



**A note on the Flow Extended 0-1 Knapsack  
Cover Inequalities for the Elementary Shortest  
Path Problem with a Capacity Constraint**

Mads Kehlet Jepsen, Simon Spoorendonk

Technical Report no. 08/02

ISSN: 0107-8283

**Dept. of Computer Science**

University of Copenhagen • Universitetsparken 1

DK-2100 Copenhagen • Denmark

# A note on the Flow Extended 0-1 Knapsack Cover Inequalities for the Elementary Shortest Path Problem with a Capacity Constraint

Mads Kehlet Jepsen · Simon Spoorendonk  
mitzi@diku.dk spooren@diku.dk  
*DIKU Department of Computer Science, University of Copenhagen  
Universitetsparken 1, DK-2100 Copenhagen Ø, Denmark*

March 5, 2008

## Abstract

This note introduces an extension to the 0-1 knapsack cover inequalities to be used in a branch-and-cut algorithm for the elementary shortest path problem with a capacity constraint. The extension leads to a set of valid inequalities that takes both the fractional usage of the edges and the capacity into account and are denoted the flow extended 0-1 knapsack cover inequalities. Computational experiments indicate that although these new inequalities improve the lower bound they also results in more fractional LP solutions which results in a larger number of branch nodes and eventually slower running times.

**Keywords:** Branch-and-Cut, Elementary Shortest Path Problem with Resource Constraints, Capacitated Vehicle Routing Problem

## 1 Introduction

This note introduces the flow extended 0-1 knapsack cover inequalities for the elementary shortest path problem with a capacity constraints (ESPPCC). This is complimentary work to the branch-and-cut (BAC) algorithm presented by Jepsen et al. (2008). Hence, for further literature review and details on the BAC algorithm we refer to the above paper. In this note we focus solely on the flow extended 0-1 knapsack cover inequalities.

The ESPPCC can be stated as: Given an undirected graph  $G(V, E)$  with nodes  $V$  and edges  $E$ , a cost  $c_e$  associated to each edge  $e \in E$ , a load  $d_i$  associated to each node  $i \in V$ , an upper limit on the amount of accumulated load  $Q$ , a source node  $s \in V$ , and a target node  $t \in V$ ; find the path between  $s$  and  $t$  with minimum cost satisfying that the sum of the loads from the visited nodes is not more than  $Q$ .

In the following, variable  $y_i$  indicate the use of node  $i \in V \setminus \{s, t\}$ , and variable  $x_e$  indicate the use of edge  $e \in E$  where  $e(i, j)$  denotes the end nodes  $i$  and  $j$  of  $e$ . When describing the model some shorthand notation will be used. For the set of edges  $T$  let

$$x(T) = \sum_{e \in T} x_e$$

Furthermore, for a set of nodes  $S \subseteq V$  let the set of edges  $\delta(S) = \{e(i, j) : i \in S \wedge j \in V \setminus S\}$  denote the edges between  $S$  and  $V \setminus S$  with  $\delta(\{i\}) = \delta(i)$  for a node  $i \in V$ . Also, for a set of nodes  $S$  let

$$y(S) = \sum_{i \in S} y(i)$$

and let  $E(S) = \{e(i, j) : i \in S \wedge j \in S\}$  be the set of edges between the nodes in  $S$ .

The mathematical model of ESPPCC is then:

$$\min \sum_{e \in E} c_e x_e \tag{1}$$

$$\text{s.t. } x(\delta(s)) = 1 \tag{2}$$

$$x(\delta(t)) = 1 \tag{3}$$

$$\sum_{e \in \delta(i)} x_e = 2y_i \quad \forall i \in V \setminus \{s, t\} \tag{4}$$

$$\sum_{i \in V} d_i y_i \leq Q \tag{5}$$

$$x(E(S)) \leq y(S) - y_i \quad \forall i \in S, \forall S \subseteq V, |S| \geq 2 \tag{6}$$

$$x_e \in \{0, 1\} \quad \forall e \in E \tag{7}$$

$$y_i \in \{0, 1\} \quad \forall i \in V \setminus \{s, t\} \tag{8}$$

The objective function (1) minimizes the overall edge cost. Constraints (2) and (3) ensure that the path starts in the source node and ends in the target. Constraints (4) map the  $x$  and  $y$  variables. Constraint (5) imposes the capacity. Constraints (6) are the generalized subtour elimination constraints and impose connectivity of the path. Finally, constraints (7) and (8) bound the variables indicating the use of edges and nodes.

This model has  $|E| + |V - 2|$  variables and an exponential number of constraints due to (6). In a BAC algorithm these constraints will be disregarded and separated when violated to ensure feasibility. The separation of the generalized subtour elimination constraints (6) can be done by solving  $V - 2$   $s - t$ -minimum cut problems, see Jepsen et al. (2008), Wolsey (1998).

## 2 Combining Flow and Capacity

The 0-1 knapsack cover inequality for a set of nodes  $S \subseteq V$  where  $\sum_{i \in S} d_i > Q$  are given as:

$$y(S) \leq |S| - 1 \tag{9}$$

The inequality states that if a set of nodes violates the capacity then not all nodes in the set can be visited by the path. The separation problem is a minimization version of the well known 0-1 knapsack problem, see Kellerer et al. (2004), Wolsey (1998).

By exploiting the fact that since  $y_i \leq 1$  for all  $i \in V \setminus \{s, t\}$  then the flow through a set of nodes  $S$  can be less than 2 in an LP solution. Hence, scaling the right-hand-side of (9) with half the flow  $x(\delta(S))$  yields the flow extended 0-1 knapsack cover inequality

$$y(S) \leq \frac{1}{2}(|S| - 1)x(\delta(S)) \tag{10}$$

When  $x(\delta(S)) < 2$  there are cases where the inequality (10) is violated and the normal 0-1 knapsack cover inequality (9) is not.

For now it is unknown whether an efficient separation routine exists for (10), therefore a heuristic separation routine is presented. The separation problem is to find a cover  $S$ , i.e.,  $\sum_{i \in S} d_i > Q$ , where  $x(\delta(S)) < 2$  in an induced graph  $G'$  containing only nodes and edges with fractional value in the LP solution. It is assumed that the graph is connected, e.g., there are no violated generalized subtour elimination constraints (6). Noting that, the number of edges in  $\delta(S)$  for a connected set  $S$  is generally smaller than in a disconnected set we will only consider connected sets of nodes. By performing a breadth-first search rooted at each of the nodes in the induced graph the connected candidate sets are build iteratively and stored if a violation of (10) occurs. The heuristic is summarized in the pseudocode below where the induced graph  $G'$  is taken as input and the set  $S^*$  violating (10) is returned. If  $S = \emptyset$  then no violating inequalities were found.

SEPARATE-FLOW-EXTENDED-0-1-KNAPSACK-COVERS( $G'$ )

```

1   $S^* \leftarrow \emptyset$ 
2  for each node  $i \in G'$ 
3      do  $S \leftarrow \emptyset$ 
4           $\Pi \leftarrow \text{BFS}(G', i)$ 
5          for each node  $j \in \Pi$ 
6              do  $S \leftarrow S \cup j$ 
7                  if  $\sum_{i \in S} d_i > Q \wedge x(\delta(S)) < 2$ 
8                      do  $S^* \leftarrow S$ 
9  return  $S^*$ 

```

The capacity of  $S$ , i.e.,  $\sum_{i \in S} d_i$ , can be calculated using accumulation during the execution of the **for** loop in line 5 while the calculation of  $x(\delta(S))$  in line 7 takes  $O(E)$  time in worst case. The BFS in line 4 performs the breath first search and returns a list  $\Pi$  of reachable nodes sorted such that the nodes closest to the start node are first. Each iteration of the **for** loop starting at line 2 takes  $O(V + E)$  for the BFS, see Cormen et al. (2001) and  $O(VE)$  for calculating  $x(\delta(S))$  for all nodes. This adds up to a worst case running time of  $O(V^2E)$ . In practice the number of nodes and edges in the induced graph is expected to much smaller than in the original graph so the running time is expected to be much faster.

### 3 Computational Results

The computational experiments are run on 10 instances based on the CVRP instances from <http://www.branchandcut.org>. The instances are divided in series A, B, E, and F according to the authors. The ESPPCC instances are pricing problems gathered when solving the CVRP with column generation, see Fukasawa et al. (2006), Jepsen et al. (2007), i.e., dual values have been subtracted from the edge costs resulting in instances with negative edge weights. All experiments are run on an AMD(R) Athlon(R) XP2400+ 2.0 GHz processor with 512 MB memory.

For the computational experiments three different cut separation settings are used:

- GSEC only separates the generalized subtour inequalities constraints (6).
- 0-1 KPC as above adding the 0-1 knapsack cover inequalities (9).
- FLOW as above adding the flow extended knapsack cover inequalities (10).

Table 1 is a comparison of the number of branch nodes (B&B) and root lower bounds (root) with the different cut separation settings. In the right most column (solution) the optimal solution is reported.

Name	GSEC		0-1 KPC		FLOW		solution
	B&B	root	B&B	root	B&B	root	
A-n32-k5-i35	51	-70.56	25	-70.56	37	-62.24	-12.18
A-n53-k7-i10	51	-82.17	23	-82.17	31	-81.91	-63.45
A-n53-k7-i36	307	-63.86	185	-63.86	203	-58.90	-22.68
B-n31-k5-i17	291	-29.55	141	-29.55	143	-27.76	-6.00
B-n50-k8-i28	207	-115.62	265	-115.62	163	-100.08	-10.12
E-n22-k4-i10	99	-27.29	45	-27.29	51	-26.24	0.00
E-n23-k3-i27	133	-37.86	49	-37.86	43	-36.59	0.00
E-n76-k10-i50	443	-35.68	191	-35.68	231	-31.92	-10.29
F-n45-k4-i53	59	-34.84	93	-34.84	191	-34.37	-13.30
F-n72-k4-i81	53	-14.61	55	-14.53	55	-14.53	-14.04

Table 1: Comparison of the number of branch nodes and lower bounds.

Setting 0-1 KPC where the 0-1 knapsack cover inequalities (9) are added does not improve the root lower bound compared to the basic GSEC setting. However, in almost all cases the smallest branch tree is obtained. This indicates an improvement of the lower bounds deeper in the branch tree where more node variables  $y$  are integer valued due to branching and fixing, hence raising the left-hand-side of (9) resulting in more violated inequalities.

The separation of the flow extended 0-1 knapsack cover inequalities (10) given by the FLOW setting results in an improvement of the lower bounds with a 10.2 % on average when compared to the 0-1 KPC setting. However, the number of branch nodes increases with an average of 7.1 %. The reason is that the LP solutions become more fractional with the FLOW setting compared to the 0-1 KPC setting. Figure 1 on the following page illustrates the fractional solution for the two cut separation settings.

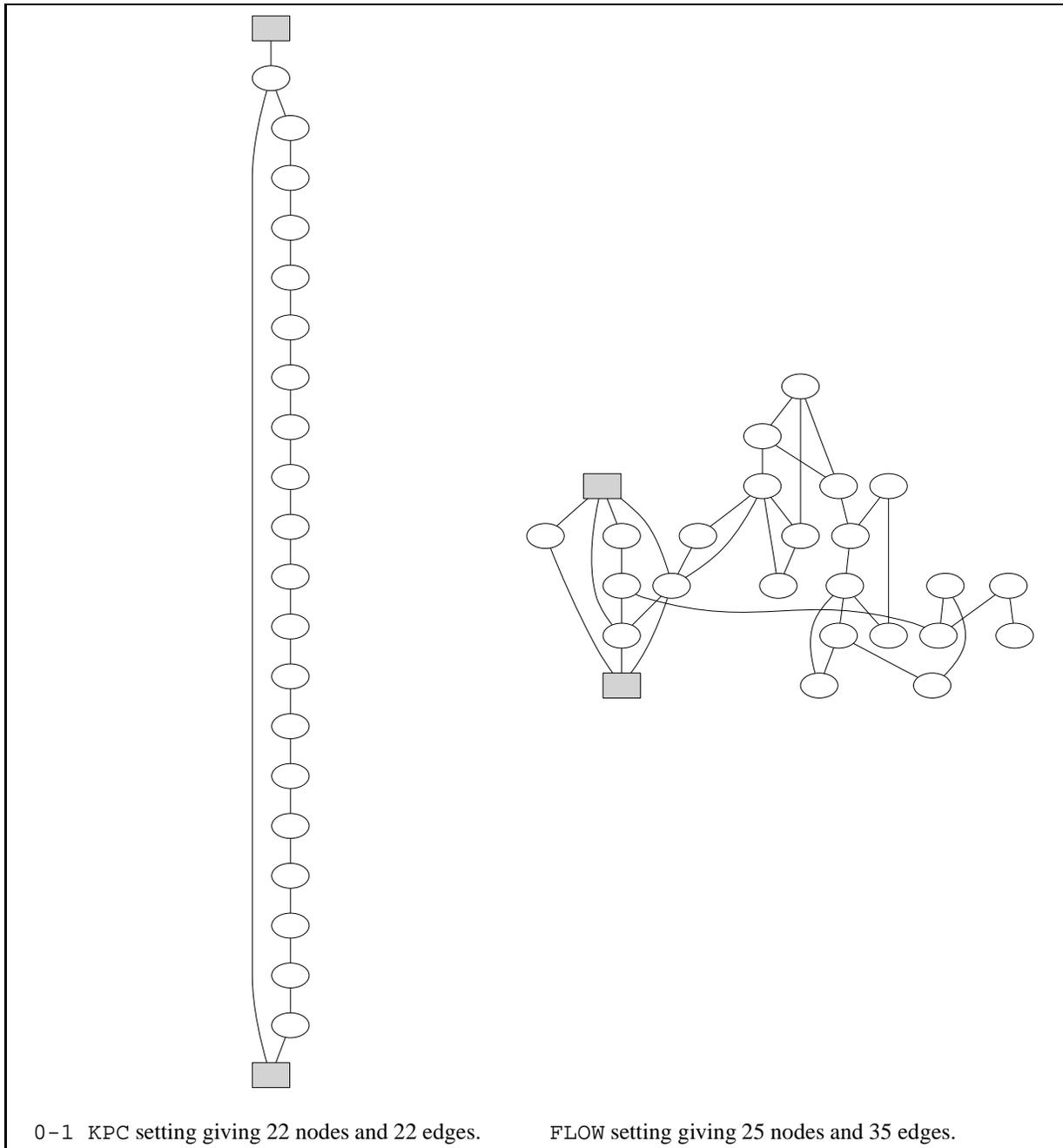


Figure 1: Root Solution of A-n32-k5-i35 using the 0-1 KPC and the FLOW settings. Only nodes and edges with non-zero flow are shown. The square nodes are the end nodes.

Table 2 is a comparison of the running times with different cut separation settings. The time (time (s)) is given in seconds for each of the three cut separation settings.

Name	GSEC time (s)	0-1 KPC time (s)	FLOW time (s)
A-n32-k5-i35	5.09	<b>3.92</b>	9.72
A-n53-k7-i10	24.74	<b>20.27</b>	27.90
A-n53-k7-i36	161.02	<b>138.46</b>	215.33
B-n31-k5-i17	<b>12.10</b>	16.13	21.38
B-n50-k8-i28	<b>240.73</b>	389.62	600.02
E-n22-k4-i10	1.43	<b>1.25</b>	1.86
E-n23-k3-i27	2.52	<b>1.35</b>	1.76
E-n76-k10-i50	1082.61	<b>702.97</b>	1365.85
F-n45-k4-i53	<b>17.79</b>	25.22	54.90
F-n72-k4-i81	<b>52.55</b>	52.98	53.34
Best	4	6	0

Table 2: Time comparison of the BAC algorithm.

The two cut separation settings GSEC and 0-1 KPC compete to be the most successful. It appears that the specifications of the instances matters, since the best running times are split between the B and F instances for the GSEC setting and the A and E instances for the 0-1 KPC setting. For the B and F instances this suggest that a lot of  $\mathcal{NP}$ -Hard 0-1 knapsack problems are solve for no gain. On the other hand the running times of the A and E instances indicates that the 0-1 knapsack cover inequalities (9) are worth the effort.

The running times clearly indicates that the FLOW setting is the least successful. The more fractional LP solutions leads to more branch nodes hence the BAC algorithm converges slower after adding the flow extended 0-1 knapsack cover inequalities (10).

## 4 Concluding Remarks

In this note we have introduced the flow extended 0-1 knapsack cover inequalities (10) for the ESP-PCC and performed computational experiments using the BAC algorithm presented in the companion paper by Jepsen et al. (2008).

The flow extended 0-1 knapsack cover inequalities did improve the root lower bound when compared to the cut separation settings using only generalized subtour elimination constraints and 0-1 knapsack cover inequalities. However, the LP solutions appeared to become much more fractional after adding the flow extended 0-1 knapsack cover inequalities which lead to an increase of branch nodes. It is possible that the flow extended 0-1 knapsack cover inequalities would be favored more by other branching rules, e.g., considering an integral flow out of a set of nodes rather than branching on fractional variables. That is, the flow extended 0-1 knapsack cover inequalities appears to have a negative effect on the convergence of the BAC algorithm which should be taken into account in any future implementations.

## References

Cormen, T. H., C. E. Leiserson, R. L. Rivest, C. Stein. 2001. *Introduction to Algorithms*. Addison.

- Fukasawa, R., H. Longo, J. Lygaard, M. Poggi de Aragão, M. Reis, E. Uchoa, R.F. Werneck. 2006. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming* **106**(3) 491–511.
- Jepsen, M., B. Petersen, S. Spoorendonk. 2008. A branch-and-cut algorithm for the elementary shortest path problem with a capacity constraint. Tech. Rep. 08-01, Department of Computer Science (DIKU), University of Copenhagen, Denmark, Universitetsparken 1, DK-2100 Copenhagen Ø, Denmark.
- Jepsen, M., B. Petersen, S. Spoorendonk, D. Pisinger. 2007. Subset-row inequalities applied to the vehicle routing problem with time windows. *Operations Research* doi:10.1287/opre.1070.0449.
- Kellerer, H., U. Pferschy, D. Pisinger. 2004. *Knapsack Problems*. Springer, Berlin, Germany.
- Wolsey, L. A. 1998. *Integer Programming*. John Wiley & Sons, Inc.