraint and the objective is to maximize the profits collected (see e.g., Golden et al. (1987); Kataoka and Morito (1988); Laporte and Martello (1990); Vansteenwegen et al. (2011)). These problems are usually referred to as Orienteering Problems.

- **Third class** - Problems in which the profit is a constraint and the objective is to minimize the travel cost (see e.g., Awerbuch et al. (1998); Balas (1989)). These problems are usually referred to as Prize-collecting TSPs.

The main contribution of this paper is to introduce, model and solve a new problem, named STSPDL, that arises in the context of maritime transport. The problem is NP-hard since the TSPDL is a special case of this problem, which is NP-hard problem by itself. In order to tackle the instances of STSPDL we propose an approach based on the general variable neighborhood search framework. The performance of this heuristic is assessed (in terms of solution quality and computational time) against the results obtained using a generic mixed integer programming solver on a diverse set of test instances.

The rest of the paper is organized as follows: In the next section we give a mathematical formulation of the STSPDL. Section 3 proposes a heuristic solution method based on general variable neighborhood search. Section 4 elaborates on the computational experiments carried out on a large set of instances. Section 5 concludes the work and highlights further research directions.

2. Mathematical formulation and preprocessing

2.1. Mixed integer programming formulation

The STSPDL may be formulated as a mixed integer programming problem (MIP) in the following way. To each potential edge $e \in E$ we associate two arcs (i, j) and (j, i) and define $A = \{(i, j)|i, j \in V\}$. The binary variable x_{ij} take the value 1, if and only if, the arc (i, j) is included in the solution. A continuous variable y_{ij} represents the load on the ship while traversing the arc (i, j). The binary decision variable z_i takes the value 1, if and only if, the port i belongs to the list of visited ports in the solution. The STSPDL may be stated as the following MIP:

$$\max \sum_{i \in V} p_i z_i - \sum_{(i,j) \in A} c_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{i \in V} x_{ij} = z_j \qquad \forall j \in V \tag{2}$$

$$\sum_{j \in V} x_{ij} = z_i \qquad \forall i \in V \tag{3}$$

$$\sum_{i \in V} y_{ij} - \sum_{i \in V} y_{ji} = d_j z_j \qquad \forall j \in V \setminus \{0\} \tag{4}$$

$$\sum_{i \in V} y_{0i} = \sum_{i \in V \setminus \{0\}} d_i z_i \tag{5}$$

$$\sum_{i \in V} y_{i0} = 0 \tag{6}$$

$$0 \leq y_{ij} \leq l_j x_{ij} \qquad \forall (i, j) \in A \tag{7}$$

$$x_{ij} + x_{ji} \leq 1 \qquad \forall (i, j) \in A \tag{8}$$

$$x_{ij} \in \{0, 1\}, y_{ij} \geq 0 \qquad \forall (i, j) \in A \tag{9}$$

The objective function (1) accounts for the profit for visiting a node minus the transit times over the edges. If node j is visited therefore there will be an arc arriving at the node and an arc leaving the node as it is stated in constraints (2)-(3). The flow conservation constraints (4) make sure that the difference between the input and output volumes is the demand of a node. The volume loaded on vessels is equal to the demand of nodes that are going to be visited. This is stated in constraints (5). Constraints (6) ensure that the last node visited will return to the depot. Finally, constraints (7) ensure that the load on a given edge (i, j) is below the draught limit of the tail node j. Constraints (8) are additionally added to reinforce the model.

2.2. Preprocessing

We address in this section some preprocessing techniques, which may reduce the size of the STSPDL problem. Based on the profit values associated to ports and the nature of draught limit constraints we may infer whether a should be visited in the optimal tour or not, as it is shown in the sequel. Let us define for each $j \in V \setminus \{0\}$, $\delta_j^{min} = min\{c_{ij} + c_{jk} - c_{ik} : (i, k) \in E \cup \{(0, 0)\}, i \neq j, k \neq j\}$.

Proposition 2.1. *If $p_j < \delta_j^{min}$, an optimal tour for the STSPDL does not contain port j.*

Proof. Let us suppose the contrary that there is an optimal tour T containing port j whose total cost equals $v(T)$. The total profit of a tour T' obtained from the tour T excluding node j will be $v(T') = v(T) - p_j + c_{ij} + c_{jk} - c_{ik}$, where i and k denote the predecessor and successor of port j relative to tour T. Since $p_j < \delta_j^{min} \leq c_{ij} + c_{jk} - c_{ik}$, it follows that $c_{ij} + c_{jk} - c_{ik} - p_j > 0$, thus $v(T') > v(T)$. Moreover, by dropping a port from a feasible tour, the resulting tour remains a feasible tour (see Section 3.2). Therefore, we can conclude that tour T' is a feasible tour with strictly greater objective function than tour T which is contradiction to the assumption that T is an optimal tour. □

Let us define for each $\delta_j^{max} = max\{c_{ij} + c_{jk} - c_{ik} : (i,k) \in E \cup \{(0,0)\}, i \neq j, k \neq j\}$.

Proposition 2.2. *If $p_j > \delta_j^{max}$, an optimal tour for the STSPDL with the relaxed draught limit constraints must contain port j.*

Proof. Let us assume that there is an optimal tour T that does not contain port j. If we insert port j at any position in the tour, we will get a feasible tour (since draught limit constraints are neglected) whose cost equals $v(T') = v(T) + p_j - c_{ij} - c_{jk} + c_{ik}$, where $v(T)$ denotes the total profit of tour T, while i, j denote the ports between which port j is inserted. Keeping in mind that $p_j > \delta_j^{max} \geq c_{ij} + c_{jk} - c_{ik}$, we infer that $p_j - c_{ij} - c_{jk} + c_{ik} > 0$ and therefore $v(T') > v(T)$. Hence, tour T' is a feasible tour with greater total profit than the optimal tour T, which is a contradiction. □

3. General Variable Neighborhood Search for STSPDL

Variable Neighborhood Search (VNS) (Mladenović and Hansen, 1997) is a flexible framework for building heuristics to solve combinatorial and continuous global optimization problems approximately. The main idea is to systematically explore several neighborhood structures during the search for an optimal (or near-optimal) solution. The foundations of VNS are based on the following observations: i) a local optimum relative to one neighborhood structure is not necessarily the local optimum for another neighborhood structure; ii) a global optimum is a local optimum with respect to all neighborhood structures; and iii) for many problems, empirical evidence shows that all local optima are relatively close to each other.

A VNS based heuristic consists of applying alternately an improvement procedure, a shaking procedure, and a neighborhood change step, until reaching predefined stopping condition(s). The improvement procedure used within VNS heuristic may be either simple local search that explores one neighborhood structure, or some more advanced procedure that explores several neighborhood structures. Such explorations could also be organized in different ways for instance *sequential Variable Neighborhood Descent (seqVND)* (see e.g., (Hansen et al., 2010) for more details), *Composite (Nested) VND* (see e.g., Todosijević et al. (2015) where full Nested VND was applied for the first time) or *Mixed Nested VND* (see e.g., Ilić et al. (2010) for a recent application of Mixed Nested VND). If some more advanced procedure is used as the improvement procedure, the resulting heuristic is called General Variable Neighborhood Search (GVNS) (see Hansen et al. (2010)).

On the other hand, shaking procedure is employed to help exiting a local optima traps where the improvement procedure has got stuck. Typical stopping criteria for VNS are maximal number of performed iterations, or maximum allowed CPU time, t_{max}. The VNS based heuristics have been successfully applied to solving many optimization problems (see e.g., (Hansen et al., 2010; Mladenović et al., 2016) for some recent successful applications).

In the rest of the section, we describe the main ingredients of our GVNS heuristic, i.e., an initial solution used, neighborhood structures, a sequential variable neighborhood descent and a shaking procedure.

3.1. Initial Solution

A relatively good initial solution can be produced by starting from an empty solution (a tour starting from and ending at the depot) and iteratively inserting into it new nodes and in a right order. In the sequel, we use ports and nodes, interchangeably.

3.2. Neighborhood structures and feasibility checking

The STSPDL structure suggests two types of neighborhood structures. The first type of neighborhoods are the ones that influence the choice of node (port) to be visited, while the second type neighborhoods are those that are concerned with the order of visiting the nodes. These neighborhoods stem from the classical neighborhoods used for solving the traveling salesman problem and the knapsack problem, respectively.

3.2.1. Location neighborhoods

The location neighborhoods used within our GVNS are based on the following moves defined relative to a given tour T:

- DROP_MOVE that removes a port from the tour T,

- ADD_MOVE that inserts a port not included in the tour T, between two ports visited consecutively in a tour T,

- SWAP_MOVE that replaces a port currently in the tour T by a port not including in the tour T.

Clearly, some of the neighbors of a given solution T with respect to the aforementioned neighborhoods are infeasible as the newly introduced node can result into a higher volume of supply from the depot and therefore infeasibility with respect to the draught limit.

Feasibility checking of DROP_MOVE. Given a feasible solution $T = (0, i_1, i_2, i_3, \dots i_{k-1}, i_k, i_{k+1}, \dots, i_m, 0)$, if we eliminate a port i_k from the tour, the ports on the subtour $(0, i_1, i_2, i_3, \dots i_{k-1})$ will be visited with smaller load onboard the ship, while the ports in the subtour $(i_{k+1}, \dots, i_m, 0)$ will be visited with the same load onboard the ship as in the current tour T. This means that draught limit constraints of all ports in the resulting tour $T' = (0, i_1, i_2, i_3, \dots i_{k-1}, i_{k+1}, \dots, i_m, 0)$ will be respected.

Feasibility checking of ADD_MOVE. To perform an ADD_MOVE that inserts a port $i_{j'}$ into a tour between nodes i_j and i_{j+1}, the feasibility of this move may be checked by only examining whether such a move preserves the feasibility on the first part of the tour, i.e. $(0, i_1, i_2, i_3, \dots, i_j)$. By inserting the new port $i_{j'}$, the ship must start a tour from the depot carrying larger volume (i.e., the load will be increased by an amount equal to the demand of port $i_{j'}$) and therefore draught limit constraints for all ports in the first part $(0, i_1, i_2, i_3, \dots, i_j)$ must be verified. However, as the vessel will be unloaded for the demand of port $i_{j'}$, the volume of onboard does not change in the second part, namely, $(i_{j+1}, \dots, i_m, 0)$.

Feasibility checking of SWAP_MOVE. The feasibility of a move which replaces the port i_j by a port $i_{j'}$ may be checked in three distinguished cases:

a) *Demands of ports i_j and $i_{j'}$ are equal, i.e., $d_{i_j} = d_{i_{j'}}$.* In this case, it is obvious that such swap move yields a feasible solution.
b) *The demand of port i_j is bigger than the demand of port $i_{j'}$, i.e., $d_{i_j} > d_{i_{j'}}$.* In this case, it is enough to check whether draught limits of ports at the part starting from the port i_{j+1} as well as the draught limit of the node $i_{j'}$ to be inserted will be violated or not once the move is performed.
c) *The demand of port i_j is smaller than the demand of port $i_{j'}$, i.e., $d_{i_j} < d_{i_{j'}}$.* In this case, it suffices to verify whether the feasibility will be preserved on the initial part of tour $(0, i_1, i_2, i_3, \dots, i_j)$ or not. The rationale for this case is the same as for the ADD_MOVE.

3.2.2. Traveling salesman neighborhoods

The traveling salesman neighborhood structures are those successfully applied for solving the TSPDL (see Todosijević et al. (2014)). They are based on the most common moves performed on a TSP solution, i.e., *2-opt* moves and OR-opt moves (Johnson and McGeoch, 1997). The *2-opt* move breaks down two edges of a current solution, and inserts two new edges by inverting the part of a solution in such a way that the resulting solution is still a tour. One variant of the 2-opt move is the so-called *1-opt* move which is applicable on four consecutive ports. On the other hand, an OR-opt move relocates a chain of consecutive customers without inverting any part of a solution. If a chain contains k ports, we call such move *OR-opt-*k move. If a chain of k consecutive ports is moved backward, that move will be called *backward OR-opt-*k. Similarly, if a chain is moved forward, the move will be called *forward OR-opt-*k. Each of these moves defines one neighborhood structure of the given TSP solution as a set of all solutions obtained by performing that move on the given TSP solution.

In order to recognize whether a given move is feasible or not, without actually performing it, we again need just to keep the array of load (L) updated and check whether the feasibility will be maintained on some part of a tour if we execute that move (for more details see Todosijević et al. (2014)) .

3.3. Sequential Variable Neighborhood Descent

The basic sequential VND procedure works as in the sequel. We create an ordered list of several neighborhood structures that are examined one after another respecting the established order. Let $\mathcal{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_{\ell_{max}}\}$ be a set of operators defining the neighborhood structures and the order of their examination. Starting from a given initial solution x, the basic sequential VND procedure iteratively explores its neighborhood structures in the predefined order $\mathcal{N}_\ell$, $1 \leq \ell \leq \ell_{max}$, one after another. As soon as an improvement of the incumbent solution in some neighborhood structure occurs, the basic sequential VND procedure resumes search in the next neighborhood structure of the new incumbent solution. The whole process is stopped if the current incumbent solution can not be improved with respect to any of ℓ_{max} neighborhood structures.

The previously described neighborhood structures are used to define two basic Sequential Variable Neighborhood Descent procedures. The first, one known as `seqVND-LOC`, examines the location neighborhoods, while the second one, called `seqVND-TSP`, examines the traveling salesman neighborhoods. The key steps of these two procedures are given in algorithm 1 (`XX` $\in \{$`TSP, LOC`$\}$, $l_{max} = 3$ in `seqVND-LOC` and $l_{max} = 8$ in `seqVND-TSP`). The procedure `seqVND-LOC` explores neighborhoods based on ADD_MOVE ($\mathcal{N}'_1$), SWAP_MOVE ($\mathcal{N}'_2$) and DROP_MOVE ($\mathcal{N}'_3$) moves in that order, one after another. On the other hand `SeqVND-TSP` procedure examines neighborhoods based on 1-opt ($\mathcal{N}''_1$), 2-opt ($\mathcal{N}''_2$), backward OR-opt-3 ($\mathcal{N}''_3$), forward OR-opt-3 ($\mathcal{N}''_4$), backward OR-opt-2 ($\mathcal{N}''_5$), forward OR-opt-2 ($\mathcal{N}''_6$), backward OR-opt-1 ($\mathcal{N}''_7$), forward OR-opt-1 ($\mathcal{N}''_8$) moves, respectively. Note that different orders to explore neighborhoods in `seqVND-LOC` and `seqVND-TSP` are investigated in our preliminary computational experiments, and the proposed orders are among the best ones.

These two sequential VND procedures are embedded in the improvement procedure that applies `seqVND-LOC` and `seqVND-TSP`, alternately, until reaching a solution that cannot be improved neither by `seqVND-LOC` nor by `seqVND-TSP`. The key steps of this procedure are given in algorithm 2. Note that the proposed `seq-VND` procedure may be also used as a constructive heuristic and thus it may be directly applied on an empty tour.

Algorithm 1: seqVND-XX

Function `seqVND-XX`(T)
$l \leftarrow 1$;
repeat
  $T' \leftarrow \texttt{argmax}\{v(T'') : T'' \in \mathcal{N}'_\ell(T), T'' feasible\}$;
  $l \leftarrow l + 1$;
  if $v(T') > v(T)$ **then**
    $T \leftarrow T'$; $l \leftarrow 1$;
  end
until $l > l_{max}$;
return T;

Algorithm 2: Improvement procedure

Function `Improvement_procedure`(T)
repeat
  $T' \leftarrow$ `seqVND-LOC`(T);
  $T'' \leftarrow$ `SeqVND-TSP`(T');
  if $v(T'') > v(T)$ **then**
    $T \leftarrow T''$;
  end
until $v(T'') \leq v(T)$;
return T;

3.4. *Shaking procedure*

In order to escape from local optima, the proposed GVNS uses the shaking function `Shake`(T, k) presented in algorithm 3. The shaking function at the output returns a solution obtained by preforming k times a random OR-opt-1 move on a given solution T. More precisely the shaking function at each iteration chooses at random one port from the tour T and moves it, either forward or backward, after another port, also chosen at random.

Algorithm 3: Shaking procedure

Function `Shake`(T, k)
for $i = 1$ **to** k **do**
  `Select` $T' \in$ `OR_opt_1`(T) `at random`;
  $T \leftarrow T'$;
end
return T;

3.5. *Steps of the proposed GVNS*

The steps of the GVNS that we propose for solving the STSPDL are presented in algorithm 4. For the local search step, our GVNS uses the procedure given in algorithm 2, while to diversify the search it applies the shaking procedure given in algorithm 3.

At the input, our GVNS procedure requires two parameters as shown in algorithm 4. The first one denoted by t_{max} represents the CPU time allowed before terminating a run, while the second one, named k_{max}, represents the maximum number of iterations that can be executed within the inner loop of the Shaking procedure. As it is presented in algorithm 4, before starting the main GVNS loop (Step 3), the improvement procedure is applied on the empty solution in order to try to construct a good quality solution.

Algorithm 4: GVNS for solving STSPDL

```
Function GVNS (k_max, t_max)
Create the initial solution T as described in Section 3.1;
T ← Improvement_procedure(T);
repeat
    k ← 1;
    while k ≤ k_max do
        T' ← Shake(T, k) ;
        T'' ← Improvement_procedure(T');
        k ← k + 1;
        if v(T'') > v(T) then
            T ← T''; k ← 1;
        end
    end
until CpuTime() > t_max;
return T;
```

4. Computational results

All experiments described in this section were carried out on an Intel(R) Core(TM) i7-4770 CPU@ 3.4Ghx(2x) and 16 Gb RAM. The proposed GVNS algorithm is coded in C++. CPLEX 12.6.3 has been used as a general MIP solver to solve the formulation presented in Section 2.

4.1. Test instances

We generated a testbed by adapting the benchmark instances of the TSPDL. The set of TSPDL benchmark instances contains 300 instances derived from 11 classical TSP instances (i.e. burma14, ulysses16, ulysses22, fri26, bayg29, gr17, gr22, gr48, KroA100, KroA200 and pcb442 —see Reinelt (1991)). A TSPDL instance is converted to an instance of the STSPDL by assigning profit values to all ports except the depot. The profit value p_j assigned to a port j is chosen as a random number from the interval $[\delta_j^{min}, \delta_j^{max}]$, where $\delta_j^{min} = \min\{c_{ij} + c_{jk} - c_{ik} : (i,k) \in E \cup \{(0,0)\}, i \neq j, k \neq j\}$ and $\delta_j^{max} = \max\{c_{ij} + c_{jk} - c_{ik} : (i,k) \in E \cup \{(0,0)\}, i \neq j, k \neq j\}$. The rationale for choosing this profit values is given in propositions 2.1 and 2.2.

4.2. Experiments with our testbed

For every instance with $|V|$ nodes, we let GVNS to run for $|V|$ seconds and record the time at which the best solution has been found. In GVNS, we have only one single parameter k_{max}, which is set in all experiments to 30.

In Table 1 and Table 2, the first column represents the instance name. The first block spanning from column two to four, represents the statistics of GVNS. The objective function of the GVNS solution is reported in the second column. 'Tb/Tx' represent the time at which the best know solution has been found (Tb) and the maximum available time (Tx). The column 'Len.' reports the number of nodes in a STSPDL tour.

Upon termination of GVNS, we inject its best found feasible solution to CPLEX as a MIPStart parameter and let CPLEX run for a time limit (tLim) of 100 seconds and a MIP relative gap (EpGap) of 1 percent. The summary of results are reported in the second block, i.e. 'MIPStart CPLEX'. '#Tol' and '#Opt' report the number of instances for which CPLEX terminated by optimal tolerance and optimal, respectively. 'Obj' represents the objective function at this state and 'T (sec.)' report the time spent in CPLEX. The best upper bound of CPLEX is reported in 'UB' and the relative gap in 'Gap'. Finally we report the number of processed branch-and-bound nodes in '#Nodes'. Here we set the emphasis on MIPEmphasisOptimality and generate the FracCuts and LiftProjCuts, aggressively.

We then run CPLEX with the same MIP relative gap and no MIP start while the time limit is set to the time limit of heuristic plus the previous time limit of CPLEX (i.e. $|V| + 100$). The same columns are present for the 'CPLEX (no MIPStart)'.

Table 1: The summary of experimental results on the small size instances.

	GVNS			MIPStart CPLEX							CPLEX (no MIPStart)						
instance	Obj	Tb/Tx	Len.	#Tol	#Opt	Obj	T (sec.)	UB	Gap	#Nodes	#Tol	#Opt	Obj	T (sec.)	UB	Gap	#Nodes
bayg29-10	5537.40	0.02	25.00	9	0	5537.40	11.49	5580.08	0.78	2043.30	9	9	5501.90	16.48	5570.71	1.30	1962.70
bayg29-25	5492.30	0.02	24.60	9	0	5492.30	14.47	5556.75	1.19	1989.20	7	1	5448.10	29.08	5532.01	1.61	5250.90
bayg29-50	5370.70	0.01	23.30	5	0	5370.70	60.17	5473.10	1.94	10683.60	3	0	5231.80	102.73	5471.48	4.74	19074.80
burma14-10	10197.64	0.01	13.09	2	8	10197.64	9.19	10212.74	0.66	1649.73	5	5	10196.09	11.92	10222.02	1.18	2874.18
burma14-25	10493.70	0.00	12.00	6	4	10493.70	0.52	10541.11	0.46	433.70	3	7	10491.30	1.23	10510.36	0.19	844.90
burma14-50	10347.20	0.00	11.90	5	5	10347.20	2.87	10388.48	0.41	3890.80	5	5	10345.50	3.82	10393.24	0.47	4082.80
fri26-10	3719.70	0.02	22.70	9	1	3719.70	0.78	3745.94	0.71	0.00	7	3	3711.70	1.71	3729.62	0.49	229.10
fri26-25	3638.50	0.01	20.60	4	1	3638.50	50.71	3689.81	1.42	11165.40	6	0	3618.00	62.62	3683.33	1.85	13975.20
fri26-50	3627.40	0.04	20.60	7	0	3627.40	34.52	3678.77	1.43	7723.40	6	1	3585.60	48.71	3666.97	2.35	11029.80
gr17-10	7105.40	0.00	14.90	9	1	7105.40	0.44	7149.00	0.62	217.80	5	5	7100.10	0.48	7123.47	0.33	188.30
gr17-25	7009.70	0.00	14.60	8	1	7009.70	16.19	7065.21	0.80	13737.60	8	1	7006.70	20.19	7047.27	0.59	20807.70
gr17-50	6813.20	0.00	13.80	6	2	6813.20	29.18	6878.82	0.98	24748.60	8	0	6800.10	38.29	6869.64	1.04	33681.90
gr21-10	9826.50	0.00	16.90	2	8	9826.50	0.31	9842.31	0.16	38.80	5	5	9821.40	0.56	9853.85	0.33	94.70
gr21-25	9783.90	0.00	16.90	5	5	9783.90	0.31	9814.15	0.31	0.00	7	3	9777.90	0.78	9809.81	0.33	0.00
gr21-50	9355.10	0.01	17.40	8	1	9355.10	19.44	9470.28	1.26	8362.60	9	0	9345.20	29.81	9451.55	1.16	13135.30
gr48-10	31988.10	0.82	42.90	10	0	31988.10	17.89	32270.74	0.89	883.00	6	0	31803.60	82.86	32166.88	1.15	4106.40
gr48-25	31573.00	2.77	42.10	2	0	31573.00	93.35	32070.19	1.58	4320.80	0	0	29391.60	148.04	32055.62	9.40	10507.70
gr48-50	31051.40	1.67	40.70	1	0	31051.40	90.23	32006.14	3.09	5624.20	1	0	27020.40	134.51	32009.71	20.60	11668.70
ulysses16-10	24654.30	0.00	14.90	10	0	24654.30	0.12	24854.64	0.81	0	6	4	24625.20	0.40	24712.61	0.35	0.00
ulysses16-25	24480.40	0.00	14.70	10	0	24480.40	0.65	24685.82	0.84	193.70	8	2	24439.40	0.80	24568.81	0.53	245.90
ulysses16-50	24201.20	0.00	14.20	8	1	24201.20	10.38	24413.65	0.89	8531.40	8	1	24176.10	12.26	24367.14	0.81	9659.10
ulysses22-10	32293.10	0.00	20.70	10	0	32293.10	0.34	32536.51	0.76	0.00	6	4	32239.80	0.60	32367.77	0.40	0.00
ulysses22-25	31953.40	0.01	19.80	9	0	31953.4	10.78	32231.21	0.90	3945.00	9	0	31854.50	16.04	32127.01	0.86	5102.70
ulysses22-50	31754.90	0.01	19.20	8	0	31754.90	21.99	32053.82	0.94	6325.80	8	0	31600.50	34.07	31966.85	1.18	10124.80

Table 2: The summary of experimental results on the large size instances.

	GVNS			MIPStart CPLEX							CPLEX (no MIPStart)						
instance	Obj	Tb/Tx	Len.	#Tol	#Opt	Obj	T (sec.)	UB	Gap	#Nodes	#Tol	#Opt	Obj	T (sec.)	UB	Gap	#Nodes
kroA100-50	300518.50	25.03	93.80	3	0	300518.50	73.12	304194.79	1.20	655.20	0	0	200646.80	200.15	303597.50	59.15	1524.60
kroA100-75	298835.30	31.55	91.80	0	0	298835.30	100.05	304101.27	1.60	696.90	0	0	188688.60	200.07	303683.10	68.79	1498.70
kroA200-50	629624.10	144.46	190.50	5	0	629626.70	80.31	636204.48	1.00	0.00	0	0	323999.90	300.16	635746.10	114.94	15.80
kroA200-75	625635.90	130.14	187.30	0	0	625635.90	100.05	635866.64	1.80	0.00	0	0	303303.00	300.35	635682.60	116.21	36.70
pcb442-50	1474133.60	411.00	422.30	10	0	1474133.60	50.93	1486468.80	1.00	0.00	0	0	0.00	542.21	1485854.00	N.A.	0
pcb442-75	1470333.90	398.98	417.30	3	0	1470333.90	85.47	1486426.15	1.00	0.00	0	0	0.00	542.20	1485586.00	N.A.	0

In Table 1, one observes that in all cases the average solution values reported by 'MIPStart' coincide with the ones provided by GVNS. In 162 case from among the 240 instances, the solution reported by GVNS is indeed optimal while in 38 case, CPLEX shows that the solution are within the optimality tolerance. In 'no MIPStart', this figures reach to 145 and 56, respectively. One also observes that the average deviation from optimality in the solutions reported by GVNS is less than 2 percent (except in one case). In general the number of processed branch-and-bound nodes is reduced in the 'MIPStart' when compared to the 'no MIPStart' case.

From Table 2, we may conclude the following. Neither the 'no MIPStart' nor the 'MIPstart' do not succeed to solve any instance to optimality. Yes, 'no MIPStart' does not succeed to find a solution within the optimality tolerance for any instance, while 'MIPStart' succeeds to do so for 21 out of 60 instances. On all data sets, except one instance, average solution values reported by 'MIPStart' coincide with the ones provided by GVNS (the difference on this one data set comes from the one test instance where CPLEX succeeds to improve further the solution provided by GVNS (see Appendix)). In addition, CPLEX confirms that the average percentage of deviation from optimality in solutions reported by GVNS is less than 2 percent. On the other hand 'no MIPStart' reports the solutions of very poor quality as may be observed from the reported average percentage deviations from the optimality. When comparing 'MIPStart' and 'no MIPStart' scenarios we see that the injected GVNS solution helps to significantly reduce the average number of processed branch-and-bound nodes.

In Figure 1 and Figure 2, one observes that 'no MIPstart' exhibits a very poor performance compared to the GVNS and 'MIPstart' on both data sets. This is especially true on the large scale instances. On the small instances, 'MIPstart' and GVNS exhibit a similar performance, while on large instances 'MIPstart' is slightly better since on just one instance succeeded to improve the solution provided by GVNS. On the small instances, 'no MIPstart' on about 85 percent of instances provides solutions with percentage deviations not greater than 1% from the best known values, while on about 95 percent of instances the deviation is not greater than 6.5 percent. In addition, on less than half of the instances 'no MIPstart' succeeds to reach best known values. On the other hand on large scale instances on less than 10 percent of instances solution values provided by 'no MIPstart' deviate from the best known values by no more than 20%. This fact clearly shows a poor performance of 'no MIPstart' approach on larger instances.

5. Concluding remarks

The main contribution of this paper is to introduce a new variant of the well-known traveling salesman problem arising in maritime transport (STSPDL), develop the corresponding mathematical model and propose a very efficient solution approach.

The proposed solution algorithm is based on the general variable neighborhood search (GVNS) framework. The proposed GVNS examines eleven neighborhood structures. These neighborhood structures are explored in an efficient way using feasibility checking procedures able to quickly recognize feasible solutions.

Our extensive computational experiments confirm that the proposed solution approach is capable of producing optimal solution as long as it is at hand and outperforms CPLEX 12.6 general MIP solver with respect to the time and/or solution quality.

Future work may include exact approaches or matheuristics in order to further enhance the results presented here. We will particularly focus on identifying some valid inequalities and proposing a branch-and-cut framework.

References

Awerbuch, B., Azar, Y., Blum, A., Vempala, S., 1998. New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. SIAM Journal on computing 28 (1), 254–262.

Balas, E., 1989. The prize collecting traveling salesman problem. Networks 19 (6), 621–636.

Battarra, M., Pessoa, A. A., Subramanian, A., Uchoa, E., 2014. Exact algorithms for the traveling salesman problem with draft limits. European Journal of Operational Research 235 (1), 115–128.

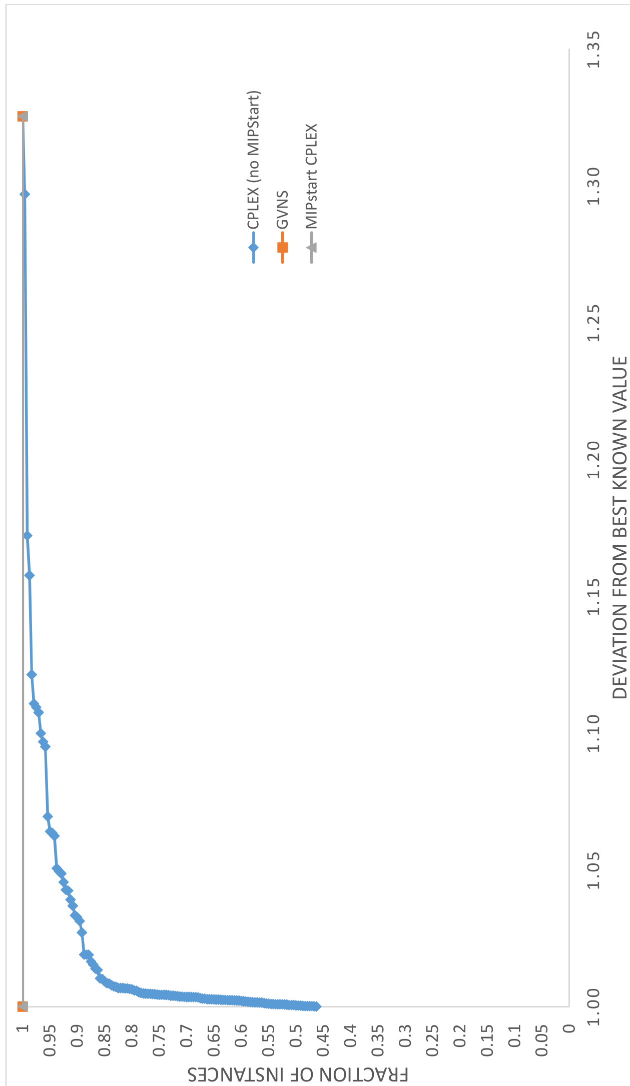

Figure 1: The performance profile for the small instances.

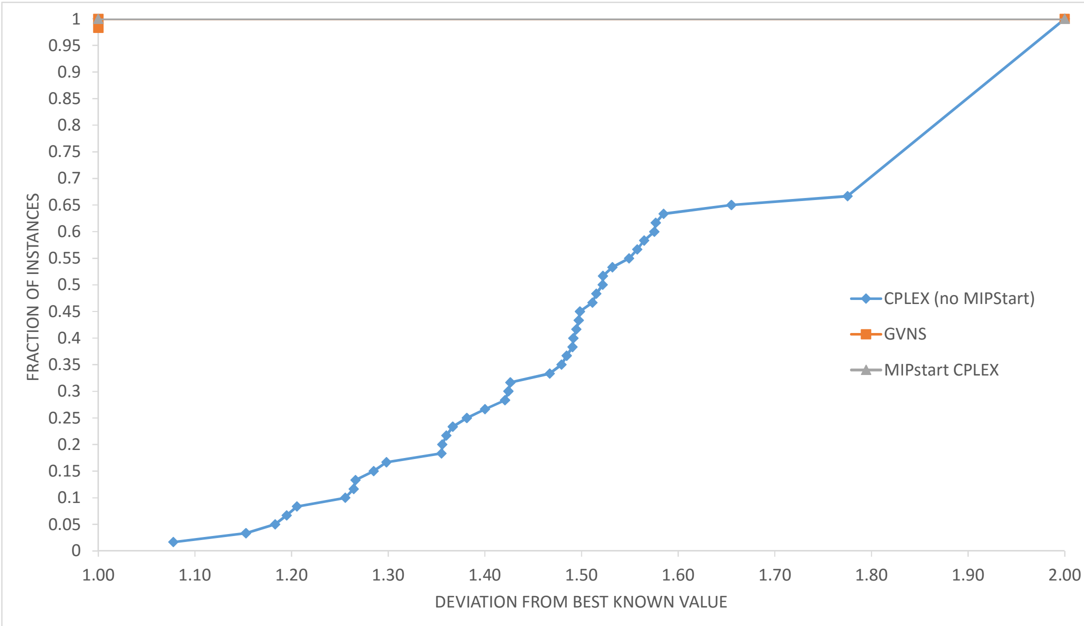

Figure 2: The performance profile for the large instances.

Dell'Amico, M., Maffioli, F., Värbrand, P., 1995. On prize-collecting tours and the asymmetric travelling salesman problem. International Transactions in Operational Research 2 (3), 297–308.

Feillet, D., Dejax, P., Gendreau, M., 2005. Traveling salesman problems with profits. Transportation science 39 (2), 188–205.

Golden, B. L., Levy, L., Vohra, R., 1987. The orienteering problem. Naval Research Logistics (NRL) 34 (3), 307–318.

Hansen, P., Mladenović, N., Pérez, J. A. M., 2010. Variable neighbourhood search: methods and applications. Annals of Operations Research 175 (1), 367–407.

Ilić, A., Urošević, D., Brimberg, J., Mladenović, N., 2010. A general variable neighborhood search for solving the uncapacitated single allocation p-hub median problem. European Journal of Operational Research 206 (2), 289–300.

Johnson, D. S., McGeoch, L., 1997. The traveling salesman problem: a case study in local optimization. In: Aarts, E., Lenstra, J. (Eds.), Local Search in Combinatorial Optimization. John Wiley and Sons, New York, pp. 215–310.

Kataoka, S., Morito, S., 1988. An algorithm for single constraint maximum collection problem. J. OPER. RES. SOC. JAPAN. 31 (4), 515–530.

Laporte, G., Martello, S., 1990. The selective travelling salesman problem. Discrete applied mathematics 26 (2), 193–207.

Mladenović, N., Hansen, P., 1997. Variable neighborhood search. Computers & Operations Research 24 (11), 1097–1100.

Mladenović, N., Todosijević, R., Urošević, D., 2016. Less is more: Basic variable neighborhood search for minimum differential dispersion problem. Information Sciences 326, 160–171.

Rakke, J. G., Christiansen, M., Fagerholt, K., Laporte, G., 2012. The traveling salesman problem with draft limits. Computers & Operations Research 39 (9), 2161–2167.

Reinelt, G., 1991. TSPLIB - a traveling salesman problem library. ORSA Journal on Computing 3, 376–384.

Todosijević, R., Mjirda, A., Mladenović, M., Hanafi, S., Gendron, B., 2014. A general variable neighborhood search variants for the travelling salesman problem with draft limits. Optimization Letters.

Todosijević, R., Urošević, D., Mladenović, N., Hanafi, S., 2015. A general variable neighborhood search for solving the uncapacitated r-allocation p-hub median problem. Optimization Letters doi10.1007/s11590-015-0867-6.

Vansteenwegen, P., Souffriau, W., Van Oudheusden, D., 2011. The orienteering problem: A survey. European Journal of Operational Research 209 (1), 1–10.

Appendix A

Table 1 and Table 2 report detailed computational result on the instances of moderate and large size.

Table 3: Numerical experiments on the instances of small size.

	GVNS			MIPStart CPLEX						CPLEX (no MIPStart)					
instance	Obj	Tb/Tx	Len.	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes
bayg29-10-1-p	5607.0	0.001/29	25	OptimalTol	5607.0	0.22	5635.07	0.5	0	OptimalTol	5570.0	0.74	5615.9	0.82	0
bayg29-10-2-p	5583.0	0.032/29	26	OptimalTol	5583.0	0.56	5630.41	0.85	0	OptimalTol	5575.0	3.14	5609.29	0.62	0
bayg29-10-3-p	5502.0	0.002/29	25	OptimalTol	5502.0	12.12	5557.01	1.0	3335	OptimalTol	5501.0	25.83	5545.0	0.8	6076
bayg29-10-4-p	5601.0	0.001/29	24	OptimalTol	5601.0	0.27	5624.36	0.42	0	OptimalTol	5593.0	0.91	5611.25	0.33	0
bayg29-10-5-p	5607.0	0.003/29	25	OptimalTol	5607.0	0.26	5623.31	0.29	0	OptimalTol	5569.0	0.68	5614.09	0.81	0
bayg29-10-6-p	5607.0	0.004/29	25	OptimalTol	5607.0	0.31	5633.74	0.48	0	OptimalTol	5586.0	0.65	5614.71	0.51	0
bayg29-10-7-p	5421.0	0.007/29	25	OptimalTol	5421.0	0.42	5473.54	0.97	0	OptimalTol	5418.0	0.66	5451.58	0.62	0
bayg29-10-8-p	5275.0	0.013/29	25	AbortTLim	5275.0	100.02	5384.72	2.08	17098	AbortTLim	5069.0	129.02	5426.27	7.05	13540
bayg29-10-9-p	5594.0	0.1/29	26	OptimalTol	5594.0	0.27	5614.86	0.37	0	OptimalTol	5561.0	0.41	5612.95	0.93	0
bayg29-10-10-p	5577.0	0.007/29	24	OptimalTol	5577.0	0.42	5623.8	0.84	0	OptimalTol	5577.0	2.76	5606.06	0.52	11
bayg29-25-1-p	5225.0	0.006/29	25	AbortTLim	5225.0	100.01	5445.62	4.22	11079	AbortTLim	4961.0	129.02	5373.13	8.31	21935
bayg29-25-2-p	5559.0	0.095/29	26	OptimalTol	5559.0	0.7	5610.27	0.92	0	OptimalTol	5519.0	6.85	5574.17	1.0	2559
bayg29-25-3-p	5560.0	0.051/29	25	OptimalTol	5560.0	0.46	5614.44	0.98	0	OptimalTol	5545.0	6.53	5581.44	0.66	1493
bayg29-25-4-p	5576.0	0.001/29	23	OptimalTol	5576.0	0.3	5617.7	0.75	0	OptimalTol	5573.0	2.34	5599.3	0.47	0
bayg29-25-5-p	5552.0	0.003/29	24	OptimalTol	5552.0	0.58	5604.68	0.95	0	OptimalTol	5528.0	3.81	5583.19	1.0	0
bayg29-25-6-p	5430.0	0.002/29	24	OptimalTol	5430.0	0.23	5438.32	0.15	0	Optimal	5430.0	0.46	5430.0	0.0	0
bayg29-25-7-p	5486.0	0.006/29	25	OptimalTol	5486.0	37.86	5540.86	1.0	7814	AbortTLim	5440.0	129.01	5531.16	1.68	23617
bayg29-25-8-p	5577.0	0.002/29	24	OptimalTol	5577.0	0.5	5628.24	0.92	0	OptimalTol	5553.0	0.67	5606.54	0.96	0
bayg29-25-9-p	5549.0	0.009/29	25	OptimalTol	5549.0	1.03	5604.3	1.0	0	OptimalTol	5525.0	9.41	5580.19	1.0	2512
bayg29-25-10-p	5409.0	0.012/29	25	OptimalTol	5409.0	3.03	5463.04	1.0	999	OptimalTol	5407.0	2.65	5461.01	1.0	393
bayg29-50-1-p	5470.0	0.098/29	24	OptimalTol	5470.0	40.47	5524.69	1.0	10044	AbortTLim	5386.0	129.03	5516.71	2.43	19194
bayg29-50-2-p	5041.0	0.006/29	23	AbortTLim	5041.0	100.01	5348.94	6.11	15821	AbortTLim	4998.0	129.02	5388.95	7.82	31616
bayg29-50-3-p	5286.0	0.005/29	24	AbortTLim	5286.0	100.02	5373.7	1.66	18147	AbortTLim	4919.0	129.02	5375.47	9.28	20663
bayg29-50-4-p	5355.0	0.004/29	24	OptimalTol	5355.0	43.22	5408.55	1.0	8926	AbortTLim	5267.0	129.02	5395.89	2.45	25321
bayg29-50-5-p	5560.0	0.004/29	23	OptimalTol	5560.0	0.49	5611.14	0.92	0	OptimalTol	5543.0	4.18	5579.43	0.66	1475
bayg29-50-6-p	5322.0	0.003/29	21	AbortTLim	5322.0	100.03	5401.61	1.5	17916	AbortTLim	5095.0	129.01	5438.61	6.74	18005
bayg29-50-7-p	5284.0	0.004/29	24	AbortTLim	5284.0	100.02	5465.27	3.43	16567	AbortTLim	4949.0	129.02	5476.59	10.66	25881
bayg29-50-8-p	5502.0	0.007/29	24	OptimalTol	5502.0	14.19	5557.01	1.0	2749	OptimalTol	5502.0	113.66	5510.16	0.15	24644
bayg29-50-9-p	5508.0	0.007/29	24	OptimalTol	5508.0	3.21	5563.06	1.0	1147	OptimalTol	5508.0	6.29	5555.74	0.87	1403
bayg29-50-10-p	5379.0	0.001/29	22	AbortTLim	5379.0	100.02	5477.01	1.82	15519	AbortTLim	5151.0	129.02	5477.22	6.33	22546
burma14-10-1-p	10628.0	0.001/14	12	Optimal	10628.0	0.06	10628.0	0.0	0	Optimal	10628.0	0.08	10628.0	0.0	0
burma14-10-2-p	10718.0	0/14	12	Optimal	10718.0	0.08	10718.0	0.0	0	Optimal	10718.0	0.09	10718.0	0.0	0
burma14-10-3-p	10718.0	0.002/14	12	Optimal	10718.0	0.06	10718.0	0.0	0	OptimalTol	10718.0	0.06	10719.0	0.01	0
burma14-10-4-p	10355.0	0.001/14	12	OptimalTol	10355.0	0.43	10411.5	0.55	0	OptimalTol	10355.0	0.87	10356.0	0.01	0
burma14-10-5-p	10718.0	0/14	12	OptimalTol	10718.0	0.06	10776.5	0.55	0	OptimalTol	10718.0	0.05	10776.5	0.55	0
burma14-10-6-p	10718.0	0.001/14	12	Optimal	10718.0	0.16	10718.0	0.0	0	Optimal	10718.0	0.16	10718.0	0.0	0
burma14-10-7-p	10718.0	0.001/14	12	Optimal	10718.0	0.05	10718.0	0.0	0	OptimalTol	10718.0	0.15	10817.9	0.93	0
burma14-10-8-p	10695.0	0.001/14	12	Optimal	10695.0	0.06	10695.0	0.0	0	Optimal	10695.0	0.14	10695.0	0.0	0
burma14-10-9-p	10628.0	0.002/14	12	Optimal	10628.0	0.08	10628.0	0.0	0	Optimal	10628.0	0.41	10628.0	0.0	0
burma14-10-10-p	10718.0	0.001/14	12	Optimal	10718.0	0.06	10718.0	0.0	0	OptimalTol	10718.0	0.06	10806.4	0.82	0
burma14-25-1-p	10114.0	0.001/14	12	OptimalTol	10114.0	1.78	10215.1	1.0	2101	OptimalTol	10114.0	5.97	10167.3	0.53	4559
burma14-25-2-p	10599.0	0.001/14	12	Optimal	10599.0	0.17	10599.0	0.0	0	Optimal	10599.0	0.39	10599.0	0.0	0

Table 3: Numerical experiments on the instances of small size.

	GVNS			MIPStart CPLEX						CPLEX (no MIPStart)					
instance	Obj	Tb/Tx	Len.	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes
burma14-25-3-p	10641.0	0/14	12	Optimal	10641.0	0.06	10641.0	0.0	0	Optimal	10641.0	0.17	10641.0	0.0	0
burma14-25-4-p	10509.0	0.001/14	12	OptimalTol	10509.0	0.18	10576.2	0.64	0	Optimal	10509.0	0.34	10509.0	0.0	0
burma14-25-5-p	10718.0	0.002/14	12	OptimalTol	10718.0	0.06	10762.9	0.42	0	Optimal	10718.0	0.68	10718.0	0.0	0
burma14-25-6-p	10496.0	0.002/14	12	Optimal	10496.0	0.16	10496.0	0.0	0	Optimal	10496.0	0.45	10496.0	0.0	0
burma14-25-7-p	10695.0	0.001/14	12	OptimalTol	10695.0	0.05	10750.8	0.52	0	Optimal	10695.0	0.33	10695.0	0.0	0
burma14-25-8-p	10651.0	0.001/14	12	Optimal	10651.0	0.3	10651.0	0.0	0	Optimal	10651.0	0.15	10651.0	0.0	0
burma14-25-9-p	10292.0	0.003/14	12	OptimalTol	10292.0	1.14	10394.9	1.0	989	OptimalTol	10292.0	2.3	10327.3	0.34	2403
burma14-25-10-p	10222.0	0.003/14	12	OptimalTol	10222.0	1.3	10324.2	1.0	1247	OptimalTol	10198.0	1.48	10300.0	1.0	1487
burma14-50-1-p	10178.0	0.002/14	12	OptimalTol	10178.0	1.31	10279.6	1.0	2068	OptimalTol	10178.0	2.02	10279.7	1.0	1881
burma14-50-2-p	10654.0	0.002/14	12	OptimalTol	10654.0	0.07	10667.0	0.12	0	Optimal	10654.0	0.11	10654.0	0.0	0
burma14-50-3-p	10141.0	0.001/14	12	OptimalTol	10141.0	0.88	10242.3	1.0	437	OptimalTol	10141.0	1.11	10242.3	1.0	514
burma14-50-4-p	10654.0	0.00/14	12	Optimal	10654.0	0.06	10654.0	0.0	0	Optimal	10654.0	0.14	10654.0	0.0	0
burma14-50-5-p	10480.0	0.001/14	12	Optimal	10480.0	0.35	10480.0	0.0	0	Optimal	10480.0	0.59	10480.0	0.0	0
burma14-50-6-p	10176.0	0.001/14	12	Optimal	10176.0	0.32	10176.0	0.0	0	OptimalTol	10167.0	0.6	10252.3	0.84	0
burma14-50-7-p	10718.0	0.00/14	12	OptimalTol	10718.0	0.04	10819.3	0.95	0	OptimalTol	10718.0	0.33	10811.6	0.87	0
burma14-50-8-p	10498.0	0.001/14	12	Optimal	10498.0	0.22	10498.0	0.0	0	Optimal	10498.0	0.5	10498.0	0.0	0
burma14-50-9-p	10413.0	0.001/14	12	Optimal	10413.0	0.07	10413.0	0.0	0	Optimal	10413.0	0.33	10413.0	0.0	0
burma14-50-10-p	9560.0	0.004/14	11	OptimalTol	9560.0	25.41	9655.6	1.0	36403	OptimalTol	9552.0	32.51	9647.51	1.0	38433
fri26-10-1-p	3731.0	0.005/26	23	OptimalTol	3731.0	0.41	3760.19	0.78	0	Optimal	3731.0	0.81	3731.0	0.0	0
fri26-10-2-p	3731.0	0.023/26	23	OptimalTol	3731.0	0.43	3759.54	0.76	0	OptimalTol	3724.0	0.92	3737.0	0.35	0
fri26-10-3-p	3696.0	0.01/26	23	OptimalTol	3696.0	0.63	3731.61	0.96	0	OptimalTol	3683.0	2.29	3717.19	0.93	0
fri26-10-4-p	3728.0	0.003/26	22	OptimalTol	3728.0	0.98	3736.56	0.23	0	OptimalTol	3728.0	0.92	3732.87	0.13	0
fri26-10-5-p	3687.0	0.063/26	23	OptimalTol	3687.0	2.92	3720.98	0.92	0	OptimalTol	3684.0	2.59	3720.33	0.99	336
fri26-10-6-p	3731.0	0.027/26	23	OptimalTol	3731.0	0.34	3749.42	0.49	0	OptimalTol	3702.0	0.81	3734.2	0.87	0
fri26-10-7-p	3701.0	0.007/26	21	OptimalTol	3701.0	0.83	3737.86	1.0	0	OptimalTol	3676.0	6.3	3707.95	0.87	1955
fri26-10-8-p	3731.0	0.033/26	23	OptimalTol	3731.0	0.47	3766.14	0.94	0	Optimal	3731.0	0.78	3731.0	0.0	0
fri26-10-9-p	3730.0	0.007/26	23	Optimal	3730.0	0.45	3730.0	0.0	0	Optimal	3730.0	1.18	3730.0	0.0	0
fri26-10-10-p	3731.0	0.005/26	23	OptimalTol	3731.0	0.36	3767.07	0.97	0	OptimalTol	3728.0	0.49	3754.63	0.71	0
fri26-25-1-p	3679.0	0.00/26	21	OptimalTol	3679.0	1.5	3714.11	0.95	0	OptimalTol	3679.0	2.39	3715.63	1.0	205
fri26-25-2-p	3582.0	0/26	20	AbortTLim	3582.0	100.02	3628.34	1.29	28699	OptimalTol	3582.0	98.91	3617.82	1.0	31570
fri26-25-3-p	3603.0	0.016/26	21	AbortTLim	3603.0	100.01	3660.52	1.6	18937	AbortTLim	3582.0	126.01	3653.32	1.99	28173
fri26-25-4-p	3524.0	0.055/26	18	AbortTLim	3524.0	100.02	3645.15	3.44	20618	AbortTLim	3414.0	126.0	3635.66	6.49	19954
fri26-25-5-p	3707.0	0.021/26	21	OptimalTol	3707.0	0.87	3743.89	1.0	0	OptimalTol	3698.0	1.36	3730.04	0.87	0
fri26-25-6-p	3580.0	0.007/26	19	AbortTLim	3580.0	100.01	3660.88	2.26	22401	AbortTLim	3547.0	126.0	3658.62	3.15	30104
fri26-25-7-p	3610.0	0.007/26	21	AbortTLim	3610.0	100.02	3672.5	1.73	20269	AbortTLim	3595.0	126.01	3669.31	2.07	25383
fri26-25-8-p	3728.0	0.001/26	22	Optimal	3728.0	1.09	3728.0	0.0	0	OptimalTol	3711.0	1.74	3733.5	0.61	0
fri26-25-9-p	3672.0	0.021/26	22	OptimalTol	3672.0	2.6	3708.7	1.0	730	OptimalTol	3672.0	15.45	3694.89	0.62	4363
fri26-25-10-p	3700.0	0.002/26	21	OptimalTol	3700.0	0.99	3736.03	0.97	0	OptimalTol	3700.0	2.28	3724.54	0.66	0
fri26-50-1-p	3581.0	0/26	18	AbortTLim	3581.0	100.02	3646.07	1.82	19041	AbortTLim	3464.0	126.0	3650.92	5.4	28622
fri26-50-2-p	3656.0	0.004/26	21	OptimalTol	3656.0	3.71	3692.54	1.0	680	OptimalTol	3652.0	4.58	3688.52	1.0	1212
fri26-50-3-p	3712.0	0.015/26	21	OptimalTol	3712.0	0.99	3747.31	0.95	0	OptimalTol	3699.0	1.7	3726.48	0.74	0
fri26-50-4-p	3596.0	0.299/26	21	OptimalTol	3596.0	7.57	3631.94	1.0	3306	OptimalTol	3584.0	24.43	3619.84	1.0	9913
fri26-50-5-p	3537.0	0.005/26	21	AbortTLim	3537.0	100.01	3634.71	2.76	23551	AbortTLim	3365.0	126.0	3644.85	8.32	30065

Table 3: Numerical experiments on the instances of small size.

	GVNS			MIPStart CPLEX						CPLEX (no MIPStart)					
instance	Obj	Tb/Tx	Len.	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes
fri26-50-6-p	3704.0	0.007/26	22	OptimalTol	3704.0	0.85	3730.76	0.72	0	Optimal	3704.0	2.67	3704.0	0.0	0
fri26-50-7-p	3605.0	0.006/26	21	OptimalTol	3605.0	27.79	3641.0	1.0	6761	OptimalTol	3588.0	65.05	3617.44	0.82	9706
fri26-50-8-p	3529.0	0.003/26	19	AbortTLim	3529.0	100.02	3636.04	3.03	23655	AbortTLim	3462.0	126.0	3636.31	5.04	26937
fri26-50-9-p	3677.0	0.011/26	20	OptimalTol	3677.0	2.24	3713.55	0.99	0	OptimalTol	3661.0	4.41	3697.58	1.0	1245
fri26-50-10-p	3677.0	0.003/26	22	OptimalTol	3677.0	1.98	3713.75	1.0	240	OptimalTol	3677.0	6.28	3683.79	0.18	2598
gr17-10-1-p	7097.0	0.002/17	15	OptimalTol	7097.0	0.48	7163.08	0.93	0	OptimalTol	7097.0	0.68	7164.18	0.95	0
gr17-10-2-p	7055.0	0/17	15	OptimalTol	7055.0	0.13	7096.51	0.59	0	Optimal	7055.0	0.19	7055.0	0.0	0
gr17-10-3-p	7155.0	0.001/17	15	OptimalTol	7155.0	0.2	7189.92	0.49	0	OptimalTol	7155.0	0.2	7195.09	0.56	0
gr17-10-4-p	6817.0	0.001/17	14	OptimalTol	6817.0	2.32	6885.16	1.0	2178	OptimalTol	6817.0	2.3	6885.04	1.0	1883
gr17-10-5-p	7155.0	0.001/17	15	OptimalTol	7155.0	0.31	7183.16	0.39	0	OptimalTol	7102.0	0.18	7158.8	0.8	0
gr17-10-6-p	7155.0	0.006/17	15	OptimalTol	7155.0	0.31	7209.96	0.77	0	Optimal	7155.0	0.25	7155.0	0.0	0
gr17-10-7-p	7155.0	0.001/17	15	Optimal	7155.0	0.19	7155.0	0.0	0	Optimal	7155.0	0.32	7155.0	0.0	0
gr17-10-8-p	7155.0	0.001/17	15	OptimalTol	7155.0	0.11	7226.27	1.0	0	Optimal	7155.0	0.26	7155.0	0.0	0
gr17-10-9-p	7155.0	0.002/17	15	OptimalTol	7155.0	0.16	7184.86	0.42	0	OptimalTol	7155.0	0.14	7156.62	0.02	0
gr17-10-10-p	7155.0	0.002/17	15	OptimalTol	7155.0	0.15	7196.04	0.57	0	Optimal	7155.0	0.23	7155.0	0.0	0
gr17-25-1-p	7019.0	0.003/17	14	OptimalTol	7019.0	1.12	7082.67	0.91	0	OptimalTol	7019.0	1.72	7030.45	0.16	1420
gr17-25-2-p	6707.0	0.005/17	15	OptimalTol	6707.0	59.22	6774.07	1.0	50998	OptimalTol	6707.0	80.35	6774.07	1.0	76605
gr17-25-3-p	7025.0	0.002/17	14	Optimal	7025.0	0.32	7025.0	0.0	0	OptimalTol	7025.0	0.19	7056.67	0.45	0
gr17-25-4-p	7090.0	0.005/17	15	OptimalTol	7090.0	0.24	7156.68	0.94	0	OptimalTol	7086.0	0.77	7126.31	0.57	0
gr17-25-5-p	7155.0	0.002/17	15	OptimalTol	7155.0	0.21	7214.66	0.83	0	OptimalTol	7130.0	0.21	7188.51	0.82	0
gr17-25-6-p	7090.0	0.002/17	15	OptimalTol	7090.0	0.39	7156.97	0.94	0	OptimalTol	7090.0	1.08	7129.39	0.56	173
gr17-25-7-p	7155.0	0.003/17	15	OptimalTol	7155.0	0.08	7171.18	0.23	0	OptimalTol	7155.0	0.22	7156.54	0.02	0
gr17-25-8-p	7152.0	0.002/17	15	OptimalTol	7152.0	0.12	7180.69	0.4	0	Optimal	7152.0	0.2	7152.0	0.0	0
gr17-25-9-p	7099.0	0.001/17	15	OptimalTol	7099.0	0.18	7168.87	0.98	0	OptimalTol	7098.0	0.18	7166.57	0.97	0
gr17-25-10-p	6605.0	0.004/17	13	AbortTLim	6605.0	100.02	6721.3	1.76	86378	AbortTLim	6605.0	117.02	6692.17	1.32	129879
gr17-50-1-p	7025.0	0.001/17	14	OptimalTol	7025.0	0.93	7079.41	0.77	0	OptimalTol	7025.0	1.07	7092.41	0.96	205
gr17-50-2-p	6997.0	0.002/17	15	OptimalTol	6997.0	0.8	7055.42	0.83	0	OptimalTol	6996.0	1.26	7035.22	0.56	507
gr17-50-3-p	6545.0	0.025/17	14	AbortTLim	6545.0	100.02	6728.43	2.8	74471	AbortTLim	6504.0	117.01	6739.96	3.63	75409
gr17-50-4-p	6476.0	0.001/17	13	OptimalTol	6476.0	36.29	6540.75	1.0	37958	OptimalTol	6476.0	29.78	6523.09	0.73	29378
gr17-50-5-p	7099.0	0.001/17	15	Optimal	7099.0	0.18	7099.0	0.0	0	OptimalTol	7099.0	0.59	7134.13	0.49	0
gr17-50-6-p	6939.0	0.005/17	14	OptimalTol	6939.0	1.55	7008.37	1.0	774	OptimalTol	6938.0	1.82	7007.31	1.0	693
gr17-50-7-p	6570.0	0.001/17	13	OptimalTol	6570.0	47.0	6635.7	1.0	46838	AbortTLim	6482.0	117.0	6589.22	1.65	143440
gr17-50-8-p	6478.0	0.001/17	12	AbortTLim	6478.0	100.02	6569.64	1.41	84373	OptimalTol	6478.0	104.49	6542.78	1.0	80491
gr17-50-9-p	6849.0	0.001/17	14	OptimalTol	6849.0	4.67	6917.49	1.0	3072	OptimalTol	6849.0	9.38	6858.36	0.14	6696
gr17-50-10-p	7154.0	0.001/17	14	Optimal	7154.0	0.34	7154.0	0.0	0	OptimalTol	7154.0	0.49	7173.94	0.28	0
gr21-10-1-p	9884.0	0.001/21	17	Optimal	9884.0	0.14	9884.0	0.0	0	Optimal	9884.0	0.2	9884.0	0.0	0
gr21-10-2-p	9594.0	0/21	17	OptimalTol	9594.0	1.57	9689.39	0.99	388	OptimalTol	9584.0	2.75	9649.48	0.68	947
gr21-10-3-p	9815.0	0.002/21	17	OptimalTol	9815.0	0.16	9877.75	0.64	0	OptimalTol	9789.0	0.29	9880.54	0.94	0
gr21-10-4-p	9882.0	0.007/21	17	Optimal	9882.0	0.18	9882.0	0.0	0	Optimal	9882.0	0.34	9882.0	0.0	0
gr21-10-5-p	9884.0	0.001/21	17	Optimal	9884.0	0.15	9884.0	0.0	0	OptimalTol	9884.0	0.26	9934.32	0.51	0
gr21-10-6-p	9882.0	0.002/21	17	Optimal	9882.0	0.15	9882.0	0.0	0	Optimal	9882.0	0.34	9882.0	0.0	0
gr21-10-7-p	9765.0	0.001/21	16	Optimal	9765.0	0.24	9765.0	0.0	0	OptimalTol	9765.0	0.46	9845.0	0.82	0
gr21-10-8-p	9871.0	0.001/21	17	Optimal	9871.0	0.15	9871.0	0.0	0	Optimal	9871.0	0.44	9871.0	0.0	0

Table 3: Numerical experiments on the instances of small size.

	GVNS			MIPStart CPLEX						CPLEX (no MIPStart)					
instance	Obj	Tb/Tx	Len.	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes
gr21-10-9-p	9804.0	0.001/21	17	Optimal	9804.0	0.13	9804.0	0.0	0	OptimalTol	9789.0	0.2	9826.2	0.38	0
gr21-10-10-p	9884.0	0.002/21	17	Optimal	9884.0	0.18	9884.0	0.0	0	Optimal	9884.0	0.29	9884.0	0.0	0
gr21-25-1-p	9884.0	0.001/21	17	Optimal	9884.0	0.16	9884.0	0.0	0	OptimalTol	9849.0	0.2	9907.03	0.59	0
gr21-25-2-p	9790.0	0.002/21	17	OptimalTol	9790.0	0.16	9886.53	0.99	0	OptimalTol	9790.0	0.88	9847.92	0.59	0
gr21-25-3-p	9802.0	0.002/21	17	OptimalTol	9802.0	0.2	9897.21	0.97	0	Optimal	9802.0	0.77	9802.0	0.0	0
gr21-25-4-p	9884.0	0.002/21	17	Optimal	9884.0	0.17	9884.0	0.0	0	Optimal	9884.0	0.16	9884.0	0.0	0
gr21-25-5-p	9745.0	0.001/21	17	Optimal	9745.0	0.38	9745.0	0.0	0	OptimalTol	9720.0	1.17	9809.83	0.92	0
gr21-25-6-p	9520.0	0.004/21	17	OptimalTol	9520.0	0.84	9566.55	0.49	0	OptimalTol	9520.0	2.23	9543.96	0.25	0
gr21-25-7-p	9790.0	0.004/21	17	OptimalTol	9790.0	0.22	9847.07	0.58	0	OptimalTol	9790.0	0.24	9843.79	0.55	0
gr21-25-8-p	9658.0	0/21	16	Optimal	9658.0	0.66	9658.0	0.0	0	OptimalTol	9658.0	1.64	9670.63	0.13	0
gr21-25-9-p	9882.0	0.002/21	17	Optimal	9882.0	0.12	9882.0	0.0	0	OptimalTol	9882.0	0.19	9904.93	0.23	0
gr21-25-10-p	9884.0	0.002/21	17	OptimalTol	9884.0	0.21	9891.1	0.07	0	Optimal	9884.0	0.31	9884.0	0.0	0
gr21-50-1-p	9474.0	0.01/21	18	OptimalTol	9474.0	21.8	9568.74	1.0	10071	OptimalTol	9474.0	38.0	9525.48	0.54	18837
gr21-50-2-p	9336.0	0.004/21	18	OptimalTol	9336.0	15.92	9429.35	1.0	6159	OptimalTol	9336.0	44.88	9340.66	0.05	16673
gr21-50-3-p	9702.0	0.001/21	17	Optimal	9702.0	0.4	9702.0	0.0	0	OptimalTol	9687.0	1.81	9754.09	0.69	0
gr21-50-4-p	9394.0	0.002/21	16	OptimalTol	9394.0	14.44	9487.92	1.0	10381	OptimalTol	9387.0	19.59	9480.86	1.0	14047
gr21-50-5-p	9128.0	0.006/21	17	OptimalTol	9128.0	5.68	9219.26	1.0	4495	OptimalTol	9128.0	11.37	9219.24	1.0	7391
gr21-50-6-p	9613.0	0.003/21	17	OptimalTol	9613.0	1.14	9708.69	1.0	311	OptimalTol	9613.0	2.62	9696.0	0.86	507
gr21-50-7-p	8846.0	0.006/21	18	AbortTLim	8846.0	100.02	9248.75	4.55	33263	AbortTLim	8814.0	121.01	9247.99	4.92	49882
gr21-50-8-p	9242.0	0.015/21	17	OptimalTol	9242.0	29.06	9334.42	1.0	16886	OptimalTol	9198.0	50.04	9289.98	1.0	21340
gr21-50-9-p	9321.0	0.064/21	18	OptimalTol	9321.0	4.06	9414.08	1.0	2060	OptimalTol	9320.0	5.53	9413.15	1.0	1855
gr21-50-10-p	9495.0	0.002/21	18	OptimalTol	9495.0	1.89	9589.57	1.0	0	OptimalTol	9495.0	3.2	9548.06	0.56	821
gr48-10-1-p	32248.0	0.088/48	43	OptimalTol	32248.0	0.69	32322.8	0.23	0	OptimalTol	32248.0	2.51	32318.8	0.22	0
gr48-10-2-p	31853.0	0.016/48	43	OptimalTol	31853.0	11.05	32171.4	1.0	544	AbortTLim	31648.0	148.05	31978.2	1.04	4909
gr48-10-3-p	31827.0	7.604/48	42	OptimalTol	31827.0	40.51	32145.2	1.0	3298	AbortTLim	31508.0	148.04	32139.1	2.0	10232
gr48-10-4-p	31899.0	0.028/48	43	OptimalTol	31899.0	9.24	32218.0	1.0	666	AbortTLim	31458.0	148.05	32131.9	2.14	9281
gr48-10-5-p	31779.0	0.169/48	42	OptimalTol	31779.0	59.8	32096.0	1.0	2026	AbortTLim	31179.0	148.02	32007.0	2.66	6984
gr48-10-6-p	31769.0	0.054/48	43	OptimalTol	31769.0	52.67	32086.1	1.0	2296	OptimalTol	31769.0	123.18	32086.4	1.0	3758
gr48-10-7-p	32097.0	0.056/48	43	OptimalTol	32097.0	1.49	32394.4	0.93	0	OptimalTol	32023.0	5.85	32279.1	0.8	86
gr48-10-8-p	32191.0	0.064/48	43	OptimalTol	32191.0	0.64	32448.2	0.8	0	OptimalTol	32168.0	28.81	32217.8	0.15	1492
gr48-10-9-p	32097.0	0.053/48	43	OptimalTol	32097.0	1.84	32385.2	0.9	0	OptimalTol	32058.0	58.1	32213.8	0.49	2988
gr48-10-10-p	32121.0	0.077/48	44	OptimalTol	32121.0	0.94	32440.1	0.99	0	OptimalTol	31977.0	17.99	32296.7	1.0	1334
gr48-25-1-p	31736.0	0.119/48	44	OptimalTol	31736.0	95.86	32053.3	1.0	3338	AbortTLim	31408.0	148.04	32050.4	2.05	5805
gr48-25-2-p	31788.0	0.032/48	42	OptimalTol	31788.0	37.44	32105.5	1.0	2305	AbortTLim	30729.0	148.05	32036.4	4.25	6435
gr48-25-3-p	31601.0	0.024/48	41	AbortTLim	31601.0	100.03	31934.5	1.06	4147	AbortTLim	30745.0	148.03	31944.2	3.9	10551
gr48-25-4-p	31604.0	20.844/48	43	AbortTLim	31604.0	100.04	32096.6	1.56	4416	AbortTLim	27770.0	148.04	32116.5	15.65	14710
gr48-25-5-p	31155.0	2.407/48	42	AbortTLim	31155.0	100.02	32005.1	2.73	6360	AbortTLim	28045.0	148.02	32029.5	14.21	12637
gr48-25-6-p	31643.0	0.217/48	42	AbortTLim	31643.0	100.04	32143.7	1.58	5835	AbortTLim	29617.0	148.03	32115.0	8.43	11971
gr48-25-7-p	31396.0	3.526/48	42	AbortTLim	31396.0	100.03	32004.8	1.94	3645	AbortTLim	26450.0	148.02	32051.3	21.18	9234
gr48-25-8-p	31628.0	0.035/48	43	AbortTLim	31628.0	100.02	32082.6	1.44	5682	AbortTLim	30064.0	148.05	32046.0	6.59	11811
gr48-25-9-p	31547.0	0.072/48	41	AbortTLim	31547.0	100.04	32081.5	1.69	4530	AbortTLim	28056.0	148.05	32120.1	14.49	13308
gr48-25-10-p	31632.0	0.383/48	41	AbortTLim	31632.0	100.01	32194.3	1.78	2950	AbortTLim	31032.0	148.06	32046.8	3.27	8615

Table 3: Numerical experiments on the instances of small size.

	GVNS			MIPStart CPLEX						CPLEX (no MIPStart)					
instance	Obj	Tb/Tx	Len.	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes
gr48-50-1-p	31470.0	0.034/48	40	AbortTLim	31470.0	100.03	32154.3	2.17	4814	AbortTLim	28088.0	148.06	32092.4	14.26	16067
gr48-50-2-p	30985.0	0.386/48	41	AbortTLim	30985.0	100.06	31950.5	3.12	5470	AbortTLim	29054.0	148.03	31945.3	9.95	7781
gr48-50-3-p	31957.0	0.02/48	41	OptimalTol	31957.0	2.03	32256.7	0.94	0	OptimalTol	31957.0	12.8	32216.3	0.81	1254
gr48-50-4-p	30952.0	0.03/48	40	AbortTLim	30952.0	100.02	31976.4	3.31	8490	AbortTLim	29544.0	148.03	31997.5	8.3	9816
gr48-50-5-p	30577.0	0.197/48	43	AbortTLim	30577.0	100.04	31885.6	4.28	6412	AbortTLim	21500.0	148.04	31935.4	48.54	12556
gr48-50-6-p	30268.0	6.217/48	39	AbortTLim	30268.0	100.04	31736.7	4.85	7512	AbortTLim	20421.0	148.03	31771.7	55.58	16012
gr48-50-7-p	31395.0	0.833/48	40	AbortTLim	31395.0	100.03	32092.8	2.22	5005	AbortTLim	27961.0	148.02	32063.5	14.67	15759
gr48-50-8-p	31310.0	0.262/48	41	AbortTLim	31310.0	100.02	32018.2	2.26	5557	AbortTLim	28337.0	148.04	32047.5	13.09	14845
gr48-50-9-p	30489.0	1.059/48	40	AbortTLim	30489.0	100.02	32021.4	5.03	6909	AbortTLim	25242.0	148.04	32020.9	26.86	12348
gr48-50-10-p	31111.0	7.623/48	42	AbortTLim	31111.0	100.02	31968.8	2.76	6073	AbortTLim	28100.0	148.02	32006.6	13.9	10249
ulysses16-10-1-p	24658.0	0.002/16	15	OptimalTol	24658.0	0.06	24900.7	0.98	0	OptimalTol	24647.0	0.1	24697.4	0.2	0
ulysses16-10-2-p	24658.0	0.001/16	15	OptimalTol	24658.0	0.16	24865.5	0.84	0	OptimalTol	24595.0	0.29	24824.7	0.93	0
ulysses16-10-3-p	24658.0	0.005/16	15	OptimalTol	24658.0	0.23	24888.1	0.93	0	Optimal	24658.0	0.27	24658.0	0.0	0
ulysses16-10-4-p	24658.0	0.004/16	15	OptimalTol	24658.0	0.14	24852.6	0.79	0	Optimal	24658.0	0.25	24658.0	0.0	0
ulysses16-10-5-p	24658.0	0.004/16	15	OptimalTol	24658.0	0.05	24896.8	0.97	0	Optimal	24658.0	0.18	24658.0	0.0	0
ulysses16-10-6-p	24621.0	0.001/16	14	OptimalTol	24621.0	0.09	24662.5	0.17	0	OptimalTol	24621.0	0.3	24622.2	0.0	0
ulysses16-10-7-p	24658.0	0.002/16	15	OptimalTol	24658.0	0.1	24884.4	0.92	0	OptimalTol	24619.0	1.46	24850.6	0.94	0
ulysses16-10-8-p	24658.0	0.004/16	15	OptimalTol	24658.0	0.08	24904.5	1.0	0	OptimalTol	24544.0	0.38	24667.1	0.5	0
ulysses16-10-9-p	24658.0	0.002/16	15	OptimalTol	24658.0	0.12	24851.0	0.78	0	OptimalTol	24594.0	0.3	24832.1	0.97	0
ulysses16-10-10-p	24658.0	0.003/16	15	OptimalTol	24658.0	0.2	24840.3	0.74	0	Optimal	24658.0	0.45	24658.0	0.0	0
ulysses16-25-1-p	24652.0	0.001/16	14	OptimalTol	24652.0	0.07	24891.0	0.97	0	OptimalTol	24596.0	0.39	24819.2	0.91	0
ulysses16-25-2-p	24658.0	0.003/16	15	OptimalTol	24658.0	0.13	24821.1	0.66	0	Optimal	24658.0	0.3	24658.0	0.0	0
ulysses16-25-3-p	24658.0	0.001/16	15	OptimalTol	24658.0	0.13	24771.7	0.46	0	Optimal	24658.0	0.45	24658.0	0.0	0
ulysses16-25-4-p	24116.0	0.006/16	15	OptimalTol	24116.0	1.84	24357.1	1.0	599	OptimalTol	24074.0	2.49	24183.5	0.45	1461
ulysses16-25-5-p	24652.0	0.00/16	14	OptimalTol	24652.0	0.14	24878.9	0.92	0	OptimalTol	24602.0	0.4	24683.2	0.33	0
ulysses16-25-6-p	24488.0	0.003/16	15	OptimalTol	24488.0	0.37	24681.6	0.79	0	OptimalTol	24393.0	0.56	24634.0	0.99	0
ulysses16-25-7-p	24071.0	0.019/16	15	OptimalTol	24071.0	1.62	24309.8	0.99	37	OptimalTol	24071.0	1.32	24171.5	0.42	257
ulysses16-25-8-p	24658.0	0.004/16	15	OptimalTol	24658.0	0.14	24870.2	0.86	0	OptimalTol	24491.0	0.17	24672.7	0.74	0
ulysses16-25-9-p	24658.0	0.003/16	15	OptimalTol	24658.0	0.14	24841.9	0.75	0	OptimalTol	24658.0	0.22	24773.1	0.47	0
ulysses16-25-10-p	24193.0	0.00/16	14	OptimalTol	24193.0	1.9	24434.9	1.0	1301	OptimalTol	24193.0	1.67	24434.9	1.0	741
ulysses16-50-1-p	24253.0	0.003/16	15	OptimalTol	24253.0	0.89	24313.7	0.25	0	OptimalTol	24253.0	1.8	24494.7	1.0	432
ulysses16-50-2-p	23873.0	0.004/16	14	OptimalTol	23873.0	1.07	24111.6	1.0	1649	OptimalTol	23851.0	1.15	24089.4	1.0	1529
ulysses16-50-3-p	22883.0	0.004/16	13	AbortTLim	22883.0	100.02	23531.7	2.83	83489	AbortTLim	22804.0	116.01	23576.6	3.39	94320
ulysses16-50-4-p	24481.0	0.004/16	14	Optimal	24481.0	0.24	24481.0	0.0	0	OptimalTol	24481.0	0.58	24482.1	0.0	0
ulysses16-50-5-p	24652.0	0.001/16	14	OptimalTol	24652.0	0.11	24884.0	0.94	0	OptimalTol	24562.0	0.17	24784.0	0.9	0
ulysses16-50-6-p	24616.0	0.001/16	14	OptimalTol	24616.0	0.07	24855.6	0.97	0	OptimalTol	24616.0	0.23	24617.0	0.0	0
ulysses16-50-7-p	24254.0	0.002/16	14	OptimalTol	24254.0	1.2	24496.2	1.0	176	OptimalTol	24194.0	1.63	24435.9	1.0	310
ulysses16-50-8-p	24572.0	0.00/16	14	OptimalTol	24572.0	0.04	24632.1	0.24	0	Optimal	24572.0	0.4	24572.0	0.0	0
ulysses16-50-9-p	24617.0	0.001/16	15	OptimalTol	24617.0	0.06	24852.4	0.96	0	OptimalTol	24617.0	0.42	24806.6	0.77	0
ulysses16-50-10-p	23811.0	0.008/16	15	OptimalTol	23811.0	0.06	23978.2	0.7	0	OptimalTol	23811.0	0.25	23813.1	0.01	0
ulysses22-10-1-p	32323.0	0.012/22	21	OptimalTol	32323.0	0.23	32584.0	0.81	0	OptimalTol	32153.0	0.79	32323.5	0.53	0
ulysses22-10-2-p	32323.0	0.003/22	21	OptimalTol	32323.0	0.35	32609.2	0.89	0	OptimalTol	32323.0	0.26	32561.7	0.74	0

Table 3: Numerical experiments on the instances of small size.

	GVNS			MIPStart CPLEX						CPLEX (no MIPStart)					
instance	Obj	Tb/Tx	Len.	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes
ulysses22-10-3-p	32323.0	0.003/22	21	OptimalTol	32323.0	0.24	32643.5	0.99	0	Optimal	32323.0	0.47	32323.0	0.0	0
ulysses22-10-4-p	32323.0	0.001/22	21	OptimalTol	32323.0	0.19	32611.1	0.89	0	OptimalTol	32162.0	0.61	32459.1	0.92	0
ulysses22-10-5-p	32323.0	0.002/22	21	OptimalTol	32323.0	0.34	32582.3	0.8	0	Optimal	32323.0	0.6	32323.0	0.0	0
ulysses22-10-6-p	32317.0	0.002/22	20	OptimalTol	32317.0	0.25	32554.7	0.74	0	OptimalTol	32247.0	0.32	32353.8	0.33	0
ulysses22-10-7-p	32306.0	0.00/22	19	OptimalTol	32306.0	0.84	32309.8	0.01	0	Optimal	32306.0	1.07	32306.0	0.0	0
ulysses22-10-8-p	32155.0	0.003/22	21	OptimalTol	32155.0	0.3	32472.1	0.99	0	OptimalTol	32085.0	0.82	32239.2	0.48	0
ulysses22-10-9-p	32289.0	0.003/22	21	OptimalTol	32289.0	0.14	32601.6	0.97	0	Optimal	32289.0	0.48	32289.0	0.0	0
ulysses22-10-10-p	32249.0	0.004/22	21	OptimalTol	32249.0	0.51	32396.8	0.46	0	OptimalTol	32187.0	0.62	32499.4	0.97	0
ulysses22-25-1-p	32317.0	0.002/22	20	OptimalTol	32317.0	0.18	32527.1	0.65	0	OptimalTol	32206.0	0.95	32465.9	0.81	0
ulysses22-25-2-p	32202.0	0/22	19	OptimalTol	32202.0	0.4	32493.9	0.91	0	OptimalTol	32124.0	1.47	32295.4	0.53	0
ulysses22-25-3-p	31577.0	0.001/22	18	OptimalTol	31577.0	1.97	31797.0	0.7	0	OptimalTol	31551.0	3.19	31863.6	0.99	232
ulysses22-25-4-p	31951.0	0.006/22	21	OptimalTol	31951.0	1.4	32269.8	1.0	170	OptimalTol	31858.0	5.31	32170.4	0.98	2346
ulysses22-25-5-p	31887.0	0.029/22	21	OptimalTol	31887.0	2.27	32205.5	1.0	493	OptimalTol	31739.0	18.92	32056.4	1.0	6429
ulysses22-25-6-p	32051.0	0.002/22	19	OptimalTol	32051.0	0.47	32320.6	0.84	0	OptimalTol	31839.0	5.52	32153.7	0.99	1559
ulysses22-25-7-p	31607.0	0.091/22	21	AbortTLim	31607.0	100.01	31945.8	1.07	38787	AbortTLim	31517.0	122.0	31908.7	1.24	40029
ulysses22-25-8-p	32261.0	0.001/22	19	OptimalTol	32261.0	0.25	32582.3	1.0	0	OptimalTol	32249.0	1.14	32317.7	0.21	0
ulysses22-25-9-p	32306.0	0.00/22	19	OptimalTol	32306.0	0.34	32523.4	0.91	0	OptimalTol	32170.0	0.64	32435.6	0.83	0
ulysses22-25-10-p	31375.0	0.006/22	21	OptimalTol	31375.0	0.55	31646.7	0.87	0	OptimalTol	31292.0	1.3	31602.7	0.99	432
ulysses22-50-1-p	31495.0	0.082/22	19	OptimalTol	31495.0	0.73	31788.6	0.93	0	OptimalTol	31495.0	1.96	31809.5	1.0	309
ulysses22-50-2-p	32076.0	0.006/22	19	OptimalTol	32076.0	0.63	32340.4	0.82	0	OptimalTol	32068.0	6.47	32139.6	0.22	2736
ulysses22-50-3-p	31836.0	0.009/22	20	OptimalTol	31836.0	2.41	32152.3	0.99	245	OptimalTol	31836.0	5.05	31940.2	0.33	1491
ulysses22-50-4-p	31358.0	0.008/22	18	AbortTLim	31358.0	100.01	31715.3	1.14	25927	AbortTLim	30203.0	122.01	31724.4	5.04	29111
ulysses22-50-5-p	31860.0	0.002/22	19	OptimalTol	31860.0	1.01	32133.0	0.86	0	OptimalTol	31785.0	2.17	32098.5	0.99	0
ulysses22-50-6-p	32173.0	0/22	19	OptimalTol	32173.0	1.27	32358.0	0.58	0	OptimalTol	32120.0	1.12	32282.2	0.51	0
ulysses22-50-7-p	32134.0	0.004/22	19	OptimalTol	32134.0	0.71	32444.3	0.97	0	OptimalTol	32006.0	1.37	32292.7	0.9	0
ulysses22-50-8-p	31504.0	0.002/22	19	OptimalTol	31504.0	11.97	31818.8	1.0	3158	OptimalTol	31379.0	75.51	31657.4	0.89	30019
ulysses22-50-9-p	31053.0	0.016/22	20	AbortTLim	31053.0	100.01	31499.8	1.44	33928	AbortTLim	31053.0	122.01	31581.2	1.7	37011
ulysses22-50-10-p	32060.0	0.003/22	20	OptimalTol	32060.0	1.16	32287.7	0.71	0	OptimalTol	32060.0	2.99	32142.8	0.26	571

Table 4: Numerical experiments on the instances of larger size.

	GVNS			MIPStart CPLEX						CPLEX (no MIPStart)					
instance	Obj	Tb/Tx	Len.	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes
kroA100-50-1-p	300091.0	89.964/100	95	AbortTLim	300091.0	100.05	303983.8	1.0	362	AbortTLim	173810.0	200.03	303678.0	74.72	1528
kroA100-50-2-p	300311.0	17.435/100	93	AbortTLim	300311.0	100.03	304075.61	1.0	968	AbortTLim	152679.0	200.05	303435.0	98.74	1511
kroA100-50-3-p	301564.0	2.208/100	92	OptimalTol	301564.0	9.84	304515.46	1.0	0	AbortTLim	242836.0	200.05	303902.0	25.15	1569
kroA100-50-4-p	299565.0	22.494/100	92	AbortTLim	299565.0	100.07	304129.11	2.0	1547	AbortTLim	253819.0	200.04	303468.0	19.56	1515
kroA100-50-5-p	301002.0	5.324/100	93	AbortTLim	301002.0	100.04	304137.96	1.0	900	AbortTLim	220889.0	200.04	303629.0	37.46	1440
kroA100-50-6-p	300756.0	0.759/100	94	AbortTLim	300756.0	100.05	304355.19	1.0	968	AbortTLim	151277.0	200.33	303596.0	100.69	1513
kroA100-50-7-p	299502.0	34.446/100	95	AbortTLim	299502.0	100.03	304025.62	2.0	651	AbortTLim	172434.0	200.47	303583.0	76.06	1489
kroA100-50-8-p	301413.0	53.982/100	94	OptimalTol	301413.0	9.7	304389.14	1.0	0	AbortTLim	215550.0	200.23	303593.0	40.85	1621
kroA100-50-9-p	299655.0	3.088/100	94	AbortTLim	299655.0	100.04	304082.81	1.0	1156	AbortTLim	145233.0	200.14	303549.0	109.01	1630
kroA100-50-10-p	301326.0	20.569/100	96	OptimalTol	301326.0	11.35	304253.21	1.0	0	AbortTLim	277941.0	200.08	303542.0	9.21	1430
kroA100-75-1-p	298384.0	59.578/100	93	AbortTLim	298384.0	100.11	304341.68	2.0	846	AbortTLim	150785.0	200.07	303748.0	101.44	1615
kroA100-75-2-p	299519.0	2.079/100	91	AbortTLim	299519.0	100.05	304277.31	2.0	1038	AbortTLim	237952.0	200.08	303929.0	27.73	1595
kroA100-75-3-p	299627.0	29.806/100	92	AbortTLim	299627.0	100.04	304096.73	1.0	400	AbortTLim	223042.0	200.07	303542.0	36.09	1442
kroA100-75-4-p	299576.0	29.139/100	91	AbortTLim	299576.0	100.05	304009.01	1.0	886	AbortTLim	210293.0	200.07	303468.0	44.31	1429
kroA100-75-5-p	299055.0	80.823/100	94	AbortTLim	299055.0	100.05	304080.53	2.0	496	AbortTLim	171504.0	200.06	303765.0	77.12	1638
kroA100-75-6-p	297155.0	13.819/100	90	AbortTLim	297155.0	100.05	304131.21	2.0	839	AbortTLim	141955.0	200.07	303586.0	113.86	1627
kroA100-75-7-p	296782.0	10.774/100	93	AbortTLim	296782.0	100.03	303954.95	2.0	538	AbortTLim	218360.0	200.07	303604.0	39.04	1525
kroA100-75-8-p	299713.0	6.897/100	92	AbortTLim	299713.0	100.06	304128.86	1.0	971	AbortTLim	244850.0	200.08	303610.0	24.0	1420
kroA100-75-9-p	298140.0	81.351/100	91	AbortTLim	298140.0	100.06	304013.22	2.0	364	AbortTLim	155254.0	200.07	303545.0	95.51	1553
kroA100-75-10-p	300402.0	1.268/100	91	AbortTLim	300402.0	100.03	303979.19	1.0	591	AbortTLim	132891.0	200.1	304034.0	128.78	1143
kroA200-50-1-p	628446.0	119.568/200	194	AbortTLim	628446.0	100.24	636075.36	1.0	0	AbortTLim	266933.0	300.13	635604.0	138.11	21
kroA200-50-2-p	630584.0	102.224/200	190	OptimalTol	630584.0	40.49	636542.82	1.0	0	AbortTLim	406665.0	300.15	635837.0	56.35	17
kroA200-50-3-p	629306.0	194.809/200	190	AbortTLim	629332.0	100.06	636191.5	1.0	0	AbortTLim	307558.0	300.15	635665.0	106.68	12
kroA200-50-4-p	630582.0	168.231/200	188	OptimalTol	630582.0	51.88	636642.86	1.0	0	AbortTLim	316435.0	300.34	635668.0	100.88	23
kroA200-50-5-p	629122.0	127.757/200	190	AbortTLim	629122.0	100.05	635888.88	1.0	0	AbortTLim	398445.0	300.18	635769.0	59.56	8
kroA200-50-6-p	628744.0	189.357/200	192	AbortTLim	628744.0	100.04	636018.53	1.0	0	AbortTLim	404910.0	300.13	635845.0	57.03	17
kroA200-50-7-p	629862.0	135.355/200	189	OptimalTol	629862.0	79.24	636044.01	1.0	0	AbortTLim	141577.0	300.14	635894.0	349.15	30
kroA200-50-8-p	630455.0	23.748/200	190	OptimalTol	630455.0	56.27	636412.29	1.0	0	AbortTLim	378122.0	300.16	635785.0	68.14	12
kroA200-50-9-p	628984.0	188.109/200	189	AbortTLim	628984.0	100.05	635999.64	1.0	0	AbortTLim	283518.0	300.1	635663.0	124.21	6
kroA200-50-10-p	630156.0	195.409/200	193	OptimalTol	630156.0	74.76	636228.95	1.0	0	AbortTLim	335836.0	300.12	635731.0	89.3	12
kroA200-75-1-p	626107.0	127.017/200	188	AbortTLim	626107.0	100.05	635943.51	2.0	0	AbortTLim	318945.0	300.16	635758.0	99.33	48
kroA200-75-2-p	625195.0	91.166/200	188	AbortTLim	625195.0	100.05	635743.19	2.0	0	AbortTLim	298944.0	300.14	635695.0	112.65	27
kroA200-75-3-p	625222.0	74.885/200	184	AbortTLim	625222.0	100.06	635794.48	2.0	0	AbortTLim	272163.0	301.61	635749.0	133.59	25
kroA200-75-4-p	627291.0	67.268/200	187	AbortTLim	627291.0	100.04	635780.63	1.0	0	AbortTLim	260443.0	300.13	635591.0	144.04	81
kroA200-75-5-p	624344.0	140.857/200	185	AbortTLim	624344.0	100.04	635961.54	2.0	0	AbortTLim	264207.0	300.84	635712.0	140.61	17
kroA200-75-6-p	625778.0	193.827/200	188	AbortTLim	625778.0	100.05	636094.13	2.0	0	AbortTLim	400391.0	300.1	635724.0	58.78	5
kroA200-75-7-p	625069.0	184.619/200	187	AbortTLim	625069.0	100.04	635744.49	2.0	0	AbortTLim	322135.0	300.12	635694.0	97.34	64
kroA200-75-8-p	625523.0	117.291/200	187	AbortTLim	625523.0	100.05	635704.21	2.0	0	AbortTLim	215676.0	300.14	635576.0	194.69	63
kroA200-75-9-p	626635.0	198.447/200	190	AbortTLim	626635.0	100.06	636002.83	1.0	0	AbortTLim	293237.0	300.13	635537.0	116.73	12
kroA200-75-10-p	625195.0	105.975/200	189	AbortTLim	625195.0	100.04	635897.38	2.0	0	AbortTLim	386889.0	300.13	635790.0	64.33	25
pcb442-50-1-p	1474919.0	381.07/442	421	OptimalTol	1474919.0	49.95	1486473.64	1.0	0	AbortTLim	0.0	542.2	1486470.0	N.A.	0
pcb442-50-2-p	1475162.0	411.182/442	423	OptimalTol	1475162.0	45.91	1486479.88	1.0	0	AbortTLim	0.0	542.24	1486480.0	N.A.	0

Table 4: Numerical experiments on the instances of larger size.

	GVNS			MIPStart CPLEX						CPLEX (no MIPStart)					
instance	Obj	Tb/Tx	Len.	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes	CPX.Sts	Obj	T (sec.)	UB	Gap	#Nodes
pcb442-50-3-p	1477328.0	415.848/442	425	OptimalTol	1477328.0	53.0	1486476.22	1.0	0	AbortTLim	0.0	542.18	1484330.0	N.A.	0
pcb442-50-4-p	1472737.0	439.607/442	420	OptimalTol	1472737.0	50.84	1486447.05	1.0	0	AbortTLim	0.0	542.22	1486430.0	N.A.	0
pcb442-50-5-p	1473014.0	401.369/442	422	OptimalTol	1473014.0	54.52	1486473.56	1.0	0	AbortTLim	0.0	542.2	1486460.0	N.A.	0
pcb442-50-6-p	1473685.0	370.391/442	424	OptimalTol	1473685.0	51.32	1486486.62	1.0	0	AbortTLim	0.0	542.19	1486490.0	N.A.	0
pcb442-50-7-p	1474547.0	426.135/442	423	OptimalTol	1474547.0	51.82	1486465.84	1.0	0	AbortTLim	0.0	542.18	1484550.0	N.A.	0
pcb442-50-8-p	1473209.0	417.906/442	422	OptimalTol	1473209.0	51.51	1486429.59	1.0	0	AbortTLim	0.0	542.24	1484400.0	N.A.	0
pcb442-50-9-p	1473955.0	432.224/442	423	OptimalTol	1473955.0	52.15	1486475.86	1.0	0	AbortTLim	0.0	542.2	1486480.0	N.A.	0
pcb442-50-10-p	1472780.0	414.249/442	420	OptimalTol	1472780.0	48.31	1486479.75	1.0	0	AbortTLim	0.0	542.21	1486450.0	N.A.	0
pcb442-75-1-p	1473784.0	420.251/442	420	OptimalTol	1473784.0	49.94	1486428.06	1.0	0	AbortTLim	0.0	542.2	1486430.0	N.A.	0
pcb442-75-2-p	1470463.0	369.161/442	416	AbortTLim	1470463.0	100.17	1486388.48	1.0	0	AbortTLim	0.0	542.17	1484270.0	N.A.	0
pcb442-75-3-p	1471305.0	408.013/442	420	AbortTLim	1471305.0	100.16	1486450.01	1.0	0	AbortTLim	0.0	542.19	1486450.0	N.A.	0
pcb442-75-4-p	1472696.0	359.303/442	417	OptimalTol	1472696.0	55.91	1486410.68	1.0	0	AbortTLim	0.0	542.19	1486390.0	N.A.	0
pcb442-75-5-p	1468677.0	419.85/442	413	AbortTLim	1468677.0	100.16	1486452.46	1.0	0	AbortTLim	0.0	542.23	1484480.0	N.A.	0
pcb442-75-6-p	1468561.0	416.17/442	416	AbortTLim	1468561.0	100.18	1486458.02	1.0	0	AbortTLim	0.0	542.19	1484270.0	N.A.	0
pcb442-75-7-p	1470779.0	316.433/442	419	AbortTLim	1470779.0	100.16	1486436.81	1.0	0	AbortTLim	0.0	542.18	1484340.0	N.A.	0
pcb442-75-8-p	1468757.0	421.304/442	418	AbortTLim	1468757.0	100.18	1486421.78	1.0	0	AbortTLim	0.0	542.2	1486420.0	N.A.	0
pcb442-75-9-p	1466470.0	422.058/442	417	AbortTLim	1466470.0	100.16	1486392.4	1.0	0	AbortTLim	0.0	542.2	1486390.0	N.A.	0
pcb442-75-10-p	1471847.0	437.265/442	417	OptimalTol	1471847.0	47.72	1486422.81	1.0	0	AbortTLim	0.0	542.2	1486420.0	N.A.	0