Electronic Research Archive

http://www.aimspress.com/journal/era

ERA, 31(2): 615–632.
DOI: 10.3934/era.2023030
Received: 18 September 2022
Revised: 05 November 2022
Accepted: 08 November 2022
Published: 15 November 2022

Research article

Approximate solution of the shortest path problem with resource constraints and applications to vehicle routing problems

Abdelkader Lamamri and Mohammed Hachama*

Department of Mathematics, Université Blida 1, Blida, Algeria

* **Correspondence:** Email: hachamam@gmail.com.

Abstract: Vehicle routing problem (VRP) is a fundamental combinatorial optimization and integer programming problem with several important applications. The VRP is usually solved by using branch-and-bound techniques requiring solving a shortest path problem with resource constraints (SPPRC) and the determination of a lower bound, which can be computed by using column generation. The SPPRC entails finding the minimum cost elementary path in a valuated graph that is subject to constraints on resource consumption. The proposed exact solutions to this hard NP-hard problem require an excessive computation time which increases with the number of resources. In this paper, we propose a new approximate resolution of the SPPRC for acyclic and cyclic graphs. Our method is based on a Lagrangian relaxation of a subset of the constraints and using dominance only on a subset of the resources. This reduces the search space and allows users to efficiently compute solutions used to improve the column generation procedure. Extensive evaluation and comparison to the classical exact method show that the proposed algorithm achieves a good compromise between efficiency and quality of the SPPRC and the VRP solutions. Thus, our method can be used for practical large-scale VRP applications.

Keywords: dynamic programming; Lagrangian relaxation; shortest path problem with resources constraints; column generation; vehicle routing problem

1. Introduction

The vehicle routing problem (VRP) is a class of problems that entails finding an optimal set of routes for a fleet of vehicles to serve a set of customers. The objective of the VRP is to minimize vehicle routes costs while originating and terminating at a depot. The VRP was first proposed by Dantzig and Ramser [1] before being extended with different variants: the VRP with time window (VRPTW) [2–7], the capacitated VRP (CVRP) [8–12] and other VRPs [13–17]. In recent years, the VRPs have attracted much interest [18]. Exact solutions of the VRPs generally result in branch-and-price [13, 19], branch-and-cut [20–22] and branch-cut-and-price algorithms [22,23]. Ben Ticha et al. [24–26] proposed well-performing branch-and-price algorithms for VRPs on multigraphs with multiple time-related attributes and a homogeneous vehicle fleet. In such approaches, the linear relaxation in each branch-and-bound node is by solved column generation, which has proved to be a powerful approach [9,27,28]. Column generation has been widely used to solve a variety of large mathematical programs such as vehicle routing and crew scheduling problems [3,29,30].

Many authors have suggested solving a shortest path problem with resource constraints (SPPRC) introduced by Desrochers [31] as a multi-dimensional generalization of the shortest path problem with time windows. Resolutions methods and applications of the SPPRC have been extensively discussed in the literature [3, 29, 31–33]. These strategies range from exact methods to heuristics and meta-heuristics.

Exact SPPRC resolution techniques usually use the dynamic programming which has a pseudopolynomial complexity. Desrochers and Soumis [32] proposed a label correcting reaching algorithm that extends the Ford-Bellman algorithm to take resource constraints into account. The algorithm has been shown to be successful for tight resource constraints. Feillet et al. [34] adapted the Desrochers algorithm to solve exactly the ESPPRC (SPPRC with elementary path) pricing problems in the context of the VRPTW. Since that, this algorithm has been the backbone of a number of algorithms based on column generation applied to several important problems such as vehicle routing and crew management [16,26]. Table 1 summarizes some reviewed literature that applied column-generation for VRPs as classified based on solution approaches used for solving the sub-problem (SPPRC or ESPPRC).

The main drawback of exact resolution techniques is the computational time which increases with the number of resources, making them suitable for small-scale problems only. To speed up the computations and obtain solutions in a reasonable time, some authors proposed approximate, heuristics and meta-heuristics techniques [14,18,50–55]. These algorithms yield near-optimal solutions and are more suitable for real-life larger-scale applications (e.g., thousands of customers from, dozens of depots with numerous vehicles subject to a variety of constraints). Nagih and Soumis [14] proposed a basic heuristic to quickly produce feasible solutions. But, this approach has been limited to acyclic graphs and its speed has not been proved. Some authors have used simulated annealing [56], genetic algorithms [52, 53], non-dominated sorting genetic algorithms [54] and simplified swarm optimization [55]. Meta-heuristic methods allows to avoid getting trapped in local optima but this comes at a price of slower convergence.

In this paper, we propose a Lagrangian relaxation based algorithm to approximately solve the SPPRC for arbitrary, acyclic and cyclic graphs. In our approach, called "dominance on Lagrangian cost for the SPPRC" (DLC-SPPRC), a dominance is expressed on a subset of the resources while the rest of the resources are dualized in the objective. Optimized parameter update schemes are used to ensure better performances (descent steps, Lagrange multipliers, subgradients).

The rest of the paper is organized as follows. Section 2 describes formally the SPPRC. Section 3 presents our approximate solution method that speeds up the computations while keeping a good approximate solution. In Section 4, we present an application to column generation and show the results obtained for various VRP datasets. The paper ends with the final conclusions.

2. SPPRC

Table 1. Summary of some papers based on the sub-problem solution method.

Authors	Year	Elementary	Non-Elementary	Class of VRP
Desrosiers et al. [3]	1995		*	VRPTW
Kohl et al. [4]	1999		*	VRPTW
Feillet et al. [34]	2004	*		VRPTW
Irnich and Villeneuve [5]	2006		*	VRPTW
Righini and Salani [35]	2006	*		VRPTW
Nagih and Soumis [14]	2006		*	Other VRPs
Fukasawa et al. [10]	2006		*	CVRP
Chabrier [36]	2006	*		VRPTW
Feillet et al. [37]	2007	*		VRPTW
Tagmouti et al. [38]	2007	*		VRPTW
Jepsen et al. [39]	2008	*		VRPTW
Baldacci et al. [11]	2008	*	*	CVRP
Righini and Salani [40]	2008	*		VRPTW
Desaulniers et al. [41]	2008	*		VRPTW
Qureshi et al. [42]	2009	*		Other VRPs
Baldacci et al. [7]	2010		*	VRPTW
Baldacci et al. [12]	2011	*	*	CVRP
Bettinelli et al. [43]	2011	*		Other VRPs
Liberatore et al. [44]	2011	*		VRPTW
Dabia et al. [13]	2013	*	*	Other VRPs
Duque et al. [45]	2015	*		Other VRPs
Fukasawa et al. [8]	2015		*	CVRP
Lozano et al. [46]	2016	*		VRPTW
Pecin et al. [28]	2017	*		VRPTW
Pecin et al. [9]	2017	*	*	CVRP
Lera-Romero and Miranda-Bront [47]	2018	*		Other VRPs
Ben Ticha et al. [26]	2019	*		VRPTW
Himmich et al. [15]	2020		*	Other VRPs
Sadykov et al. [6]	2021		*	VRPTW
Behnke et al. [16]	2021		*	Other VRPs
Dalmeijer and Desaulniers [48]	2021	*		Other VRPs
Mathlouthi et al. [17]	2021	*	*	Other VRPs
Taş [49]	2021	*		Other VRPs

2.1. Notations and preliminaries

Consider a graph $G = (V, A)$, where $V = N \cup \{o, d\}$ is a set of nodes $N = \{v_1, \ldots, v_n\}$ that would be visited from an origin $o = v_0$ to a destination $d = v_{n+1}$ and a set of arcs $A \subset V \times V$. We denote by $\mathcal{R}$ a set of resources of cardinality $|\mathcal{R}|$. An arc $(i, j) \in A$ costs c_{ij} and consumes a quantity $t^r_{ij} \geq 0$ of each resource $r \in \mathcal{R}$. We suppose that the resources satisfy the triangle inequality. If $(i, j) \in A$, then the node i is called a predecessor of j and j is called a successor of i. A path is a finite sequence of nodes

$$P = (v_{k_0} = v_0, v_{k_1}, \ldots, v_{k_m}), \quad \text{where} \quad (v_{k_i}, v_{k_{i+1}}) \in A, \forall i = 0, \ldots, m-1.$$

To each node v_j of P, we associate C^P_j which is the sum of the costs of its composite arcs up to v_j. We denote by $T^{r,P}_j$ the amount of resource $r \in \mathcal{R}$ used to reach the node v_j. For the sake of simplification, we drop the index P from these notations and write C_j and T^r_j. The path P is said to be *feasible* if it satisfies the resource constraints

$$T^r_{k_i} \in [a^r_{k_i}, b^r_{k_i}], \quad \forall r \in \mathcal{R}, \ \forall i \in \{0, .., m\}.$$

A path P can also be represented by $X = (x_{ij})_{(i,j)\in A} \in \{0, 1\}$, where $x_{ij} = 1$ if the nodes v_i and v_j belong to P, and 0 otherwise.

2.2. Problem formulation

The SPPRC seeks a feasible path from the origin to the destination with a minimal cost. Similar to [3], one can formulate the SPPRC as follows:

$$\underset{X}{\text{minimize}} \quad \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{2.1}$$

subject to

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(i,j)\in A} x_{ji} = 0, \quad \forall i \in V \setminus \{o, d\}, \tag{2.2}$$

$$\sum_{(o,j)\in A} x_{oj} = 1; \quad \sum_{(i,d)\in A} x_{id} = 1, \tag{2.3}$$

$$x_{ij}(T^r_i + t^r_{ij} - T^r_j) \leq 0, \quad \forall (i, j) \in A, \forall r \in \mathcal{R}, \tag{2.4}$$

$$a^r_j \leq T^r_j \leq b^r_j, \quad \forall j \in V, \forall r \in \mathcal{R}, \tag{2.5}$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A, \tag{2.6}$$

where $X = (x_{ij})_{i,j}$ represents a path solution. Eqs (2.2) and (2.3) define the flow constraints on the graph; Eq (2.4) encodes the compatibility requirements between flow and time variables, and Eq (2.5) is the time windows constraint. If the problem is feasible, then there exists an optimal solution [14].

2.3. Algorithms

To solve the SPPRC, labeling algorithms have been efficiently applied by many authors, e.g., Desrochers and Soumis [32], Dabia et al. [13] and Sun et al. [57, 58]. Labels are defined for each node to encode partial paths, along with their cost and resource consumption. Each node receives labels and then extend them toward every possible successor node. The algorithm iteratively treats the nodes until no new labels are created.

To limit the proliferation of labels and reduce the computational time, dominance rules are introduced. Dominance relation is a partial ordering applied to eliminate non optimal solutions and retain only non dominated labels. The number of retained labels and, thus, the computation time, increase with the number of resources.

In the next section, we propose an algorithm to determine approximate solutions. The size of the search space is reduced by eliminating some labels with a heuristic dominance procedure.

3. Lagrange SPPRC label correcting algorithm

With each path P_i arriving to a node v_i, we associate a *label* (or *state*) vector

$$L_i = (C_i, T_i^1; \cdots ; T_i^{|\mathcal{R}|}). \tag{3.1}$$

The path associated to a label L is denoted by X^L. The path P is extended to each successor v_j with

$$C_j = C_i + c_{ij} \quad \text{and} \quad T_j^r = \max\{a_j^r, T_i^r + t_{ij}^r\}, \quad \forall r \in \mathcal{R}. \tag{3.2}$$

Let P_i and P_i' be two feasible paths arriving to a node v_i. We say that P_i is dominated by P_i' and we write $P_i \preceq P_i'$ if the label vector of P_i is component-wise less than the one of P_i'. On the other hand, we say that $P_i \leq P_i'$ if the first component of P_i is smaller than the one of P_i' or by comparing recursively the following ones if the first components are equal. At least one of the inequalities should be strict. This defines a partial order relation.

3.1. Lagrangian relaxation

To speed up the resolution of the SPPRC (2.1)–(2.6), we propose to relax a subset of resources $\mathcal{R}_1 \subset \mathcal{R}$ and apply dominance on the remaining resources $\mathcal{R}_2 = \mathcal{R} \backslash \mathcal{R}_1$ only. This is equivalent to consider Eqs (2.4) and (2.5) on $\mathcal{R}_2$ and replace Eq (2.1) by the equation

$$\underset{X}{\text{minimize}} \quad \mathcal{L}(\lambda, X), \tag{1'}$$

where the Lagrangian function $\mathcal{L}$ is defined by

$$\mathcal{L}(\lambda, X) = \sum_{(i,j)\in A} x_{ij} \left(c_{ij} + \sum_{r\in\mathcal{R}_1} \lambda_{i,j}^r \big(\max(a_j, T_i^r + t_{ij}^r) - b_j^r\big)\right). \tag{3.3}$$

Since the flow is supported by at most one incoming arc at each node j from its predecessors i, the multipliers $\lambda_{i,j}^r = \lambda_j^r$ are independent of i. The vector of all multipliers $\lambda_j = (\lambda_j^r)_{r\in\mathcal{R}_1}$ can be obtained by solving the following maximization problem

$$\begin{cases} \max_\lambda & \min_X \ \mathcal{L}(\lambda, X), \\ s.t. & \lambda_j^r \geq 0, \qquad \forall r \in \mathcal{R}_1. \end{cases} \tag{3.4}$$

To solve this optimization sub-problem, one can use a projected sub-gradient algorithm which proved to be efficient in practice. The updating scheme is

$$\lambda_j^{k+1} = \max\left\{\lambda_j^k + \tau_j^k g_j^k, 0\right\} \tag{3.5}$$

where $g_j^k \in \partial\mathcal{L}(\lambda, X)$ is a subgradient vector, $\tau_j^k > 0$ is a step size, and the max operator is taken for each element. We choose

$$g_j^k = \max(a_j, T_i^r + t_{ij}^r) - b_j^r, \tag{3.6}$$

$$\tau_j^k = \frac{\theta_k(Z_{\text{UB}} - \mathcal{L}(\lambda^k, X))}{\|g_j^k\|^2}, \tag{3.7}$$

where Z_{UB} is the best-known-smallest (upper)-bound of the solution of the problem described by (2.1)–(2.6) and $\theta_k \in (0, 2]$ is a scalar. One iteration for the estimation of λ proved to be sufficient (precision and speed) when integrated into column generation.

3.2. Dominance

The dominance at each node j corresponds to the determination of the Pareto optima of the multicriteria problem applied to a set of labels. With the adopted Lagrangian approach, the label vector becomes

$$L_j = \left(C_j + \sum_{r_1 \in \mathcal{R}_1} \lambda_j^{r_1}(T_j^{r_1} - b_j^{r_1});\ T_j^{r_2}, r_2 \in \mathcal{R}_2;\right). \tag{3.8}$$

When the obtained final solution is not feasible (due to relaxation), we need to solve again the relaxed SPPRC by dominating on the Lagrangian cost determined in the first phase and on the resources $\mathcal{R}_2$ and by checking the resource windows in the original space $\mathcal{R}$.

Assuming that all predecessors of the node $j \in N$ have been considered, the dominance at the node j can be interpreted as the determination of the Pareto optima for the multicriteria problem of $|\mathcal{R}_2| + 1$ functions:

$$\min_{(i;(i,j)\in A)} \left(C_i + c_{ij} + \sum_{r_1 \in \mathcal{R}_1} \lambda_j^{r_1}(T_j^{r_1} - b_j^{r_1});\quad \max\left\{a_j^{r_2}, (T_i^{r_2} + t_{ij}^{r_2})\right\}, r_2 \in \mathcal{R}_2;\right)$$
$$\text{st. } T_i^r + t_{ij}^r \le b_j^r, r \in \mathcal{R}. \tag{3.9}$$

3.3. Approximate solution procedure

First, we introduce the following notations.

- E_i: List of labels at a node v_i.
- F_i: The list of paths associated to the labels of E_i.
- S_i: Set of the successors of the node v_i indices.
- Q: List of unprocessed nodes indices.
- *Pareto*(L, E_j): Add a new label L to a set of non-dominated labels E_j while keeping the non-dominance property.
- Z_L is the cost associated with a label L (first component of Eq (3.8)).

Our improved label-correcting procedure is detailed in Algorithms 1 and 2.

Algorithm 1: Modified label-correcting algorithm

input : An arbitrary directed graph $G = (V, A)$, unnecessarily acyclic; $\mathcal{R}$ a set of constraints; $\mathcal{R}_1 \subset \mathcal{R}$ subset of constraints to be included in the objective function; Lagrange multipliers $\lambda_i \in \mathbb{R}_+^{|\mathcal{R}_1|}$ where $v_i \in V - \{o\}$; $\mathcal{R}_2$ a set of resources

output : All non-dominated paths from the source to the destination E_{n+1}, and the associated list of paths F_{n+1}.

```
// Initialization;
R2 = R − R1;
for v_i ∈ V do
|   E_i ← ∅
end
E_0 ← {(0, 0, ··· , 0)};    // The size of the labels is: |E_o| = |R2| + 1
Q ← {0}
// Main loop;
while Q ≠ ∅ do
    // Choose a node to treat;
    Choose i ∈ Q;
    for j ∈ S_i do
        for L_i ∈ E_i do
            if ∀r2 ∈ R2, T_i^{r2} + t_{ij}^{r2} ≤ b_j^{r2} then
                // Create new labels L_j for j by extending those of i;
                L_j = (C_j + Σ_{r1∈R1} λ_j^{r1}(T_j^{r1} − b_j^{r1}); T_j^{r2}, r2 ∈ R2).
                // Apply dominance ;
                E_j ← Pareto(L_j, E_j)
                // Update F_j;
                Update F_j using E_j.
            end
        end
        if L_j has changed then
        |   Q ← Q ∪ {j}
        end
    end
    Q ← Q \ {i}
end
```

where the label is $L_j = \left(C_j + \sum_{r_1 \in \mathcal{R}_1} \lambda_j^{r_1}(T_j^{r_1} - b_j^{r_1});\ T_j^{r_2}, r_2 \in \mathcal{R}_2\right)$ and the feasibility test is $\forall r_2 \in \mathcal{R}_2, T_i^{r_2} + t_{ij}^{r_2} \le b_j^{r_2}$.

Algorithm 2: DLC-SPPRC: Dominance on Lagrangian cost for the SPPRC.

input : An arbitrary directed graph $G = (V, A)$, unnecessarily acyclic; $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ a set of resources constraints composed of two types.

output : $\bar{E}$, all non-dominated feasible paths from the source to the destination node d.

// Parameters choice;

Choose $\theta \in (0,\ 2]$.

$k_{max} = 10$.

// Initialization;

for $v_i \in V - \{o\}$ **do**

  $\lambda_i^{(0)} = (0, \dots, 0)$ a vector of size $|\mathcal{R}_1|$.

// Initial value of the Lagrangian corresponding to the solution of (3.4)

$\bar{Z}_{max} \leftarrow -\infty$.

// Choose an initial solution of the problem (2.1)–(2.6).

$X^0 = (x^0_{ij})_{ij}$.

// Compute the objective function value (1) (upper bound).

$Z_u \leftarrow \sum_{(i,j)\in A} c_{ij} x^0_{ij}$.

$\bar{E} \leftarrow \emptyset$.

$k = 0$.

// Step 1;

while *($k < k_{max}$)* **do**

  // Apply Lagrangian dominance relaxation;

  $[E_{n+1}, F_{n+1}] =$ **Algorithm1**$(G, (\lambda_i^{(k)})_i, \mathcal{R}, \mathcal{R}_1, \mathcal{R}_2)$.

  $(L, X^L) \leftarrow$ the smallest label of E_{n+1} and its associated path F_{n+1}.

  // Update $\bar{Z}_{max}$

  if *($\bar{Z}_{max} < Z_L$)* **then**

    $\bar{Z}_{max} \leftarrow Z_L$

  else

    $\theta \leftarrow \theta/2$.

  if X^L *is feasible* **then**

    if $Z_u > Z_L$ **then**

      $Z_u \leftarrow Z_L$

      $\bar{E} \leftarrow$ all feasible solutions of E_{n+1}.

  // Update Lagrange multipliers

  for $v_i \leftarrow V - \{o\}$ **do**

    $\lambda_i^{(k+1)} \leftarrow$ Apply Eqs (3.6), (3.7) and (3.5).

  if $\|\lambda_i^{(k+1)} - \lambda_i^{(k)}\| < \varepsilon\ \forall i$ **then**

    break

  $k \leftarrow k + 1$.

// Step 2

if $\bar{E} = \emptyset$ **then**

  $[\bar{E},\ ] =$ **Algorithm1**$(G, (\lambda_i^{(k)})_i, \mathcal{R}, \mathcal{R}_1, \mathcal{R})$.

4. Experiments

To solve the *VRPTW*, we use the Dantzig-Wolfe decomposition which defines K independent *sub-problems* and a global *master problem*. Applying a column generation, we alternatively solve the master problem and the K sub-problems. We addressed the VRPTW instances with a column generation approach by using the SPPRC as a sub-problem.

We used the Solomon data sets [59] and Homberger 200 customer instances. Each instance contains customers locations, resources (two for each costumer, i.e., $|\mathcal{R}| = 2$) and constraints. These data sets are classified in three categories: the r-instances (customers are located randomly), the c-instances (customers are located in clusters) and the rc-instances (mixed random and clustered structures). Furthermore, each family of instances is divided into two types. The first type (r1xx, c1xx, rc1xx) has a short scheduling horizon, i.e., small time windows, that allows only a few customers per route (approximately 5 to 10). The second type (r2xx, c2xx, rc2xx) has a long scheduling horizon permitting many customers (more than 30) to be serviced by the same vehicle. We applied our approximate DLC-SPPRC algorithm and the classical SPPRC [32].

We implemented our algorithm using Java programming language. For the simulation, we used a CPU Intel Core i9-9900KF (8 cores), 3.60 GHz, RAM 32 GB, running under Windows 10 (64 bit). For the SPPRC, we implemented the Desrochers and Soumis algorithm [32]. Linear programs for restricted master problems are solved with ILOG CPLEX 20.1.

Tables 2–5 report the iteration number (N_i), the lower bound (L_i^b), the computational time in seconds (T_i) and the number of generated columns (C_i), where $i = 1$ for the SPPRC and $i = 2$ for the DLC-SPPRC. A gap is computed as

$$Gap = 100 \times \frac{L_1^b - L_2^b}{L_1^b}. \tag{4.1}$$

Figure 1 shows the total number of generated columns.

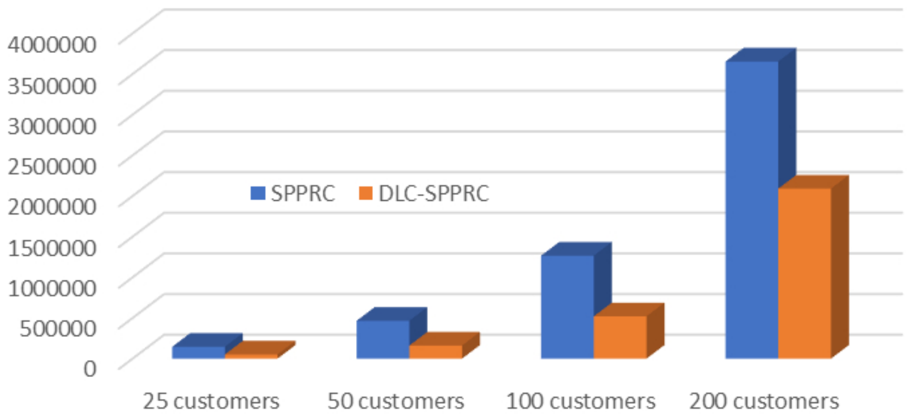

Figure 1. Number of generated columns for the Solomon and Homberger instances.

For most of the cases, we obtained the optimal solutions in reasonable times. As in [34], the column generation process is initiated with an adaptation of the Clarke and Wright algorithm [60]. The sub-problem is stopped at each iteration when 500 labels have been extended to the depot with a negative cost.

Table 2. Comparison of two approaches for solving the VRPTW for Solomon's instances with 25 customers: approximate DLC-SPPRC algorithm (2) and the classical SPPRC [32].

	SPPRC				DLC-SPPRC				Comparison	
Instance	L_1^b	T_1	N_1	C_1	L_2^b	T_2	N_2	C_2	Gap	T_1/T_2
c101	191.3	0.1163	33	579	191.3	0.1045	69	396	0.00	1.11
c102	189.2	0.285	40	1452	189.175	0.2817	82	976	0.01	1.01
c103	187.857	0.4005	43	1578	183.214	0.3721	66	807	2.47	1.08
c104	184.339	0.4012	45	1896	183.457	0.2618	44	787	0.48	1.53
c105	191.3	0.0651	36	718	191.3	0.0319	48	368	0.00	2.04
c106	191.3	0.0438	38	698	191.3	0.0372	73	386	0.00	1.18
c107	191.3	0.0476	27	749	191.3	0.0397	48	416	0.00	1.20
c108	187.84	0.1046	30	1090	185.809	0.0734	41	564	1.08	1.43
c109	181.927	0.1442	29	1132	185.314	0.0752	32	578	-1.86	1.92
c201	214.7	0.2174	89	2240	214.7	0.1531	237	1019	0.00	1.42
c202	214.7	0.3528	86	2637	214.7	0.2835	175	1266	0.00	1.24
c203	213.775	2.0617	76	3820	213.775	1.0133	153	1562	0.00	2.03
c204	207.174	4.3672	80	4880	207.172	2.0133	125	2083	0.00	2.17
c205	196.525	0.2943	43	3064	196.525	0.1322	88	1089	0.00	2.23
c206	194.018	0.5541	48	3472	194.015	0.2823	85	1875	0.00	1.96
c207	200.443	1.6901	67	4840	200.442	0.5865	84	2237	0.00	2.88
c208	183.863	1.0371	50	4124	183.861	0.3695	69	1751	0.00	2.81
r101	617.1	0.0065	10	108	617.1	0.0053	17	73	0.00	1.23
r102	546.333	0.0172	13	314	546.333	0.017	29	201	0.00	1.01
r103	454.067	0.0464	22	538	454.067	0.0342	35	280	0.00	1.36
r104	414.85	0.0715	25	735	414.85	0.0544	39	374	0.00	1.31
r105	530.5	0.014	19	254	530.5	0.011	25	145	0.00	1.27
r106	457.3	0.0439	22	543	457.3	0.023	27	265	0.00	1.91
r107	415.125	0.0573	25	618	415.125	0.0446	35	329	0.00	1.28
r108	389.424	0.0846	22	796	389.424	0.0623	32	422	0.00	1.36
r109	439.425	0.0247	19	341	439.425	0.0215	30	243	0.00	1.15
r110	419.072	0.0402	19	502	419.072	0.034	31	331	0.00	1.18
r111	412.815	0.044	19	554	412.815	0.036	31	302	0.00	1.22
r112	365.03	0.0994	27	818	365.03	0.07	35	464	0.00	1.42
r201	448.5	0.0706	16	877	448.5	0.0468	43	510	0.00	1.51
r202	374.092	0.1505	28	1479	374.092	0 122	53	754	0.00	1.23
r203	337.517	0.4222	37	2044	337.51	0.2901	62	897	0.00	1.46
r204	303.991	1.737	51	3900	303.993	0.615	74	1454	0.00	2.82
r205	365.475	0.2141	31	1744	365.475	0.1885	74	1043	0.00	1.14
r206	317.984	1.2275	42	3775	317.983	0.3397	59	1119	0.00	3.61
r207	309.623	3.0179	53	3413	309.612	0.7511	67	1377	0.00	4.02
r208	291.203	51.794	88	9718	291.217	3.4682	100	2952	0.00	14.93
r209	327.256	0.6002	34	2590	327.258	0.2808	53	1133	0.00	2.14
r210	340.505	0.3063	31	2033	340.505	0.182	49	875	0.00	1.68
r211	299.456	7.8134	57	5667	299.46	1.5388	71	1590	0.00	5.08
rc101	390.15	0.0187	18	284	390.15	0.0178	31	203	0.00	1.05
rc102	347.08	0.0591	20	467	347.08	0.0489	32	284	0.00	1.21
rc103	313.976	0.1227	21	751	309.907	0.08	25	330	1.30	1.53
rc104	287.539	0.255	29	888	292.458	0.1133	22	326	-1.71	2.25
rc105	408.525	0.0433	17	516	408.525	0.0299	32	202	0.00	1.45
rc106	314.274	0.0795	23	660	306.044	0.0618	33	343	2.62	1.29
rc107	281.289	0.258	31	1043	274.389	0.1383	33	460	2.45	1.87
rc108	270.573	0.4552	35	939	250.412	0.2489	40	596	7.45	1.83
rc201	315.853	0.1049	30	1325	315.853	0.0673	59	464	0.00	1.56
rc202	233.586	7.4873	57	3545	233.588	1.404	68	1043	0.00	5.33
rc203	178.264	31.379	93	8566	178.283	2.9895	77	1756	-0.01	10.50
rc204	153.909	90.1709	141	12217	153.554	4.7891	107	2803	0.23	18.83
rc205	258.915	0.3965	30	1960	258.915	0.1849	50	562	0.00	2.14
rc206	188.287	4.0061	71	6013	188.249	0.7239	70	1096	0.02	5.53
rc207	169.161	11.7082	79	7492	169.19	0.9203	68	1812	-0.02	12.72
rc208	138.287	145.445	163	16490	139.544	4.3513	121	3469	-0.91	33.43

Table 3. Comparison of two approaches for solving the VRPTW for Solomon's instances with 50 customers: approximate DLC-SPPRC algorithm (2) and the classical SPPRC [32].

	SPPRC				DLC-SPPRC				Comparison	
Instance	L_1^b	T_1	N_1	C_1	L_2^b	T_2	N_2	C_2	Gap	T_1/T_2
c101	362.4	0.4482	60	1527	362.4	0.4382	155	1052	0.00	1.02
c102	360.25	1.2188	83	2922	360.275	0.9699	136	1792	-0.01	1.26
c103	360.25	2.5992	106	3599	353.341	1.7055	182	2264	1.92	1.52
c104	352.266	6.1074	119	5660	346.952	2.8453	127	2480	1.51	2.15
c105	362.4	0.5268	81	2172	361.2	0.3185	166	1259	0.33	1.65
c106	362.4	0.3838	88	2169	362.4	0.2125	125	973	0.00	1.81
c107	362.4	0.4258	73	2435	361.2	0.3272	150	1433	0.33	1.30
c108	359.81	0.6627	62	2595	356.31	0.539	106	1469	0.97	1.23
c109	354.316	0.9185	57	2784	340.938	0.6901	94	1528	3.78	1.33
c201	360.2	6.6625	213	13828	360.2	1.6169	489	3335	0.00	4.12
c202	360.2	8.1671	200	12216	360.2	2.5508	259	3440	0.00	3.20
c203	359.8	38.6973	252	15108	359.8	14.1948	333	5850	0.00	2.73
c204	347.404	114.6872	243	18375	347.679	35.7323	320	8201	-0.08	3.21
c205	341.763	9.5662	145	13681	341.769	3.5832	223	4713	0.00	2.67
c206	338.519	19.4161	180	15798	338.379	4.8996	222	5187	0.04	3.96
c207	348.631	38.0489	203	18358	348.613	7.9878	235	7176	0.01	4.76
c208	331.276	20.6723	180	15147	331.278	3.7519	174	4892	0.00	5.51
r101	1043.367	0.0447	23	463	1043.367	0.0421	44	276	0.00	1.06
r102	909	0.1435	29	987	909	0.1272	54	534	0.00	1.13
r103	756.117	0.3896	43	1468	756.117	0.293	59	790	0.00	1.33
r104	608.521	1.2382	66	2600	608.491	0.8123	87	1255	0.00	1.52
r105	890.187	0.0958	26	821	890.187	0.0834	43	395	0.00	1.15
r106	789.433	0.3278	38	1255	789.433	0.2506	59	690	0.00	1.31
r107	697.767	0.504	46	1793	697.767	0.3672	60	855	0.00	1.37
r108	578.482	1.6057	68	2761	578.482	1.2553	106	1613	0.00	1.28
r109	727.515	0.2537	36	1403	727.515	0.1712	53	614	0.00	1.48
r110	675.457	0.4694	42	1578	675.246	0.3234	59	761	0.03	1.45
r111	658.752	0.592	47	1927	658.752	0.4077	65	913	0.00	1.45
r112	582.715	1.3422	63	2502	582.611	0.7894	82	1170	0.02	1.70
r201	754.098	0.5013	42	2332	754.098	0.4006	91	1243	0.00	1.25
r202	637.556	1.9978	62	4231	637.562	1.1956	106	1717	0.00	1.67
r203	538.526	5.4656	78	4791	538.529	2.7795	140	2696	0.00	1.97
r204	441.005	329.0178	229	23629	441.021	28.7788	272	7887	0.00	11.43
r205	595.538	2.0053	66	4505	595.55	1.0479	115	1970	0.00	1.91
r206	530.527	8.3613	87	7378	530.512	3.2432	142	3182	0.00	2.58
r207	467.282	34.0113	145	11323	467.287	7.7842	177	4139	0.00	4.37
r208	423.781	2234.4085	302	34624	423.799	120.1348	309	9984	0.00	18.60
r209	535.282	8.0839	85	6534	535.284	2.9226	139	2811	0.00	2.77
r210	532.1	5.5001	86	6296	532.096	2.3795	135	2803	0.00	2.31
r211	457.426	31.1272	129	10024	457.43	6.4031	172	4082	0.00	4.86
rc101	826.613	0.1158	31	716	826.613	0.0998	47	428	0.00	1.16
rc102	706.606	0.4957	47	1160	706.556	0.3879	62	652	0.01	1.28
rc103	612.24	1.2763	58	1756	605.145	0.9114	77	1008	1.16	1.40
rc104	524.119	4.7657	88	3146	511.583	1.7867	86	1417	2.39	2.67
rc105	746.314	0.2826	38	1091	745.928	0.2183	52	625	0.05	1.29
rc106	633.228	0.5801	43	1407	628.517	0.4113	53	741	0.74	1.41
rc107	570.669	2.1353	73	2734	561.863	1.1098	87	1273	1.54	1.92
rc108	525.123	3.43	79	2630	501.695	1.984	96	1477	4.46	1.73
rc201	530.505	1.1744	64	3001	530.505	0.5838	79	1015	0.00	2.01
rc202	416.517	25.8833	132	8997	416.518	5.6157	147	2801	0.00	4.61
rc203	326.285	216.7017	264	20712	326.294	15.1477	184	5153	0.00	14.31
rc204	263.83	18,964.6011	431	47037	264.301	70.3698	333	12067	-0.18	269.50
rc205	481.609	6.5427	96	7435	481.621	2.5118	126	2633	0.00	2.60
rc206	363.888	56.1137	189	15075	363.892	7.3344	173	3830	0.00	7.65
rc207	333.071	282.0786	239	23198	333.071	18.2862	196	5016	0.00	15.43
rc208	265.107	1974.0373	456	41459	264.912	86.3844	399	10,952	0.07	22.85

Table 4. Comparison of two approaches for solving the VRPTW for Solomon's instances with 100 customers: approximate DLC-SPPRC algorithm (2) and the classical SPPRC [32].

	SPPRC				DLC-SPPRC				Comparison	
Instance	L_1^b	T_1	N_1	C_1	L_2^b	T_2	N_2	C_2	Gap	T_1/T_2
c101	827.3	2.2196	122	3553	827.3	2.4001	383	2852	0.00	0.92
c102	827.3	9.7928	191	7359	820.3	7.3079	277	4279	0.85	1.34
c103	826.3	21.8133	223	9480	817.879	17.6085	344	5663	1.02	1.24
c104	821.597	55.5135	327	13285	793.653	47.0154	544	8789	3.40	1.18
c105	827.3	2.9878	142	4392	821.2	2.9093	308	3648	0.74	1.03
c106	827.3	4.0281	134	4928	827.4	3.5591	258	3438	-0.01	1.13
c107	827.3	3.0508	132	4626	819.6	4.128	331	4435	0.93	0.74
c108	817.352	6.5019	160	6311	801.696	5.2622	219	3318	1.92	1.24
c109	809.265	12.6714	231	9430	778.527	7.4906	256	4669	3.80	1.69
c201	589.1	44.9491	418	28845	589.1	11.6671	528	6526	0.00	3.85
c202	589.1	179.6573	552	44694	589.1	61.4668	722	12,701	0.00	2.92
c203	585.767	1223.024	995	75,865	585.767	446.2332	1488	32,852	0.00	2.74
c204	582.383	4893.6566	1630	136,556	582.226	990.1879	2056	58,070	0.03	4.94
c205	582.369	151.9166	472	42,390	582.363	44.6547	617	15,218	0.00	3.40
c206	575.993	346.5503	614	56,564	575.845	61.8254	652	16,738	0.03	5.61
c207	570.524	367.8769	544	52,839	570.52	74.188	691	18,018	0.00	4.96
c208	570.255	423.4152	635	59,277	570.278	65.6802	614	17,867	0.00	6.45
r101	1631.15	0.416	44	1458	1631.15	0.3951	92	829	0.00	1.05
r102	1466.6	1.5553	63	2627	1466.6	1.5157	126	1469	0.00	1.03
r103	1203.241	5.3805	104	4549	1203.2	3.7518	154	2123	0.00	1.43
r104	937.062	28.4824	190	9368	936.742	22.0167	310	4715	0.03	1.29
r105	1341.194	1.2316	64	2714	1341.194	0.8896	92	1270	0.00	1.38
r106	1212.355	6.0697	116	4719	1212.312	3.0527	143	2155	0.00	1.99
r107	1036.956	13.5413	148	6645	1037.263	7.2842	192	3094	-0.03	1.86
r108	891.646	44.4008	249	12269	890.074	24.6322	336	5525	0.18	1.80
r109	1097.456	4.8182	101	4941	1097.343	2.2239	114	1915	0.01	2.17
r110	1021.333	11.5028	135	6226	1021.138	5.6338	171	2783	0.02	2.04
r111	1006.032	12.7601	153	7009	1005.671	7.4647	200	3273	0.04	1.71
r112	892.576	35.3526	201	9372	887.714	18.0717	283	4819	0.54	1.96
r201	1080.749	9.8496	119	9102	1080.771	6.4248	231	3947	0.00	1.53
r202	933.446	93.7052	241	15273	933.458	23.7589	353	6624	0.00	3.94
r203	756.739	451.4435	402	28,667	756.731	87.1933	565	11,958	0.00	5.18
r204	640.238	7638.2746	638	56,850	640.272	559.2558	863	22660	-0.01	13.66
r205	838.773	76.568	242	19,494	838.772	29.6386	372	8314	0.00	2.58
r206	749.068	474.0996	370	28,002	749.066	84.8317	538	12161	0.00	5.59
r207	668.711	3609.1966	568	50,978	668.711	281.5144	703	19,200	0.00	12.82
r208					610.278	1197.6882	1001	31,544		0.00
r209	750.455	445.0778	292	28162	750.454	82.0649	484	11,179	0.00	5.42
r210	753.985	205.3728	309	23,599	753.993	63.3126	494	11,025	0.00	3.24
r211	650.834	1793.735	573	47,048	650.845	167.9773	741	17,532	0.00	10.68
rc101	1567.449	1.0734	61	2250	1567.282	0.8967	92	1159	0.01	1.20
rc102	1380.209	2.9682	84	3319	1378.326	2.6881	124	1776	0.14	1.10
rc103	1170.318	12.2669	155	5901	1164.407	7.3091	177	2656	0.51	1.68
rc104	1052.551	38.1889	190	8215	1027.508	19.5054	250	4057	2.38	1.96
rc105	1453.894	2.213	72	2938	1453.147	1.6459	105	1565	0.05	1.34
rc106	1248.958	4.2376	97	3982	1242.824	2.694	124	1847	0.49	1.57
rc107	1117.371	12.9665	146	5703	1096.177	7.1789	179	2847	1.90	1.81
rc108	1035.93	32.3093	198	7553	1019.784	13.1855	225	3831	1.56	2.45
rc201	1107.012	13.2285	149	9682	1107.011	6.554	227	3656	0.00	2.02
rc202	880.343	127.7105	265	1, 7952	880.329	29.5938	339	6574	0.00	4.32
rc203	693.53	902.8656	475	3,5439	693.1	146.6249	591	13829	0.06	6.16
rc204	607.663	14,066.2658	815	7,8145	606.758	587.2997	854	25540	0.15	23.95
rc205	967.105	59.4491	230	15595	967.097	19.7616	304	5936	0.00	3.01
rc206	852.167	130.5895	305	24,025	852.178	30.1325	348	7607	0.00	4.33
rc207	767.951	567.5535	417	29,098	767.982	80.8772	456	10,815	0.00	7.02
rc208	627.276	3223.8542	638	44904	627.155	223.5683	726	17,074	0.02	14.42

Table 5. Comparison of two approaches for solving the VRPTW for Homberger's instances with 200 customers: approximate DLC-SPPRC algorithm (2) and the classical SPPRC [32].

	SPPRC				DLC-SPPRC				Comparison	
Instance	L_1^b	T_1	N_1	C_1	L_2^b	T_2	N_2	C_2	Gap	T_1/T_2
$c_{1\_2\_1}$	2698.6	21.6988	258	9936	2694.3	18.046	422	6025	0.2	1.20
$c_{1\_2\_2}$	2681.19	268.0735	636	19,085	2674.933	136.1003	732	10,144	0.2	1.97
$c_{1\_2\_3}$	2619.191	1091.5661	855	25,103	2718.196	364.9952	940	14,092	-3.8	2.99
$c_{1\_2\_4}$	2504.595	2681.358	1096	33,006	3018.105	297.099	1045	15,950	-20.5	9.03
$c_{1\_2\_5}$	2694.9	40.977	347	13,060	2693.35	30.2351	490	8212	0.1	1.36
$c_{1\_2\_6}$	2694.9	56.4944	354	13,018	2691.1	39.2523	481	7749	0.1	1.44
$c_{1\_2\_7}$	2694.9	57.3833	404	14,065	2690.6	40.9054	513	9006	0.2	1.40
$c_{1\_2\_8}$	2595.243	157.6131	506	17311	2570.971	65.4297	458	8383	0.9	2.41
$c_{1\_2\_9}$	2512.459	336.6065	662	21,390	2468.483	134.5933	667	11,797	1.8	2.50
$c_{1\_2\_10}$	2456.881	849.7886	743	23,092	2418.042	273.4165	829	14281	1.6	3.11
$c_{2\_2\_1}$	1915.035	586.6113	1042	70,794	1915.035	142.9373	1122	17930	0.0	4.10
$c_{2\_2\_2}$	1830.676	11,210.2029	2147	133,228	1830.262	3136.7113	2505	41,482	0.0	3.57
$c_{2\_2\_3}$	1703.946	32,211.3313	2903	168,434	1704.568	7460.8854	3318	55,403	0.0	4.32
$c_{2\_2\_4}$	1556.237	134,817.7847	4424	294,220	1555.633	27,365.9701	5969	132,146	0.0	4.93
$c_{2\_2\_5}$	1759.722	1413.2772	997	67,540	1759.242	368.3524	1098	22,931	0.0	3.84
$c_{2\_2\_6}$	1681.695	3003.6015	1155	79,219	1681.713	742.8865	1250	29,383	0.0	4.04
$c_{2\_2\_7}$	1709.103	3694.6209	1506	100,137	1709.081	906.8754	1533	34,220	0.0	4.07
$c_{2\_2\_8}$	1626.857	7029.1749	1330	98,417	1626.946	1292.0355	1445	37,099	0.0	5.44
$c_{2\_2\_9}$	1635.193	11,642.7703	1584	108,311	1634.903	2111.9977	1840	43,214	0.0	5.51
$c_{2\_2\_10}$	1572.073	13,937.5185	1602	113,454	1571.286	2543.9178	1752	46,140	0.1	5.48
$r_{1\_2\_1}$	4654.913	26.9629	199	10,342	4651.425	16.4504	363	4431	0.1	1.64
$r_{1\_2\_2}$	3905.961	338.0128	445	22,615	3908.449	110.3577	581	8513	-0.1	3.06
$r_{1\_2\_3}$	3282.964	2155.3814	676	31,966	3528.17	216.1976	859	12,419	-7.5	9.97
$r_{1\_2\_4}$	2925.629	4974.6947	1081	56,343	3749.5	83.8418	678	11,857	-28.2	59.33
$r_{1\_2\_5}$	3978.831	68.9078	261	14,902	3945.239	26.7507	321	5028	0.8	2.58
$r_{1\_2\_6}$	3383.356	605.1315	458	23,455	3442.907	110.0345	539	8589	-1.8	5.50
$r_{1\_2\_7}$	2971.763	2556.5618	672	31,896	3462.465	128.5044	654	10,654	-16.5	19.89
$r_{1\_2\_8}$	2777.372	5019.4831	1357	74,400	3701.82	135.8771	864	15,700	-33.3	36.94
$r_{1\_2\_9}$	3592.027	134.3127	300	17795	3594.651	35.4428	310	5635	-0.1	3.79
$r_{1\_2\_10}$	3080.993	611.0459	529	28032	3149.732	86.7837	544	9846	-2.2	7.04
$r_{2\_2\_1}$	3284.637	252.7638	500	38,393	3284.626	116.7495	803	12,874	0.0	2.17
$r_{2\_2\_2}$	2438.04	5482.703	1069	82,918	2438.042	964.8113	1512	26,275	0.0	5.68
$r_{2\_2\_3}$					1851.098	10,488.4304	3301	60,935		
$r_{2\_2\_4}$					2181.11	4187.5913	5641	134,920		
$r_{2\_2\_5}$	2702.26	1143.0031	784	65,349	2702.254	314.7569	1007	20,114	0.0	3.63
$r_{2\_2\_6}$					2051.403	4754.4525	2050	41,661		
$r_{2\_2\_7}$					1781.771	6469.1145	3047	68,141		
$r_{2\_2\_8}$					1524.816	25,399.6232	6141	151,513		
$r_{2\_2\_9}$	2372.809	4375.8383	1069	92,700	2372.895	872.3769	1425	28,674	0.0	5.02
$r_{2\_2\_10}$	2057.437	51,478.3753	1533	135,588	2054.675	2828.0145	1790	35,968	0.1	18.20
$rc_{1\_2\_1}$	3408.547	77.1215	254	14,080	3335.566	28.9713	251	4772	2.1	2.66
$rc_{1\_2\_2}$	3079.317	701.3758	477	22,792	3064.343	170.516	610	10,268	0.5	4.11
$rc_{1\_2\_3}$	2851.48	2135.8783	649	31,119	3171.093	203.1931	745	12,443	-11.2	10.51
$rc_{1\_2\_4}$	2704.62	3337.8092	1016	44,426	3499.367	94.3266	670	11,472	-29.4	35.39
$rc_{1\_2\_5}$	3175.998	371.7829	291	18201	3040.87	108.5267	353	7519	4.3	3.43
$rc_{1\_2\_6}$	3137.163	276.9911	296	18,063	3060.842	69.2983	308	6950	2.4	4.00
$rc_{1\_2\_7}$	3002.62	686.0828	351	22,094	2798.384	225.8202	458	9610	6.8	3.04
$rc_{1\_2\_8}$	2889.813	982.34	453	26,034	3005.031	143.7637	526	11,799	-4.0	6.83
$rc_{1\_2\_9}$	2870.361	985.5621	427	25,897	2906.32	161.3079	474	10,730	-1.3	6.11
$rc_{1\_2\_10}$	2797.848	1378.7379	516	30,122	2765.18	261.1241	658	14,439	1.2	5.28
$rc_{2\_2\_1}$	2418.139	1220.3385	636	59,964	2418.217	408.3209	1094	16,622	0.0	2.99
$rc_{2\_2\_2}$					1910.279	4305.9053	2020	35,924		
$rc_{2\_2\_3}$					2248.714	1358.3973	1925	42,749		
$rc_{2\_2\_4}$					2010.182	12,228.8797	9810	314,665		
$rc_{2\_2\_5}$					2159.079	2153.8087	1429	40,285		
$rc_{2\_2\_6}$					2085.187	2778.6479	1376	35,622		
$rc_{2\_2\_7}$					1927.925	4422.8466	1680	51,136		
$rc_{2\_2\_8}$					1902.738	3276.8221	1726	63,367		
$rc_{2\_2\_9}$					1860.267	3542.0999	1719	64,506		
$rc_{2\_2\_10}$					1875.748	3580.6746	2317	86,154		

From Tables 2–5, we can draw the following conclusions. Our algorithm was faster in all cases (228). In 15 instances (7%), the SPPRC algorithm could not obtain a solution in reasonable time (more than 10 hours for one iteration). Our algorithm was 10 times faster in 37 cases (16% of the cases) and 1.5 times faster in 166 case (73% of the cases). On average, we achieved a gain of 564% of the computational time. For the precision, in 188 cases (82%), we obtained similar precisions (a gap less than 1%) whereas the difference was more than 1% only in 25 cases (11%) and greater than 5% only in 2 cases (0.88%).

5. Conclusions

This paper proposes an approximate solution algorithm to solve the SPPRC, which relies on a Lagrangian relaxation. Our method applies to both acyclic and cyclic graphs. Dominance is applied on a subset of the resources only. Optimized parameters updates schemes are used to ensure fast convergence. When applied to the VRP, our approach offers a good compromise between precision and computational time, and thus, can be applied to large-scale practical problems. In the future, we plan to apply our algorithm to solve various problems such as branch-and-price, branch-and-cut and branch-and-price-and-cut. In addition, we will explore more advanced optimization algorithms (heuristics, meta-heuristics, etc.) that have been successfully applied in other domains, such as, online learning, scheduling, multi-objective optimization, transportation, medicine, and data classification.

Conflict of interest

The authors declare that there is no conflicts of interest.

References

1. G. B. Dantzig, J. H. Ramser, The truck dispatching problem, *Manage. Sci.*, **6** (1959), 80–91. https://doi.org/10.1287/mnsc.6.1.80
2. M. Desrochers, J. Desrosiers, M. Solomon, A new optimization algorithm for the vehicle routing problem with time windows, *Oper. Res.*, **40** (1982), 342–354. https://doi.org/10.1287/opre.40.2.342
3. J. Desrosiers, Y. Dumas, M. M. Solomon, F. Soumis, Time constrained routing and scheduling, *Handbooks Oper. Res. Manage. Sci.*, **8** (1995), 35–139. https://doi.org/10.1016/S0927-0507(05)80106-9
4. N. Kohl, J. Desrosiers, O. B. Madsen, M. M. Solomon, F. Soumis, 2-path cuts for the vehicle routing problem with time windows, *Transp. Sci.*, **33** (1999), 101–116. https://doi.org/10.1287/trsc.33.1.101
5. S. Irnich, D. Villeneuve, The shortest-path problem with resource constraints and k-cycle elimination for $k \geq 3$, *INFORMS J. Comput.*, **18** (2006), 391–406. https://doi.org/10.1287/ijoc.1040.0117
6. R. Sadykov, E. Uchoa, A. Pessoa, A bucket graph-based labeling algorithm with application to vehicle routing, *Transp. Sci.*, **55** (2021), 4–28. https://doi.org/10.1287/trsc.2020.0985

7. R. Baldacci, E. Bartolini, A. Mingozzi, R. Roberti, An exact solution framework for a broad class of vehicle routing problems, *Comput. Manage. Sci.*, **7** (2010), 229–268. https://doi.org/10.1007/s10287-009-0118-3

8. R. Fukasawa, Q. He, Y. Song, A branch-cut-and-price algorithm for the energy minimization vehicle routing problem, *Transp. Sci.*, **50** (2016), 23–34. https://doi.org/10.1287/trsc.2015.0593

9. D. Pecin, A. Pessoa, M. Poggi, E. Uchoa, Improved branch-cut-and-price for capacitated vehicle routing, *Math. Program. Comput.*, **9** (2017), 61–100. https://doi.org/10.1007/s12532-016-0108-8

10. R. Fukasawa, H. Longo, J. Lysgaard, M. P. D. Aragão, M. Reis, E. Uchoa, et al., Robust branch-and-cut-and-price for the capacitated vehicle routing problem, *Math. Program.*, **106** (2006), 491–511. https://doi.org/10.1007/s10107-005-0644-x

11. R. Baldacci, N. Christofides, A. Mingozzi, An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts, *Math. Program.*, **115** (2008), 351–385. https://doi.org/10.1007/s10107-007-0178-5

12. R. Baldacci, A. Mingozzi, R. Roberti, New route relaxation and pricing strategies for the vehicle routing problem, *Oper. Res.*, **59** (2011), 1269–1283. https://doi.org/10.1287/opre.1110.0975

13. S. Dabia, S. Ropke, T. V. Woensel, T. D. Kok, Branch and price for the time-dependent vehicle routing problem with time windows, *Transp. Sci.*, **47** (2013), 380–396. https://doi.org/10.1287/trsc.1120.0445

14. A. Nagih, F. Soumis, Nodal aggregation of resource constraints in a shortest path problem, *Eur. J. Oper. Res.*, **172** (2006), 500–514. https://doi.org/10.1016/j.ejor.2004.09.052

15. I. Himmich, H. B. Amor, I. E. Hallaoui, F. Soumis, A primal adjacency-based algorithm for the shortest path problem with resource constraints, *Transp. Sci.*, **54** (2020), 1153–1169. https://doi.org/10.1287/trsc.2019.0941

16. M. Behnke, T. Kirschstein, C. Bierwirth, A column generation approach for an emission-oriented vehicle routing problem on a multigraph, *Eur. J. Oper. Res.*, **288** (2021), 794–809. https://doi.org/10.1016/j.ejor.2020.06.035

17. I. Mathlouthi, M. Gendreau, J. Y. Potvin, Branch-and-price for a multi-attribute technician routing and scheduling problem, in *Operations Research Forum*, Springer International Publishing, **2** (2021), 1–35. https://doi.org/10.1007/s43069-020-00044-x

18. S. Y. Tan, W. C. Yeh, The vehicle routing problem: State-of-the-art classification and review, *Appl. Sci.*, **11** (2021), 10295. https://doi.org/10.3390/app112110295

19. P. Toth, D. Vigo, *Vehicle Routing: Problems, Methods, and Applications*, SIAM, 2014.

20. L. Taccari, Integer programming formulations for the elementary shortest path problem, *Eur. J. Oper. Res.*, **252** (2016), 122–130. https://doi.org/10.1016/j.ejor.2016.01.003

21. E. Manousakis, P. Repoussis, E. Zachariadis, C. Tarantilis, Improved branch-and-cut for the inventory routing problem based on a two-commodity flow formulation, *Eur. J. Oper. Res.*, **290** (2021), 870–885. https://doi.org/10.1016/j.ejor.2020.08.047

22. G. Lera-Romero, J. J. Miranda-Bront, A branch and cut algorithm for the time-dependent profitable tour problem with resource constraints, *Eur. J. Oper. Res.*, **289** (2021), 879–896. https://doi.org/10.1016/j.ejor.2019.07.014

23. C. M. Damião, J. M. P. Silva, E. Uchoa, A branch-cut-and-price algorithm for the cumulative capacitated vehicle routing problem, *4OR-Q. J. Oper. Res.*, **2021** (2021), 1–25. https://doi.org/10.1007/s10288-021-00498-7

24. H. B. Ticha, N. Absi, D. Feillet, A. Quilliot, Empirical analysis for the VRPTW with a multigraph representation for the road network, *Comput. Oper. Res.*, **88** (2017), 103–116. https://doi.org/10.1016/j.cor.2017.06.024

25. H. B. Ticha, N. Absi, D. Feillet, A. Quilliot, Vehicle routing problems with road-network information: State of the art, *Networks*, **72** (2018), 393–406. https://doi.org/10.1002/net.21808

26. H. B. Ticha, N. Absi, D. Feillet, A. Quilliot, T. V. Woensel, A branch-and-price algorithm for the vehicle routing problem with time windows on a road network, *Networks*, **73** (2019), 401–417. https://doi.org/10.1002/net.21852

27. C. Archetti, M. G. Speranza, A survey on matheuristics for routing problems, *EURO J. Comput. Optim.*, **2** (2014), 223–246. https://doi.org/10.1007/s13675-014-0030-7

28. D. Pecin, C. Contardo, G. Desaulniers, E. Uchoa, New enhancements for the exact solution of the vehicle routing problem with time windows, *INFORMS J. Comput.*, **29** (2017), 489–502. https://doi.org/10.1287/ijoc.2016.0744

29. G. Desaulniers, J. Desrosiers, M. M. Solomon, *Column Generation*, Springer Science & Business Media, 2006.

30. M. E. Lübbecke, J. Desrosiers, Selected topics in column generation, *Oper. Res.*, **53** (2005), 1007–1023. https://doi.org/10.1287/opre.1050.0234

31. M. Desrochers, *La fabrication d'horaires de travail pour les conducteurs d'autobus par une méthode de génération de colonnes*, Université de Montréal, Centre de recherche sur les transports, 1986.

32. M. Desrochers, F. Soumis, A reoptimization algorithm for the shortest path problem with time windows, *Eur. J. Oper. Res.*, **35** (1988), 242–254. https://doi.org/10.1016/0377-2217(88)90034-3

33. G. Desaulniers, D. Villeneuve, The shortest path problem with time windows and linear waiting costs, *Transp. Sci.*, **34** (2000), 312–319. https://doi.org/10.1287/trsc.34.3.312.12298

34. D. Feillet, P. Dejax, M. Gendreau, C. Gueguen, An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems, *Networks: Int. J.*, **44** (2004), 216–229. https://doi.org/10.1002/net.20033

35. G. Righini, M. Salani, Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints, *Discrete Optim.*, **3** (2006), 255–273. https://doi.org/10.1016/j.disopt.2006.05.007

36. A. Chabrier, Vehicle routing problem with elementary shortest path based column generation, *Comput. Oper. Res.*, **33** (2006), 2972–2990. https://doi.org/10.1016/j.cor.2005.02.029

37. D. Feillet, M. Gendreau, L. M. Rousseau, New refinements for the solution of vehicle routing problems with branch and price, *INFOR: Inf. Syst. Oper. Res.*, **45** (2007), 239–256. https://doi.org/10.3138/infor.45.4.239

38. M. Tagmouti, M. Gendreau, J. Y. Potvin, Arc routing problems with time-dependent service costs, *Eur. J. Oper. Res.*, **181** (2007), 30–39. https://doi.org/10.1016/j.ejor.2006.06.028

39. M. Jepsen, B. Petersen, S. Spoorendonk, D. Pisinger, Subset-row inequalities applied to the vehicle-routing problem with time windows, *Oper. Res.*, **56** (2008), 497–511. https://doi.org/10.1287/opre.1070.0449

40. G. Righini, M. Salani, New dynamic programming algorithms for the resource constrained elementary shortest path problem, *Networks: Int. J.*, **51** (2008), 155–170. https://doi.org/10.1002/net.20212

41. G. Desaulniers, F. Lessard, A. Hadjar, Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows, *Transp. Sci.*, **42** (2008), 387–404. https://doi.org/10.1287/trsc.1070.0223

42. A. Qureshi, E. Taniguchi, T. Yamada, An exact solution approach for vehicle routing and scheduling problems with soft time windows, *Transp. Res. Part E Logist. Transp. Rev.*, **45** (2009), 960–977. https://doi.org/10.1016/j.tre.2009.04.007

43. A. Bettinelli, A. Ceselli, G. Righini, A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows, *Transp. Res. Part C Emerging Technol.*, **19** (2011), 723–740. https://doi.org/10.1016/j.trc.2010.07.008

44. F. Liberatore, G. Righini, M. Salani, A column generation algorithm for the vehicle routing problem with soft time windows, *4OR*, **9** (2011), 49–82. https://doi.org/10.1007/s10288-010-0136-6

45. D. Duque, L. Lozano, A. L. Medaglia, Solving the orienteering problem with time windows via the pulse framework, *Comput. Oper. Res.*, **54** (2015), 168–176. https://doi.org/10.1016/j.cor.2014.08.019

46. L. Lozano, D. Duque, A. L. Medaglia, An exact algorithm for the elementary shortest path problem with resource constraints, *Transp. Sci.*, **50** (2016), 348–357. https://doi.org/10.1287/trsc.2014.0582

47. G. Lera-Romero, J. J. Miranda-Bront, Integer programming formulations for the time-dependent elementary shortest path problem with resource constraints, *Electron. Notes Discrete Math.*, **69** (2018), 53–60. https://doi.org/10.1016/j.endm.2018.07.008

48. K. Dalmeijer, G. Desaulniers, Addressing orientation symmetry in the time window assignment vehicle routing problem, *INFORMS J. Comput.*, **33** (2021), 495–510. https://doi.org/10.1287/ijoc.2020.0974

49. D. Taş, Electric vehicle routing with flexible time windows: a column generation solution approach, *Transp. Lett.*, **13** (2021), 97–103. https://doi.org/10.1080/19427867.2020.1711581

50. M. Gendreau, J. Y. Potvin, O. Bräumlaysy, G. Hasle, A. Løkketangen, Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography, in *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer US, Boston, MA, (2008), 143–169. https://doi.org/10.1007/978-0-387-77778-8_7

51. J. Pasha, A. L. Nwodu, A. M. Fathollahi-Fard, G. Tian, Z. Li, H. Wang, et al., Exact and metaheuristic algorithms for the vehicle routing problem with a factory-in-a-box in multi-objective settings, *Adv. Eng. Inf.*, **52** (2022), 101623. https://doi.org/10.1016/j.aei.2022.101623

52. H. Park, D. Son, B. Koo, B. Jeong, Waiting strategy for the vehicle routing problem with simultaneous pickup and delivery using genetic algorithm, *Expert Syst. Appl.*, **165** (2021), 113959. https://doi.org/10.1016/j.eswa.2020.113959
53. H. Fan, Y. Zhang, P. Tian, Y. Lv, H. Fan, Time-dependent multi-depot green vehicle routing problem with time windows considering temporal-spatial distance, *Comput. Oper. Res.*, **129** (2021), 105211. https://doi.org/10.1016/j.cor.2021.105211
54. G. Srivastava, A. Singh, R. Mallipeddi, Nsga-ii with objective-specific variation operators for multiobjective vehicle routing problem with time windows, *Expert Syst. Appl.*, **176** (2021), 114779. https://doi.org/10.1016/j.eswa.2021.114779
55. W. C. Yeh, S. Y. Tan, Simplified swarm optimization for the heterogeneous fleet vehicle routing problem with time-varying continuous speed function, *Electronics*, **10** (2021). https://doi.org/10.3390/electronics10151775
56. M. A. Nguyen, G. T. Dang, M. H. Hà, M. T. Pham, The min-cost parallel drone scheduling vehicle routing problem, *Eur. J. Oper. Res.*, **299** (2022), 910–930. https://doi.org/10.1016/j.ejor.2021.07.008
57. P. Sun, L. P. Veelenturf, S. Dabia, T. V. Woensel, The time-dependent capacitated profitable tour problem with time windows and precedence constraints, *Eur. J. Oper. Res.*, **264** (2018), 1058–1073. https://doi.org/10.1016/j.ejor.2017.07.004
58. P. Sun, L. P. Veelenturf, M. Hewitt, T. V. Woensel, The time-dependent pickup and delivery problem with time windows, *Transp. Res. Part B Methodol.*, **116** (2018), 1–24. https://doi.org/10.1016/j.trb.2018.07.002
59. M. M. Solomon, *Vehicle Routing and Scheduling with Time Window Constraints: Models and Algorithms (Heuristics)*, PhD thesis, University of Pennsylvania, 1984.
60. G. Clarke, J. W. Wright, Scheduling of vehicles from a central depot to a number of delivery points, *Oper. Res.*, **12** (1964), 568–581. https://doi.org/10.1287/opre.12.4.568