

Partial Path Column Generation for the Elementary Shortest Path Problem with Resource Constraints

Mads Kehlet Jepsen* & Bjørn Petersen*

**Department of Computer Science, University of Copenhagen
Universitetsparken 1, 2100 København Ø, Danmark*

Abstract

This paper introduces a decomposition of the Elementary Shortest Path Problem with Resource Constraints (ESPPRC), where the path is combined by smaller sub paths. We show computational results by comparing different approaches for the decomposition and compare the best of these with existing algorithms. We show that the algorithm for many instances outperforms a bidirectional labeling algorithm.

Keywords: *Elementary Shortest Path With Resource Constraints, Column Generation, Dantzig-Wolfe, Vehicle Routing Problem*

1 Introduction

A formal definition of the ESPPRC problem solved in this conference paper is as follows: Given a directed $G(V, A)$ with node set $V \{1, \dots, |V|\}$ and arc set A , a set of resources R each with a global upper bound W_u^r . For each edge $(i, j) \in E$, c_{ij} is the cost of the edge and w_{ij}^r is the resource consumption of the edge. A path p is feasible if $\sum_{(i,j) \in E(p)} w_{ij}^r \leq W_u^r$ for all $r \in R$. The objective is to find the feasible path p with the minimum cost $\sum_{(i,j) \in E(p)} c_{ij}$ from a source node $s \in V$ to a destination node $d \in V$.

When negative cycles are allowed, the ESPPRC can be shown to be NP-complete by reduction from the longest path problem. Beasley and Christofides [2] gave a mathematical formulation of the problem where each node is considered a resource. When the graph may contain negative cost cycles Feillet et al. [8] introduced a labeling, Righini and Salani [12] proposed a bi-directional labeling algorithm and a Branch and Bound algorithm, using a relaxation where cycles are allowed¹, Boland et al. [3] gave a label correcting algorithm and Baldacci et al. [1] computed lower bounds on paths from a node in the graph to the destination and used these to speed up a bi-directional labeling algorithm.

For ESPPRC where the graph is assumed to contain no negative cost cycles, known as the resource constraint shortest path problem (SPPRC), Beasley and Christofides [2] gave a Branch and Bound algorithm based on Lagrangian dual bounds, Dumitrescu and Boland [7] suggest improved preprocessing as well as several algorithms and Carlyle et al. [5] propose a Lagrangian approach, where paths with cost between the Lagrangian bounds and the current upper bound are found using the k shortest path algorithm by Carlyle and Wood [4]

The main application of ESPPRC is as the pricing problem, when solving the Vehicle Routing Problem through Branch and Cut and Price. Chabrier [6] and Jepsen et al. [11] has done this successfully for VRPTW and Baldacci et al. [1] has recently done it for CVRP.

Labeling algorithms has so far been used very successfully for ESPPRC problems especially when time windows are included as resources. However for instances where the time windows are very large the state space becomes huge and labeling algorithms are no longer a good practical solution approach.

¹see Irnich and Villeneuve [9] for details on the relaxation

Motivated by the bi-directional labeling algorithm by Righini and Salani [12] and the fact that branch and cut has been used quite successfully to solve the ESPPRC when time window like resources are not included (see Jepsen et al. [10]), we propose a Danzig-Wolfe decomposition approach based on a model where small sub paths called partial paths are concatenated together to form the solution. Since each of the sub paths are elementary the SR-inequalities for VRPTW introduced by Jepsen et al. [11] can be used to improve the lower bound. Furthermore any valid inequality to the ESPPRC can be used.

2 Bounded partial paths

The idea behind the following mathematical model and decomposition is that any feasible path p can be seen as a sequence of $K = \{1, \dots, |K|\}$ partial paths $p_{ov_1}, p_{v_1v_2}, \dots, p_{v_kd}$. Where p_{ij} is a partial path from node i to node j .

Each of the $|K|$ partial paths can be seen as a path through the original graph. This leads to an alternative formulation of the ESPPRC where the graph is replicated $|K|$ times and edges are added between the adjacent layers. The division is done as follows:

For resource $l \in R$, let w_{\max}^l be the maximal resource consumption. For a fixed number of partial paths $|K|$, the maximal partial path length L is given as:

$$L = \left\lceil \frac{W^l}{|K|} \right\rceil + w_{\max}^l$$

Let $\delta^+(S) = \{(i, j) \in A | i \in S\}$ denote arcs out of the set S and $\delta^-(S) = \{(i, j) \in A | j \in S\}$ denote the in going arcs of S . We shall use $\delta(i)$ instead of $\delta(\{i\})$ for $i \in V$. The binary variable x_{ijk} indicates if arc $(i, j) \in A$ is used in the k 'th layer. The binary variable s_{ik} indicates if a partial path starts in node $i \in V$ in layer $k \in K$ and the binary variable t_{ik} indicates if a partial path ends in node $i \in V$ in layer k . For ease of modelling we assume that $t_{i|K|+1} = s_{i0} = 0, \forall i \in V$. The mathematical model for ESPPRC based on partial paths can be formulated as:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} \quad (1)$$

$$\text{s. t. } \sum_{(o,j) \in \delta^+(o)} x_{ojk} = 1 \quad (2)$$

$$\sum_{(i,d) \in \delta^-(S)} x_{id|K|} = 1 \quad (3)$$

$$\sum_{k \in K} \sum_{(i,j) \in A} x_{ijk} \leq 1 \quad \forall v \in V \setminus \{o, d\} \quad (4)$$

$$\sum_{k \in K} \sum_{(i,j) \in A} w_{ij}^r x_{kij} \leq W^r \quad \forall r \in R \quad (5)$$

$$\sum_{k \in K} \sum_{(i,j) \in \delta^+(S)} x_{ijk} \geq \sum_{k \in K} \sum_{(i,j) \in \delta^+(s)} x_{ijk} \quad S \subseteq V, \forall s \in S \quad (6)$$

$$\sum_{i \in V} s_{ik} = 1 \quad \forall k \in K \quad (7)$$

$$t_{ik-1} = s_{ik} \quad \forall i \in V, \forall k \in K \quad (8)$$

$$s_{ik} + \sum_{(j,i) \in \delta^-(i)} x_{jik} = t_{ik} + \sum_{(i,j) \in \delta^+(i)} x_{ijk} \quad \forall i \in V, k \in K \quad (9)$$

$$\sum_{(i,j) \in \delta^+(S)} x_{ijk} \geq \sum_{(i,j) \in \delta^+(s)} x_{ijk} \quad \forall k \in K, S \subseteq V, \forall s \in S \quad (10)$$

$$\sum_{(i,j) \in A} x_{ijk} \leq 1 \quad \forall k \in K \quad (11)$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i, j) \in A, k \in K \quad (12)$$

$$l s_{ik}, l t_{ik}, s_{ik} \in \{0, 1\} \quad \forall i \in V, k \in K \quad (13)$$

The objective 1 is to minimize the total cost of the path. Constraints 2, 3 and 4 ensure that no node is visited more than once and that the path leaves and enters source and target. Constraints 5 are the resource bounds and constraints 6 are the generalized subtour inequalities(GSEC) which prevents cycles in a solution. Constraints 7 to 11 ensure that the partial paths are elementary, connected and do not violate the reduced resource.

In the following we will make a Danzig-Wolfe reformulation of the mathematical model, where constraints 9 to 11 form K identical sub problems. The single sub problem is to find a shortest path p between two arbitrary nodes in the graph. Let $\alpha_{ij}^p = 1$ if path p uses edge (i, j) and zero otherwise, $\beta_i^p = 1$ if p starts in node i and γ_i^p indicate if p ends in node i . Let the binary variable λ^p indicate if partial path p is used and c_p be the cost of using the path. The master problem then becomes:

$$\min \sum_{p \in P} c_p \lambda_p \quad (14)$$

$$\text{s.t.} \sum_{p \in P} \sum_{(o,j) \in \delta^+(o)} \alpha_{oj}^p \lambda_p = 1 \quad (15)$$

$$\sum_{p \in P} \sum_{(i,d) \in \delta^-(S)} \alpha_{id}^p \lambda_p = 1 \quad (16)$$

$$\sum_{p \in P} \sum_{(i,j) \in A} \alpha_{ij}^p \lambda_p \leq 1 \quad \forall v \in V \setminus \{o, d\} \quad (17)$$

$$\sum_{p \in P} \sum_{(i,j) \in \delta^+(S)} \alpha_{ij}^p \lambda_p \geq \sum_{p \in P} \sum_{(i,j) \in \delta^+(s)} \alpha_{ij}^p \lambda_p \quad S \subseteq V, \forall s \in S \quad (18)$$

$$\sum_{p \in P} \sum_{(i,j) \in A} w_{ij}^r \alpha_{ij}^p \lambda_p \leq W^r \quad \forall r \in R \quad (19)$$

$$\sum_{p \in P} \lambda^p = |K| \quad (20)$$

$$\sum_{p \in P} \gamma_i^p \lambda_p = \sum_{p \in P} \beta_i^p \lambda_p \quad \forall i \in V \quad (21)$$

$$s_{ik} \in \{0, 1\} \quad \forall i \in V, k \in K \quad (22)$$

$$\lambda_p \in \{0, 1\} \quad \forall p \in P \quad (23)$$

With the exception of constraint 20 the constraints follow directly from a standard Danzig-Wolfe reformulation. Constraint 21 substitutes the $|K|$ constraints 7 and states that we must choose $|K|$ columns corresponding to one from each layer. The master model may be too large to solve, therefore delayed column generation is used to solve it.

Let π_i be the $|V|$ dual of constraints 15, 16 and 17, σ_r be the $|R|$ duals of constraints 19 and ρ_i be the $|V|$ duals of constraints 21. To calculate the reduced cost of a column in the master problem, we set the edge cost to: $\hat{c}_{ij} = c_{ij} - \pi_i - \sum_{r \in R} w_{ij}^r \sigma_r + \rho_i - \rho_j$ using standard Linear Programming theory. Let x_{ij} be the binary variable which defines if arc $(i, j) \in A$ is used, the binary variable l_s indicate if the path starts in node $i \in V$ and the binary variable l_t indicate if the path ends in node $i \in V$. The mathematical model for the pricing problem then becomes:

$$\min \sum_{(i,j) \in A} \hat{c}_{ij} x_{ij} \quad (24)$$

$$\sum_{i \in V} l_{s_i} \leq 1 \quad (25)$$

$$\sum_{i \in V} l_{t_i} \leq 1 \quad (26)$$

$$l_{s_i} + \sum_{(j,i) \in \delta^-(i)} x_{ji} = l_{t_i} + \sum_{(i,j) \in \delta^+(i)} x_{ij} \quad \forall i \in V \quad (27)$$

$$(28)$$

$$\sum_{(i,j) \in \delta^+(S)} x_{ij} \geq \sum_{(i,j) \in \delta^+(s)} x_{ij} \quad S \subseteq V, \forall s \in S \quad (29)$$

$$\sum_{(i,j) \in A} x_{ij} \leq L \quad (30)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (31)$$

$$ls_i, lt_i \in \{0, 1\} \quad \forall i \in V \quad (32)$$

A column has negative reduced cost if it is less than the dual variable of constraint 20.

To solve the pricing problem we reformulate it to an ESPPRC. This is done by substituting the variables ls_i and lt_i with arcs from a fake source node and arcs to a fake target node.

More formally we define a fake source node \bar{s} and a fake target node \bar{t} . The fake arc set $\bar{A} = \{(\bar{s}, v) : v \in V\} \cup \{(v, \bar{t}) : v \in V\}$. The pricing problem then becomes solving an ESPPRC with a single resource in the graph $\bar{G}(V \cup \bar{V}, A \cup \bar{A})$ where the cost is $\bar{c}_{ij} = \hat{c}_{ij}(i, j) \in A$ and $\bar{c}_{ij} = 0, (i, j) \in \bar{A}$

The lower bound can be improved using valid inequalities for the ESPPRC polytope, and valid inequalities for the master model such as the SR-inequalities by Jepsen et al. [11].

Furthermore the lower bound can be improved by imposing a minimum length on each of the partial paths. Let L_{min} be the minimum length of a partial path. If $2L_{min} - 1 = L$ any feasible solution with length greater than L_{min} can be constructed if constraint 20 is relaxed to $\sum_{p \in P} \lambda^p \leq |K|$.

3 Implementation

We have implemented the bidirectional labeling algorithm by Righini and Salani [12], which is used to solve the pricing problem. When we impose a lower bound on the length of the partial path we only use dominance when the labels have equal length.

The Branch-Cut-And-Price algorithm is implemented in the BCP project from coin-or.org. We use CLP as our LP solver and we separate the GSECs solving a minimum cut problem (see Wolsey [13]) for details. The SR inequalities are separated using the algorithm proposed by Jepsen et al. [11], either the first or the last node on a partial path is not considered part of the SR-cut. Branching is done on a single arc or all arcs out of a node and is added as a cut in the master model. The constraints in the original space are:

$$\sum_{k \in K} \sum_{(i,j) \in \delta^+(i)} x_{ijk} = 0 \quad \sum_{k \in K} \sum_{(i,j) \in \delta^+(i)} x_{ijk} = 1 \quad i \in V \quad (33)$$

$$\sum_{k \in K} x_{ijk} = 0 \quad \sum_{k \in K} x_{ijk} = 1 \quad (i, j) \in A \quad (34)$$

The decomposed version of the branches are:

$$\sum_{p \in P} \sum_{(i,j) \in \delta^+(i)} \alpha_{ij}^p \lambda_p = 0 \quad \sum_{p \in P} \sum_{(i,j) \in \delta^+(i)} \alpha_{ij}^p \lambda_p \quad i \in V \quad (35)$$

$$\sum_{p \in P} \alpha_{ij}^p \lambda_p = 0 \quad \sum_{p \in P} \alpha_{ij}^p \lambda_p = 1 \quad (i, j) \in A \quad (36)$$

4 Computational studies

Based on a column generation algorithm for CVRP, some instances for the ESPPRC were generated based on CVRP instances from www.branchandcut.org. We have tested several settings on several instances with 20-30 nodes and have based on these chosen some settings to use on larger instances.

For the instances generated we have bounded the partial path using both length and capacity. When using length we have chosen to restrict the maximal lower limit to 3 and the maximal upper limit to 5. For the capacity we split the path in pieces of at most a tenth of the total capacity, finally we included the SR inequalities for the different settings.

We have chosen to show the result for a single instance which was quit representative for the instances we benchmarked on. Furthermore the instance have the characteristics we are targeting to solve. The instance have 30 nodes, the maximal feasible path length is 23 and the capacity resource is 4500.

In table 1 we have compared some different settings for capacity and length. L_{\min} is the minimal value of the partial path, L_{\max} is the maximal value of the partial path. RB is the root bound and T is the time without SR inequalities. RB_{SR} and T_{SR} is the root bound and time when SR inequalities is included.

Instance	Bounded on	L_{\min}	L_{\max}	RB	T	RB_{SR}	T_{SR}
E-n30-k3-20	Capacity	0	2125	-192350	311.895	-192350	386.072
E-n30-k3-20	Capacity	0	1700	-192350	228.958	-192350	203.585
E-n30-k3-20	Capacity	0	1300	-192320	49.579	-192320	143.685
E-n30-k3-20	Capacity	0	1193	-192350	31.810	-192350	128.412
E-n30-k3-20	Capacity	0	1113	-192350	23.653	-192350	120.776
E-n30-k3-20	Capacity	0	1000	-192350	39.098	-192350	171.571
E-n30-k3-20	Capacity	0	900	-192350	20.173	-192350	118.271
E-n30-k3-20	Length	0	3	-192350	20.361	-192350	19.893
E-n30-k3-20	Length	0	4	-192350	44.407	-192350	50.455
E-n30-k3-20	Length	0	5	-192350	134.080	-192350	99.106
E-n30-k3-20	Length	0	6	-192350	255.236	-192350	269.989
E-n30-k3-20	Length	2	3	-192350	75.873	-192350	109.283
E-n30-k3-20	Length	2	4	-192350	102.402	-192350	113.751
E-n30-k3-20	Length	3	5	-192350	163.822	-192320	431.763

Table 1: Comparing different schemes

From the result in table 1 it is clear that the longer the path the poorer the algorithms perform. The main reason for this is that no matter how long the path becomes there is simply no gain in the quality of the relaxation. The value of the root bound is almost the same as the one for branch and cut has, which is -192352.787 . When including the SR inequalities only a few of the setting results in a improvement of the running time. When a lower bound on the path length is included the running time increases in all cases and the root bound is still the same.

In table 2 we have shown the solution times for the two best partial paths algorithms. T_{len} is the running time of the best length algorithm and T_{cap} is the running time of the best with capacity. The values are compared to the bi-directional labeling algorithm(T_{label}) and the Branch and Cut algorithm(T_{BAC}) by Jepsen et al. [10].

Instance	T_{BAC}	T_{label}	T_{len}	T_{cap}
E-n30-k3-20	0.44	> 1800	19.893	20.173
B-n31-k5-17	2.07	0.22	124.492	24.178
A-n32-k5-120	0.51	0.28	32.714	7.892
A-n33-k5-31	0.45	0.01	121.440	14.477
B-n34-k5-17	2.21	72.79	290.022	32.554
B-n45-k6-54	4.63	90.3	286.978	109.011
P-n45-k5-150	0.58	0.71	19.753	15.457
P-n50-k8-19	0.94	> 1800	188.008	25.350
E-n51-k5-29	2.46	> 1800	277.645	287.746

Table 2: Characteristics of the benchmark instances

The main conclusion when comparing the results in table 2 is that the branch and cut algorithm outperforms the other algorithms. The second observation is that the partial path algorithms is able to solve all instances within 30 minutes which labeling is not. It is also worth noting that the algorithm which bounds using capacity in almost all cases is considerable better than the one that bounds using length. Finally we conclude that the partial path algorithms can not compete with the Branch And Cut algorithm.

5 Conclusion and future research

In this conference paper we have introduced an alternative formulation of ESPPRC and shown how it can be solved using the Danzig-Wolfe decomposition principle. We have shown that a early prototype is better than a standard labeling algorithm, but we have not been able to show that the bound obtained is better than a standard Branch and Cut algorithm. Therefore an open problems which has arisen during this research is if there exist an instance where the bound obtained by the partial path algorithm results in a better than a Branch and Cut algorithm . We also suggest that some effort is made to find a way handle cover inequalities on the master variables in the pricing problem

References

- [1] 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.
- [2] J.E. Beasley and N. Christofides. *An algorithm for the resource constrained shortest path problem*. *Networks*, 19:379–394, 1989.
- [3] 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.
- [4] W. Matthew Carlyle and R. Kevin Wood. *Near-shortest and k-shortest simple paths*. *Netw.*, 46(2):98–109, 2005. ISSN 0028-3045. doi: <http://dx.doi.org/10.1002/net.v46:2>.
- [5] W. Matthew Carlyle, Johannes O. Royset, and R. Kevin Wood. *Lagrangian relaxation and enumeration for solving constrained shortest-path problems*. *Networks*, 2008.
- [6] A. Chabrier. *Vehicle routing problem with elementary shortest path based column generation*. *Computers & Operations Research*, 23(10):2972–2990, 2005.
- [7] I. Dumitrescu and N. Boland. *Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem*. *Networks*, 42(3):135–153, 2003.
- [8] 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.
- [9] Stefan Irnich and Daniel 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. ISSN 1526-5528. doi: <http://dx.doi.org/10.1287/ijoc.1040.0117>.
- [10] M. Jepsen, B. Petersen, and S. Spoorendonk. *A branch-and-cut algorithm for the elementary shortest path problem with a capacity constraint*. Technical report, DIKU, University of Copenhagen, Denmark, 2008.
- [11] 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. doi: 10.1287/opre.1070.0449. URL <http://or.journal.informs.org/cgi/content/abstract/56/2/497>.
- [12] 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. ISSN 1572-5286. doi: 10.1016/j.disopt.2006.05.007.
- [13] L. A. Wolsey. *Integer Programming*. John Wiley & Sons, Inc., 1998.