Technical University of Denmark

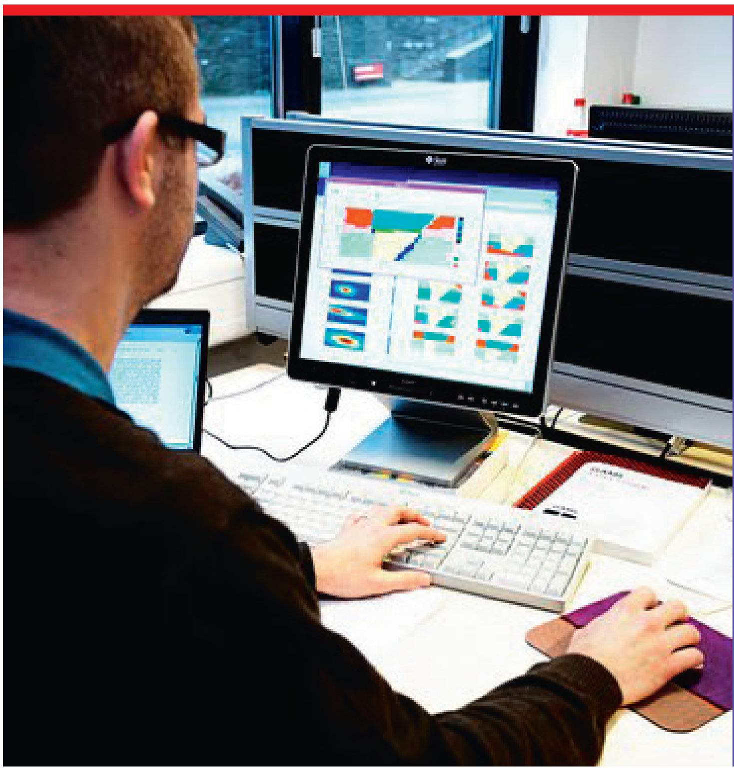
Partial Path Column Generation for the Vehicle Routing Problem

Jepsen, Mads Kehlet; Petersen, Bjørn

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

Partial Path Column Generation for the Vehicle Routing Problem



Report 12.2009

DTU Management Engineering

Mads Kehlet Jepsen
Bjørn Petersen
November 2009

DTU Management Engineering
Department of Management Engineering

Partial Path Column Generation for the Vehicle Routing Problem

Mads Kehlet Jepsen & Bjørn Petersen

DTU Management Engineering, Technical University of Denmark
Produktionstorvet 424, DK-2800 Kongens Lyngby, Denmark

Abstract

Abstract

This paper presents a column generation algorithm for the Capacitated Vehicle Routing Problem (CVRP) and the Vehicle Routing Problem with Time Windows (VRPTW). Traditionally, column generation models of the CVRP and VRPTW have consisted of a Set Partitioning master problem with each column representing a route. Elementary routes (no customer visited more than once) have shown superior results for both CVRP and VRPTW. However, the pricing problems do not scale well when the number of feasible routes increases. We suggest to relax that 'each column is a route' into 'each column is a part of the giant tour'; a so-called partial path, i.e., not necessarily starting and ending in the depot. This way, the length of the partial path can be bounded and a better control of the size of the solution space for the pricing problem can be obtained.

Keywords: *Vehicle Routing Problem, Column Generation, Elementary Shortest Path Problem with Resource Constraints*

---

1 Introduction

The CVRP can be described as follows: A set of customers, each with a demand, needs to be serviced by a number of vehicles all starting and ending at a central depot. Each customer must be visited exactly once and the capacity of the vehicles may not be exceeded. The objective is to service all customers traveling the least possible distance. In this paper we consider a homogeneous fleet, i.e., all vehicles are identical. The VRPTW extends the CVRP by imposing that each customer must be visited within a given time window. The overlap of the CVRP and the VRPTW will in the following be refered to as the VRP.

The standard Dantzig-Wolfe decomposition of the arc flow formulation of the VRP is to split the problem into a master problem (a Set Partitioning Problem) and a pricing problem (an Elementary Shortest Path Problem with Resource Constraints (ESPPRC), where capacity (and time) are the constrained resources). A restricted master problem can be solved with delayed column generation and embedded in a branch-and-bound framework to ensure integrality. Applying cutting planes either in the master or the pricing problem leads to a Branch-and-Cut-and-Price algorithm (BCP). Kohl et al. [24] implemented a successful BCP algorithm for the VRPTW by applying *sub-tour elimination* constraints and *two-path* cuts, Cook and Rich [10] generalized the *two-path* cuts to the k*-path* cuts, and Fukasawa et al. [18] applied a range of valid inequalities for the CVRP based on the branch and cut algorithm of Lysgaard et al. [27]. Common for these BCP algorithms is that all applied cuts are valid inequalities for the VRPTW respectfully the CVRP with regard to the *original* arc flow formulation, and have a structure which makes it possible to handle values of the dual variables in the pricing problem without increasing the complexity of the problem. Fukasawa et al. [18] refer to this as a *robust* approach in their paper. The topic of column generation and BCP algorithms has been surveyed by Barnhart et al. [4] and Lubbecke and Desrosiers [25]. Recently the BCP framework was extended to include valid inequalities for the master problem, more specifically by applying the subset row (SR) inequalities to the Set Partitioning master problem in Jepsen et al. [22] and later by applying Chvátal-Gomory Rank-1 (CG1) inequalities in Petersen et al. [28]. Using an approach where columns with potentially negative reduced cost is enumerated after good upper and lower bounds are found, Baldacci et al. [2] improves the lower bound by adding strengthened capacity inequalities and clique inequalities. Dror [15] showed that the ESPPRC (with time and capacity) is strongly $\mathcal{NP}$-hard, hence a relaxation of the ESPPRC was used as the pricing problem in earlier BCP approaches for the VRPTW. The relaxed pricing problem where non-elementary paths are allowed is denoted the Shortest Path Problem with Resource Constraints (SPPRC) and can be solved in pseudo-polynomial time using for instance a labeling algorithm, see Desrochers [14]. For the problem with a single capacity resource Christofides et al. [9] suggested to remove 2-cycles from the paths, which has also been done by Desrochers et al. [12] for the variant with time windows. Eliminating cycles has been extended by Irnich and Villeneuve [21] to k-cycle elimination (k-cyc-SPPRC) where cycles containing k or less edges are not permitted.

Beasley and Christofides [5] proposed to solve the ESPPRC using Lagrangian relaxation. However, labeling algorithms have recently become the most popular approach to solve the ESPPRC, see e.g. Dumitrescu [16] and Feillet et al. [17]. When solving the ESPPRC with a labeling algorithm a binary resource for each node is added which increases the complexity of the algorithm compared to solving the SPPRC or the k-cyc-SPPRC. Righini and Salani [29] developed a labeling algorithm using the idea of Dijkstra's bi-directional shortest path algorithm that expands both forward and backward from the depot and connects routes in the middle, thereby potentially reducing the running time of the algorithm. Furthermore Righini and Salani [30] and Boland et al. [6] proposed a decremental state space algorithm that iteratively solves a SPPRC and by applying resources forces nodes to be visited at most once. Recently Chabrier [7], Danna and Pape [11], and Salani [32] successfully solved several previously unsolved instances of the VRPTW from the benchmarks of Solomon [33] using a labeling algorithm for the ESPPRC. However, these algorithms have some weaknesses when dealing with very long (in the number of visited nodes) paths when resource constraints are not tight. Not so recently Christofides and Eilon [8] introduced the giant-tour representation in which all the routes are represented by one single *giant* tour, i.e., all the routes are concatenated into a single tour.

We propose a decomposition approach based on the generation of partial paths and the concatenation of these. In the bounded partial path decomposition approach the main idea is to limit the solution space of the pricing problem by bounding some resource, e.g., the number of nodes on a path or the capacity on it. The master problem combines a known number of these bounded partial paths to ensure all customers are visited.

The paper is organized as follows: In Section 2 we describe how to use the giant tour formulation of VRP to obtain the partial path formulation. Section 3 introduces a mathematical model based on partial paths and Section 4 shows how the model is decomposed using the Dantzig-Wolfe decomposition principle and describes how to calculate the reduced cost of columns when delayed column generation is used. Section 5 describes how to use the load resource to divide the solution space, Section 6 presents computational results and brief descriptions of the applied cutting planes, and Section 7 concludes the paper.

2 Bounded partial paths

The VRP can formally be stated as: Given a graph $G(V, A)$ with nodes V and arcs A, a set R of resources $R = \{\text{load (and time)}\}$ where each resource $r \in R$ has a lower bound a_i^r and an upper bound b_i^r for all $i \in V$ and a positive consumption τ_{ij}^r when using arc $(i, j) \in A : i \in C$, find a set of routes starting and ending at the depot node $0 \in V$ satisfying all resource limits, such that the cost is minimized and all customers $C = V \setminus \{0\}$ are visited.

A solution to the VRPTW: $v_0 \to c_1^1 \to \ldots \to c_{k_1}^1 \to v_0, v_0 \to c_1^2 \to \ldots \to c_{k_2}^2 \to v_0, \ldots, v_0 \to c_1^n \to \ldots \to c_{k_n}^n \to v_0$ can be represented by the giant-tour representation of Christofides and Eilon [8]:

$$v_0 \to c_1^1 \to \ldots \to c_{k_1}^1 \to v_0 \to c_1^2 \to \ldots \to c_{k_2}^2 \to v_0 \to \ldots \to v_0 \to c_1^n \to \ldots \to c_{k_n}^n \to v_0$$

which is one long path visiting all customers once and the depot several times. The consumption of resources $r \in R$ is reset each time the depot node is encountered.

The idea is to partition the problem so that the solution space of each part is smaller than the original problem. This is done by splitting the giant-tour into smaller segments by imposing an upper limit on some resource, e.g., bounding the path length in the number of nodes. In the following the number of visited customers is considered the bounding resource, i.e., the number of visits to the non-depot node set C. Each segment represents a partial path of the giant-tour. With a fixed number of customers on each partial path, say L, a fixed number of partial paths, say K, is needed to ensure that all customers are visited, i.e., $L \cdot K \geq |C|$. The partial paths can start and end in any node in V and can visit the depot several times. Example of a partial path:

$$c_1 \to c_2 \to v_0 \to c_3 \to v_0 \to c_4$$

Consider the graph $G'(V', A')$ consisting of a set of layers $K = \{1, \dots, |K|\}$, each one representing G for a partial path. Let G^k be the sub graph of G' representing layer k with node set $V^k = \{(i,k) : i \in V\}$ for all $k \in K$ and arc set $A^k = \{(i,j,k) : (i,j) \in A\}$ for all $k \in K$. Let $A^* = \{(i,i,k) : (i,k) \in V^k \wedge (i,k+1) \in V^{k+1} \wedge k \in K\}$ be the set of interconnecting arcs, i.e., the arcs connecting a layer k with the layer above k namely layer $k+1$ for all $k \in K$ and all nodes $i \in V$ (layer $|K|+1$ is defined to be layer $1 \in K$ and layer 0 is defined to be layer $|K| \in K$). Let $V' = \bigcup_{k\in K} V^k$ and let $A' = \bigcup_{k\in K} A^k \cup A^*$. An illustration of G' can be seen on Figure 1. Note, that arc (i,i,k) does not exist in A^k and that arc (i,j,k) with $i \neq j$ does exist in A^*, so all arcs $(i,j,k) \in A'$ can be uniquely indexed. With the length of a path defined as the number of customers on it, the problem is now to find partial paths of length at most L in $|K|$ layers with $L \cdot |K| \geq |C| > L \cdot (|K| - 1)$, so that each partial path p ending in node $i \in V$ is met by another partial path p' starting in i. All partial paths are combined while not visiting any customers more than once and satisfying all resource windows. A customer $c \in C$ is considered to be on a partial path p if c is visited on p and is not the end node of p.

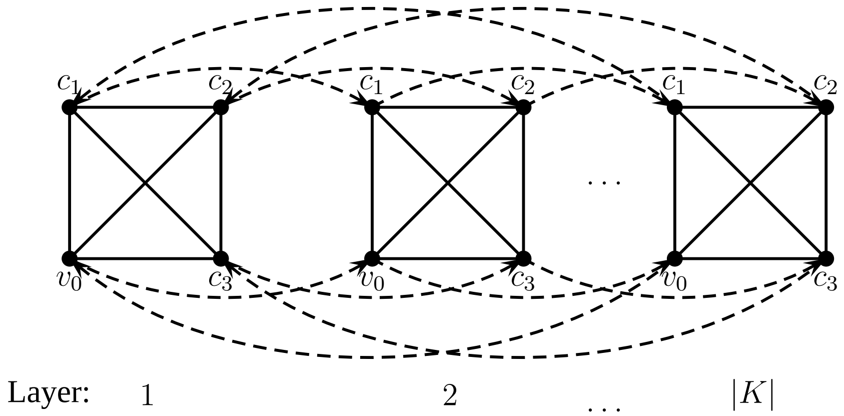


Figure 1: Illustration of G' with $|C| = 3$, $|K| = 3$, and $|L| = 1$. Edges (full-drawn) represent two arcs; one in each direction. Dashed lines are the interconnecting arcs A^*.

Let L be the upper bound on the length of each partial path, and let $|C|$ be the length of the combined path (the giant-tour). Now, exactly $|K| = \lceil |C|/L \rceil$ partial paths are needed to make the combined path, since $L \lceil |C|/L \rceil \geq |C| > L (\lceil |C|/L \rceil - 1)$. Note that given a $|K|$, L can be reduced to $L = \lceil |C|/|K| \rceil$.

In section 5 we will go into detail on how to use the load resource to divide the giant tour.

3 The Vehicle Routing Problem

A formal model was given in Section 2 and here several mathematical models are presented.

2-index formulation of the VRP In the following let c_{ij} be the cost of arc $(i,j) \in A$, x_{ij} be the binary variable indicating the use of arc $(i,j) \in A$, and T^r_{ij} (the resource stamp) be the consumption of resource $r \in R$ at the beginning of arc $(i,j) \in A$. Let $\delta^+(i)$ and $\delta^-(i)$ be the set of outgoing respectively ingoing arcs of node $i \in V$. The mathematical model of VRP adapted from Bard et al. [3] and Ascheuer et al. [1]:

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{(i,j)\in\delta^+(i)} x_{ij} = 1 \qquad \forall i \in C \tag{2}$$

$$\sum_{(j,i)\in\delta^-(i)} x_{ji} = \sum_{(i,j)\in\delta^+(i)} x_{ij} \qquad \forall i \in V \tag{3}$$

$$\sum_{(j,i)\in\delta^-(i)} (T^r_{ji} + \tau^r_{ji}x_{ji}) \leq \sum_{(i,j)\in\delta^+(i)} T^r_{ij} \qquad \forall r \in R,\ \forall i \in C \tag{4}$$

$$a_i x_{ij} \leq T^r_{ij} \leq b_i x_{ij} \qquad \forall r \in R,\ \forall (i,j) \in A \tag{5}$$

$$T^r_{ij} \geq 0 \qquad \forall r \in R,\ \forall (i,j) \in A \tag{6}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{7}$$

The objective (1) sums up the cost of the used arcs. Constraints (2) ensure that each customer is visited exactly once, and (3) are the flow conservation constraints. Constraints (4) and (5) ensure the resource windows are satisfied. It is assumed that the bounds on the depot are always satisfied. Note, no sub-tours can be present since only one resource stamp per arc exists and the arc weights are positive for all $(i,j) \in A : i \in C$.

For a one dimensional resource such as load the capacity constrains $x(\delta^+(S)) \geq r(S)$, where $r(S)$ is a lower bound on the number of vehicles needed to service the set S, can be used instead of equations (4) to (6).

3-index formulation of the VRP Let x^k_{ij} be the variable indicating the use of arc $(i,j,k) \in A'$. Problem (1)–(7) is rewritten:

$$\min \sum_{k\in K}\sum_{(i,j)\in A} c_{ij}x^k_{ij} \tag{8}$$

$$\text{s.t.} \sum_{k\in K}\sum_{(i,j)\in\delta^+(i)} x^k_{ij} = 1 \qquad \forall i \in C \tag{9}$$

$$\sum_{(i,j)\in\delta^+(i)} x^k_{ij} \leq 1 \qquad \forall k \in K,\ \forall i \in C \tag{10}$$

$$\sum_{k\in K}\left(x^{k-1}_{ii} + \sum_{(j,i)\in\delta^-(i)} x^k_{ji}\right) = \sum_{k\in K}\left(x^k_{ii} + \sum_{(i,j)\in\delta^+(i)} x^k_{ij}\right) \qquad \forall i \in V \tag{11}$$

$$x^{k-1}_{ii} + \sum_{(j,i)\in\delta^-(i)} x^k_{ji} = x^k_{ii} + \sum_{(i,j)\in\delta^+(i)} x^k_{ij} \qquad \forall k \in K,\ \forall i \in V \tag{12}$$

$$\sum_{k\in K}\sum_{i\in V} x^k_{ii} = K \tag{13}$$

$$\sum_{i \in C} \sum_{(i,j) \in A} x_{ij}^k \leq L \qquad \forall k \in K \tag{14}$$

$$\sum_{k \in K} \sum_{(j,i) \in \delta^-(i)} \left(T_{ji}^{rk} + \tau_{ji}^r x_{ji}^k\right) \leq \sum_{k \in K} \sum_{(i,j) \in \delta^+(i)} T_{ij}^{rk} \qquad \forall r \in R,\ \forall i \in C \tag{15}$$

$$\sum_{(j,i) \in \delta^-(i)} \left(T_{ji}^{rk} + \tau_{ji}^r x_{ji}^k\right) \leq \sum_{(i,j) \in \delta^+(i)} T_{ij}^{rk} \qquad \forall r \in R,\ \forall k \in K,\ \forall i \in C \tag{16}$$

$$a_i \sum_{k \in K} x_{ij}^k \leq \sum_{k \in K} T_{ij}^{rk} \leq b_i \sum_{k \in K} x_{ij}^k \qquad \forall r \in R,\ \forall (i,j) \in A \tag{17}$$

$$a_i x_{ij}^k \leq T_{ij}^{rk} \leq b_i x_{ij}^k \qquad \forall r \in R,\ \forall k \in K,\ \forall (i,j) \in A \tag{18}$$

$$T_{ij}^{rk} \geq 0 \qquad \forall r \in R,\ \forall k \in K,\ \forall (i,j) \in A \tag{19}$$

$$x_{ij}^k \in \{0,1\} \qquad \forall k \in K,\ \forall (i,j) \in A \tag{20}$$

The objective (8) sums up the cost of the used edges. Constraints (9) ensure that all customers are visited exactly once, while the redundant constraints (10) ensure that no customer is visited more than once. Constraints (11) maintain flow conservation between the original nodes V, and can be rewritten as

$$\sum_{k \in K} \sum_{(j,i) \in \delta^-(i)} x_{ji}^k = \sum_{k \in K} \sum_{(i,j) \in \delta^+(i)} x_{ij}^k \qquad \forall i \in V$$

since $\sum_{k \in K} x_{ii}^{k-1} = \sum_{k \in K} x_{ii}^k$. Constraints (12) maintain flow conservation within a layer. Constraint (13) ensures that K partial paths are selected and constraints (14) that the length of the partial path in each layer is at most L. Constraints (15) connect the resource variables on a global level and constraints (16) connect the resource variables within each single layer, note that since there is no (15) and (16) for the depot it is not constrained by resources. Constraints (17) globally enforce the resource windows and the redundant constraints (18) enforce the resource windows within each layer.

4 Dantzig-Wolfe decomposition

The 3-index formulation of the VRP (8)–(20) is Dantzig-Wolfe decomposed whereby a master and a pricing problem is obtained.

4.1 Master problem

Let λ_p be the variable indicating the use of partial path p. Using Dantzig-Wolfe decomposition where the constraints (9), (11), (13), (15), and (17) are kept in the master problem the following master problem is obtained:

$$\min \sum_{p \in P} c_p \lambda_p \tag{21}$$

$$\text{s.t.} \sum_{p \in P} \sum_{(i,j) \in \delta^+(i)} \alpha^p_{ij} \lambda_p = 1 \qquad \forall i \in C \tag{22}$$

$$\sum_{p \in P : e^p = i} \lambda_p = \sum_{p \in P : s^p = i} \lambda_p \qquad \forall i \in V \tag{23}$$

$$\sum_{p \in P} \lambda_p = K \tag{24}$$

$$\sum_{(j,i) \in \delta^-(i)} \left(T^r_{ji} + \sum_{p \in P} \tau^r_{ji} \alpha^p_{ji} \lambda_p \right) \leq \sum_{(i,j) \in \delta^+(i)} T^r_{ij} \qquad \forall r \in R,\ \forall i \in C \tag{25}$$

$$a_i \sum_{p \in P} \alpha^p_{ij} \lambda_p \leq T^r_{ij} \leq b_i \sum_{p \in P} \alpha^p_{ij} \lambda_p \qquad \forall r \in R,\ \forall (i,j) \in A \tag{26}$$

$$T^r_{ij} \geq 0 \qquad \forall r \in R,\ \forall (i,j) \in A \tag{27}$$

$$\lambda_p \in \{0,1\} \qquad \forall p \in P \tag{28}$$

Where α^p_{ij} is the number of times arc $(i,j) \in A$ is used on path $p \in P$ and s^p and e^p indicates the start respectively the end node of partial path $p \in P$. Constraints (22) ensure that each customer is visited exactly once. Constraints (23) link the partial paths together by flow conservation. Constraint (24) is the convexity constraint ensuring that K partial paths are selected. Constraints (25) and (26) enforce the resource windows.

Note, equality is not needed in constraint (22) and (24) due to the minimizing objective and positive edge cost.

Bounds: Before we turn our attention to solving the pricing problem we consider the bounds obtained by the decomposition.

Theorem 1. *Let z_{lp} be an LP-relaxed solution to* (1)–(7) *and let z_{pp} be an LP-relaxed solution to* (21)–(28) *then $Z_{lp} \leq Z_{pp}$ for all instances of VRP and $Z_{lp} < Z_{pp}$ for some instances of VRP.*

Proof. $Z_{lp} \leq Z_{pp}$ since all solutions to (21)–(28) map to solutions to (1)–(7). An instance with $Z_{lp} < Z_{pp}$ is obtained with four customers each with a demand of resource r of half the global maximum b_r of r, the distance from the customers to the depot larger than the distance between the customers, and $L = 4$. The solution to (21)–(28) would use the expensive edges four times, whereas the solution to (1)–(7) only would use them twice. □

4.2 Pricing problem:

The $|K|$ pricing problems corresponding to the master problem (21)–(28) contains constraints (10), (12), (14), (16), and (18) and can be formulated as a single ESPPRC where the depot is allowed to be visited more than once. Let s and e be a super source respectively a super target node. Arcs (s,i) and (i,e) for all $i \in V$ are added to G.

$$\min \sum_{(i,j)\in A} \bar{c}_{ij} x_{ij} \tag{29}$$

$$\text{s.t.} \sum_{(s,i)\in\delta^+(s)} x_{si} = 1 \tag{30}$$

$$\sum_{(i,e)\in\delta^-(e)} x_{ie} = 1 \tag{31}$$

$$\sum_{(i,j)\in A} x_{ij} \leq 1 \qquad \forall i \in C \tag{32}$$

$$\sum_{(j,i)\in\delta^-(i)} x_{ji} = \sum_{(i,j)\in\delta^+(i)} x_{ij} \qquad \forall i \in V \tag{33}$$

$$\sum_{i\in C} \sum_{(i,j)\in A} \tau_{ji}^{r_{bound}} x_{ij} \leq L \tag{34}$$

$$\sum_{(j,i)\in\delta^-(i)} (T_{ji}^r + \tau_{ji}^r x_{ji}) \leq \sum_{(i,j)\in\delta^+(i)} T_{ij}^r \qquad \forall r \in R,\ \forall i \in C \tag{35}$$

$$a_i x_{ij} \leq T_{ij}^r \leq b_i x_{ij} \qquad \forall r \in R,\ \forall (i,j) \in A \tag{36}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{37}$$

The objective (29) minimizes the reduced cost of a column. Constraints (30) and (31) ensure that the path starts in s respectively ends in e. Constraints (32) dictates that no node is visited more than once, thereby ensuring elementarity and constraints (33) conserve the flow. Constraint (34) ensures that the partial path does not use more then the allowed amount L of the restricted resource r_{bound}. Constraints (35) and (36) ensure the resource windows are satisfied for all customers. Note, since the depot is missing in (35) each time a path leaves the depot a resource is only restricted by its lower limit a_0^r for all $r \in R$.

Let π ($\pi_i \geq 0 : \forall i \in C$) be the duals of (22), let $\pi_0 = 0$, let μ be the duals of (23), let $\beta \leq 0$ be the dual of (24), let ν ($\nu \leq 0 : \forall i \in C$) be the duals of (25), let $\nu_0 = 0$, and let $\underline{\omega} \leq 0$ and $\overline{\omega} \geq 0$ be the dual of (26). The cost of the arcs in this ESPPRC are then given as:

$$\bar{c}_{ij} = -\beta + \begin{cases} c_{ij} - \pi_i - \tau_{ij}\nu_j - a_i\underline{\omega}_i + b_i\overline{\omega}_i & \forall (i,j) \in A \setminus (\delta^+(s) \cup \delta^-(e)) \\ \mu_j & \forall (s,j) \in \delta^+(s) \\ \mu_i & \forall (i,e) \in \delta^-(e) \end{cases}$$

and the pricing problem becomes finding the shortest path from s to e.

Solving the pricing problem: ESPPRCs can be solved by labeling algorithms. For details regarding labeling algorithms we refer to Desaulniers et al. [13], Irnich [19], Irnich and Desaulniers [20], and Righini and Salani [31].

Branching: Integrality can be obtained by branching on the original variables, which can be accomplished by cuts in the master problem (see Vanderbeck [34]), e.g., let X_{ij} be the set of partial paths that utilize arc (i, j) then the branch rule $x_{ij} = 0 \lor x_{ij} = 1$ can be expressed by:

$$\sum_{p \in X_{ij}} \lambda_p = 0 \lor \sum_{p \in X_{ij}} \lambda_p = 1.$$

5 Bounding the Load Resource

Bounding the load resource is a bit more complicated than bounding the number of customers. The issue is that we have to ensure that any feasible solution to the original problem is still feasible. To do this we shall consider a less constrained solution set namely the solutions that satisfy the bin packing problem within the VRP.

Let the total demand of the customers be $Q_t = \sum_{i \in C} d_i$. A lower bound for the number of layers needed is: $K = \lceil Q_t/L \rceil$. Assuming that the largest demand $d_{max} = \max_{i \in C} d_i \leq L$ the upper bound on the number of layers is $2K$ or alternatively we have:

$$L = 2\left\lceil \frac{Q_t}{K} \right\rceil - 1$$

We define the excess capacity Q_e as $Q_e = K \times L - Q_t$. The issue with selecting L in the above fashion is that the excess capacity $Q_e \geq Q_t - K$ which can be a very large number. The high excess capacity can potentially lead to pure bounds.

An alternative to choosing L almost twice the size of Q_t is to only let it be the largest customer d_{max} bigger than Q_t, namely $L = \left\lceil \frac{Q_t}{K} \right\rceil - 1 + d_{max}$. If $d_{max} \leq \left\lceil \frac{Q_t}{K} \right\rceil - 1$ there will be less excess capacity. It is possible to choose L even better by introducing the concept of connectors. A connector is a single arc between two nodes which combines two partial paths. Figure 2 illustrates the idea of the connectors for a single node in layer 1 and layer 2. Each node i has an connector to a node $j \in V$ where $i \neq j$ in the next layer. To model the connectors we introduce new variables y^k_{ij} for all $i, j \in V$ and for all $k \in K$. These variables substitute the variables x^k_{ii} by connecting every node $(i, k) \in V^k$ in each layer $k \in K$ with all the other nodes $(j, k+1) \in V^{k+1} : (j, k+1) \neq (i, k+1)$ in the layer above. Furthermore, constrains (11) are modified to:

$$\sum_{k \in K} \sum_{(j,i) \in \delta^-(i)} \left(x^k_{ji} + y^k_{ji}\right) = \sum_{(i,j) \in \delta^+(i)} \left(x^k_{ij} + y^k_{ij}\right), \quad \forall i \in V$$

This ensures the global flow by taking the flow of the connectors into account. A similar substitution is made in constraint (12) and (13). The connectors are also present in the resource constraints where they are added to any sum bounding the time variables. (15) is therefore changed to:

$$\sum_{k \in K} \sum_{(j,i) \in \delta^-(i)} \left(T^{rk}_{ji} + \tau^r_{ji}\left(x^k_{ji} + y^k_{ji}\right)\right) \leq \sum_{k \in K} \sum_{(i,j) \in \delta^+(i)} T^{rk}_{ij}, \quad \forall r \in R, \forall i \in C$$

Similar addition is made for constrains (16), (17), and (18). Connectors can be handled in a labeling algorithm by allowing an additional edge to be taken when capacity is reached. However, one needs to be carefull when updating reachability.

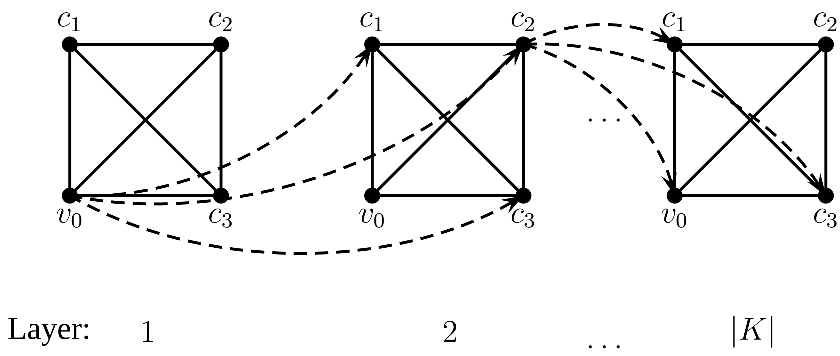


Figure 2: Illustration of connector system

6 Computational Results

In this section we present the computational results for the new model. We have focused on the strength of the model and is therefore only considering the root bound. The test is divided into a comparison with algorithms for the CVRP and the VRPTW. We run on the CVRP instances by Augerat et al. and Christofides and Eilon available at www.branchandcut.org. For the VRPTW we use the solomon 1 and 2 instances with 100 costumers.

On the CVRP instances we show how our bound compares to the Branch-and-Cut bound obtained by adding the capacity cuts and the Branch-and-Cut-and-Price bound obtained using the 2-cyc-SPPRC as the pricing problem. Both bounds have been computeded by Fukasawa et al. [18]. For the VRPTW instances we compare our bound to the Branch-and-Cut bound computed by Kallehauge et al. [23] and the elementary bounds computed by Petersen et al. [28].

For the partial path model we add the capacity constrains using the separation algorithm by Lysgaard [26] and do not include the time variables in the model. For the CVRP instances the size of the partial paths L are set to half the vehicles capacity. If there exist customers where $d_i \geq L - d_{min}$ they are removed from the total sum before K is calculated. After calculation of K, L is recalculated based on the modified total sum Q'_t and K e.g. $K = \left\lceil \frac{Q'_t}{L} \right\rceil$. For the VRPTW instances the capacity is varied between $\frac{1}{2}$ to $\frac{1}{15}$ of the vehicles capacity.

6.1 Results for CVRP

In table 1 we compare the lower bound obtained by the partial path to the bounds of Branch-and-Cut with capacity inequalities (BAC) and the Branch-and-Cut-and-Price algorithm using two cycle elimination (2-cyc). On the A, B, and E instances the bound of the partial path algorithm (PAR) is not much better than the bound of the BAC algorithm and is far from the bound of the 2-cyc algorithm. For one single instance A-n53-k7 the bound is worse. For the P instances the bound is a bit better than the BAC bound but still much worse than the 2-cyc bound. In general

Instance	BAC	PAR	2-cyc	OPT	Instance	BAC	PAR	2-cyc	OPT
A-n53-k7	996.6	996.3	1002.2	1010	B-n50-k7	740.0	741.0	741.0	741
A-n54-k7	1130.7	1133.8	1150.3	1167	B-n50-k8	1279.2	1279.8	1291.8	1312
A-n55-k9	1055.9	1056.1	1066.4	1073	B-n51-k7	1024.6	1024.6	1025.9	1032
A-n60-k9	1316.5	1316.7	1341.6	1354	B-n52-k7	745.0	745.3	746.4	747
A-n61-k9	1004.8	1006.7	1018.8	1034	B-n56-k7	703.4	703.6	704.5	707
A-n62-k8	1244.1	1249.1	1273.2	1288	B-n57-k7	1148.6	1148.6	1150.9	1153
A-n63-k9	1572.2	1578.4	1603.5	1616	B-n57-k9	1586.7	1588.8	1589.2	1598
A-n63-k10	1262.2	1264.4	1294.2	1314	B-n63-k10	1478.9	1479.5	1484.2	1496
A-n64-k9	1340.1	1345.3	1378.8	1401	B-n64-k9	858.5	859.1	860.2	861
A-n65-k9	1151.1	1152.0	1166.6	1174	B-n66-k9	1295.2	1295.8	1303.6	1316
A-n69-k9	1108.9	1110.9	1138.7	1159	B-n67-k10	1023.8	1024.0	1026.4	1032
P-n50-k8	596.9	600.8	615.7	631	B-n68-k9	1256.8	1257.0	1261.6	1272
P-n55-k10	646.7	660.3	680.0	604	B-n78-k10	1202.3	1202.4	1212.6	1221
P-n55-k15	895.1	904.6	967.5	989	E-n51-k5	514.5	514.6	519.0	521
P-n60-k10	708.3	715.5	737.2	744	E-n76-k7	661.4	663.1	669.9	682
P-n60-k15	903.3	926.9	961.2	968	E-n76-k8	711.2	714.3	726.0	735
P-n65-k10	756.5	763.7	785.2	792	E-n76-k10	789.5	796.4	816.8	830
P-n70-k10	786.9	791.8	813.4	827	E-n76-k14	948.1	964.2	1004.8	1021

Table 1: Lower bounds Results for CVRP

we can conclude that it does not appear as a good idea to pursue a Branch-and-Cut-and-Price algorithm for CVRP based on the Partial Path relaxation idea.

6.2 Results for VRPTW

The results for the VRPTW is devided into two tables. For the type 1 Solomon instances it has not been able to find a Branch-and-Cut bound in the literature and therefore only report the lower bound obtained by Branch-and-Cut-and-Price with the ESPPRC as the pricing problem (ESPPRC). Furthermore, the value of L can be chosen higher than for the type 2 Solomon instances since the type 1 Solomon instances have much smaller vehicle capacity. For the type 2 Solomon instances the computed bounds for the partial path algorithm with a Branch-and-Cut (BAC) bound are also compare.

In table 2 the results for the type 1 Solomon instances are shown. As can be seen the bound does not increased much when s is changed from 4 to 3. However, for most of the R and RC instances the bound changes a bit when changing from $s = 3$ to $s = 2$. In general, the best bound for the partial path algorithm is far from the bound of the ESPPRC algorithm on the R and RC instances.

In table 3 the bounds computed on the Solomon type 2 instances are compared. For the C instances the bound is almost the same as the BAC and ESPPRC bounds. For both R and RC the bound is often far from the BAC algorithms bound and even further from the ESPPRC

		ESPPRC	PAR		
Instance	**Opt.**		$s=2$	$s=3$	$s=4$
R101	1637.7	1631.2	1624.0	1611.9	1611.9
R102	1466.6	1466.6	1094.4	1071.8	1071.8
R103	1208.7	1206.8	880.1	874.3	874.4
R104	971.5	956.9	812.3	812.0	812.1
R105	1355.3	1346.2	1204.0	1160.1	1159.0
R106	1234.6	1227.0	943.2	937.5	937.7
R107	1064.6	1053.3	829.9	830.0	829.8
R108	932.1	913.6	809.1	808.8	809.0
R109	1146.9	1134.3	884.5	864.1	864.0
R110	1068.0	1055.6	812.9	812.4	812.4
R111	1048.7	1034.8	822.2	822.1	821.9
R112	948.6	926.8	804.3	804.3	804.3
C101	827.3	827.3	827.3	827.3	827.3
C102	827.3	827.3	819.9	819.9	820.0
C103	826.3	826.3	819.9	819.9	820.0
C104	822.9	822.9	818.0	818.0	818.0
C105	827.3	827.3	827.3	827.3	827.3
C106	827.3	827.3	827.3	827.3	827.3
C107	827.3	827.3	827.3	827.3	827.3
C108	827.3	827.3	818.9	818.8	818.9
C109	827.3	827.3	817.8	817.8	817.8
RC101	1619.8	1584.1	1324.4	1286.4	1286.4
RC102	1457.8	1406.3	1030.2	1030.1	1030.1
RC103	1258.0	1225.6	979	978.8	978.9
RC104	1132.3	1101.9	968.8	968.5	968.7
RC105	1513.7	1472.0	1097.9	1092.1	1092.1
RC106	1401.2	1318.8	1036,2	1035.0	1034.7
RC107	1207.8	1183.4	973.8	973.6	973.8
RC108	1114.2	1073.5	964.1	964.0	963.6

Table 2: Lower bound results for the VRPTW for the Solomon type 1 instances. s is the fraction of the original capacity of the vehicle, that is $L = \frac{Q}{s}$.

algorithms bound, however in the case of RC208 the best bound obtained by the partial path algorithm is better than the bound obtained by the BAC algorithm. For the type 2 Solomon instances a small increase in the bound of the partial path algorithm is seen as L increases.

		ESPPRC	**BAC**	**PAR**		
Instance	**Opt.**			$s = 8$	$s = 10$	$s = 15$
R201	1143.2	1140.3	1123.6	1055.0	1040.1	1028.2
R202	1029.6	1022.3	888.6	772.3	761.0	758.5
R203	870.8	867.0	748.1	666.3	665.6	665.6
R204	-	-	661.9	645.0	645.0	645.0
R205	949.8	939.0	899.7	795.5	785.4	779.3
R206	875.9	866.9	783.6	690.1	685.1	684.7
R207	794.0	790.7	714.8	657.5	657.5	657.5
R208	-	-	651.6	644.3	644.3	644.3
R209	854.8	841.5	785.2	693.1	686.3	684.9
R210	900.5	889.4	798.2	693.3	687.5	686.1
R211	-	-	645.1	644.3	644.3	644.3
C201	589.1	589.1	589.1	589.1	589.1	589.1
C202	589.1	589.1	589.1	587.9	587.9	587.9
C203	588.7	588.7	584.4	581.7	581.7	581.7
C204	588.1	588.1	583.5	578.6	578.6	578.6
C205	586.4	586.4	586.4	582.7	582.2	582.2
C206	586.0	586.0	586.0	582.2	582.2	582.2
C207	585.8	585.8	585.6	584.5	584.5	584.5
C208	585.8	585.8	585.8	582.2	582.2	582.2
RC201	1261.8	1256.0	1249.2	1121.0	1104.8	1099.4
RC202	1092.3	1088.1	940.1	726.2	721.6	721.6
RC203	923.7	922.6	781.6	664.6	664.1	664.1
RC204	-	-	692.7	653.1	653.1	653.1
RC205	1154.0	1147.7	1081.7	827.7	817.1	816.6
RC206	1051.1	1038.6	974.8	816.7	811.0	811.0
RC207	962.9	947.4	832.4	686.4	686.4	686.4
RC208	-	-	647.7	651.7	651.7	651.7

Table 3: Lower bound results for the VRPTW for the Solomon type 2 instances. s is the fraction of the original capacity of the vehicle, that is $L = \frac{Q}{s}$.

7 Conclusion

A new decomposition model of the VRP has been presented with the ESPPRC as the pricing problem. The model facilitates control of the running time of the pricing problem. Due to the aggregation of the model, LP relaxed bounds of (21)–(28) are better than the direct model (1)–(7). Since (21)–(28) is a relaxation of the traditional Dantzig-Wolfe decomposition model with elementary routes as columns, the LP relaxed bounds may be weaker yielding a larger branch-and-bound tree. It has been shown that the bound of the LP relaxation is sometimes better than that of a standard Branch-and-Cut algorithm, it is unfortunately far from the best obtainable bounds of Branch-and-Cut-and-Price algorithms. Especially for CVRP the results are very disappointing since we are not able to produce better bound than the two cycle elimination based relaxation. For VRPTW the results are a bit more encouraging since the bounds are in some cases better than that of the Branch-and-Cut algorithm even though the same cuts are not added. This leaves room for improvement for a Branch-and-Cut-and-Price algorithm based on partial paths.

Future work: The difference in bound quality can be decreased with the use of special purpose cutting planes, which this paper has not fucused on. Furthermore, effective cuts such as Subset Row-inequalities by Jepsen et al. [22] and Chvátal-Gomory Rank-1 cuts (see Petersen et al. [28]) can be applied to the Set Partition master problem to strengthen the bound.

More and better cuts have been added to the VRPTW Branch-and-Cut algorithm used in this paper for comparison, but all of these cuts could also be added to this model obtaining at least as good a bound.

Considering the approach of Baldacci et al. [2] where columns are enumerated dependent on strong upper and lower bounds, it should be clear that the partial path approach should contain fewer enumerated columns due to the smaller solution space of the pricing problem. Combining the relatively strong bound with the small solution space a powerful strategy should be obtained.

References

[1] N. Ascheuer, M. Fischetti, and M. Grötschel. Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. *Mathematical Programming*, 90(3):475–506, May 2001.

[2] R. Baldacci, N. Christofides, and A. Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115(2):351–385, 2008. doi: 10.1007/s10107-007-0178-5.

[3] J. F. Bard, G. Kontoravdis, and G. Yu. A branch-and-cut procedure for the vehicle routing problem with time windows. *Transportation Science*, 36(2):250–269, 2002.

[4] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1998.

[5] J.E. Beasley and N. Christofides. An algorithm for the resource constrained shortest path problem. *Networks*, 19:379–394, 1989.

[6] N. Boland, J. Dethridge, and I. Dumitrescu. Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Oper. Res. Lett.*, 34(1):58–68, 2006.

[7] A. Chabrier. Vehicle routing problem with elementary shortest path based column generation. *Computers & Operations Research*, 23(10):2972–2990, 2005.

[8] N. Christofides and S. Eilon. An algorithm for the vehicle-dispatching problem. *Operational Research Quarterly*, 20(3):309–318, 1969.

[9] N. Christofides, A. Mingozzi, and P. Toth. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, 20(1):255–282, Dec 1981. doi: 10.1007/BF01589353. URL `http://dx.doi.org/10.1007/BF01589353`.

[10] W. Cook and J. L. Rich. A parallel cutting plane algorithm for the vehicle routing problem with time windows. Technical Report TR99-04, Computational and Applied Mathematics, Rice University, Houston, Texas, USA, 1999.

[11] E. Danna and C. Le Pape. Accelerating branch-and-price with local search: A case study on the vehicle routing problem with time windows. In G. Desaulniers, J. Desrosiers, , and M. M. Solomon, editors, *Column Generation*, chapter 3, pages 30–130. Springer, 2005.

[12] M. Desrochers, J. Desrosiers, and M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40:342–354, 1992.

[13] G. Desaulniers, J. Desrosiers, J. Ioachim, I. M. Solomon, F. Soumis, and D. Villeneuve. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In T. G. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, chapter 3, pages 57–93. Springer, 1998.

[14] M. Desrochers. *La fabrication d'horaires de travail pour les conducteurs d'autobus par une méthode de génération de colonnes*. PhD thesis, Université de Montréal, Canada, 1986.

[15] M. Dror. Note on the complexity of the shortest path models for column generation in VRPTW. *Operations Research*, 42:977–979, 1994.

[16] I. Dumitrescu. *Constrained Path and Cycle Problems*. PhD thesis, Department of Mathematics and Statistics, University of Melbourne, Australia, 2002.

[17] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44(3):216–229, 2004.

[18] R. Fukasawa, H. Longo, J. Lysgaard, M. Poggi de Aragão, M. Reis, E. Uchoa, and R.F. Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, 106(3):491–511, 2006.

[19] S. Irnich. Resource extension functions: Properties, inversion, and generalization to segments. *OR Spectrum*, 30(1):113–148, January 2008.

[20] S. Irnich and G. Desaulniers. Shortest path problems with resource constraints. In G. Desaulniers, Jacques Desrosiers, and M.M. Solomon, editors, *Column Generation*, chapter 2, pages 33–65. Springer, 2005.

[21] S. Irnich and D. Villeneuve. The shortest-path problem with resource constraints and k-cycle elimination for k $\geq$ 3. *INFORMS J. on Computing*, 18(3):391–406, 2006.

[22] M. Jepsen, B. Petersen, S. Spoorendonk, and D. Pisinger. Subset-row inequalities applied to the vehicle routing problem with time windows. *Operations Research*, 56(2):497–511, 2008.

[23] Brian Kallehauge, Natashia Boland, and Oli B. G. Madsen. Path inequalities for the vehicle routing problem with time windows. *Netw.*, 49(4):273–293, 2007. ISSN 0028-3045. doi: http://dx.doi.org/10.1002/net.v49:4.

[24] N. Kohl, J. Desrosiers, O. B. G. Madsen, M. M. Solomon, and F. Soumis. 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, 33(1):101–116, 1999.

[25] M. E. Lubbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.

[26] J. Lysgaard. CVRPSEP: A package of separation routines for the capacitated vehicle routing problem. Available at: url: www.asb.dk/$\sim$lys, 2003.

[27] J. Lysgaard, A. N. Letchford, and R. W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle problem. *Mathematical Programming*, 100(A):423–445, 2004.

[28] B. Petersen, D. Pisinger, and S. Spoorendonk. Chvátal-Gomory rank-1 cuts used in a Dantzig-Wolfe decomposition of the vehicle routing problem with time windows. In B. Golden, R. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 397–420. Springer, 2008.

[29] G. Righini and M. Salani. Symmetry helps: bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. Technical Report 66, Note del Polo - Ricerca, Dipartimento di Tecnologie dell'Informazione, Universitá degli studi di Milano, 2004.

[30] G. Righini and M. Salani. New dynamic programming algorithms for the resource constrained shortest path problem. Technical Report 69, Note del Polo - Ricerca, Dipartimento di Tecnologie dell'Informazione, Universitá degli studi di Milano, 2005.

[31] G. Righini and M. Salani. Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization*, 3(3):255–273, 2006.

[32] M. Salani. *Branch-and-Price Algorithms for Vehicle Routing Problems*. PhD thesis, Universitá Degli Studi Di Milano, Facoltá di Scienza Matematiche, Fisuche e Naturali Dipartimento di Technologie dell'Informazione, Milano, Italy, 2005.

[33] M. M. Solomon. Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research*, 35:234–265, 1987.

[34] F. Vanderbeck. On Dantzig-Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm. *Operations Research*, 48(1):111–128, 2000.

This paper presents a column generation algorithm for the Capacitated Vehicle Routing Problem (CVRP) and the Vehicle Routing Problem with Time Windows (VRPTW).

Traditionally, column generation models of the CVRP and VRPTW have consisted of a Set Partitioning master problem with each column representing a route. Elementary routes (no customer visited more than once) have shown superior results for both CVRP and VRPTW. However, the pricing problems do not scale well when the number of feasible routes increases. We suggest to relax that 'each column is a route' into 'each column is a part of the giant tour'; a so-called partial path, i.e., not necessarily starting and ending in the depot. This way, the length of the partial path can be bounded and a better control of the size of the solution space for the pricing problem can be obtained.

ISBN 978-87-90855-59-8

DTU Management Engineering
Department of Management Engineering
Technical University of Denmark

Produktionstorvet
Building 424
2800 Kongens Lyngby
Tel. 45 25 48 00
Fax 45 93 34 35

www.man.dtu.dk