Breaking symmetries in the bin packing problem and applications in network design

Amel Benhamiche[2], A.Ridha Mahjoub[2], Nancy Perrot[1], Eduardo Uchoa[3]

[1] Orange Labs (France Télécom R&D)
[2] Université Paris Dauphine, LAMSADE
[3] Universidade Federal Fluminense

Paris, 26/09/2014

Outline

Outline

UNACND problem

Input

- a bidirected graph $G = (V, A)$,
- a set of commodities $K = \{(o_k, d_k), D^k\}$. D^k is the traffic amount of the commodity k,
- a set of available modules $W = \{1, 2, \ldots, m\}$,
- C and c_{ij}, $ij \in A$, denote the capacity and the cost of a module installed on ij.

UNACND problem

Input

- a bidirected graph $G = (V, A)$,
- a set of commodities $K = \{(o_k, d_k), D^k\}$. D^k is the traffic amount of the commodity k,
- a set of available modules $W = \{1, 2, \dots, m\}$,
- C and c_{ij}, $ij \in A$, denote the capacity and the cost of a module installed on ij.

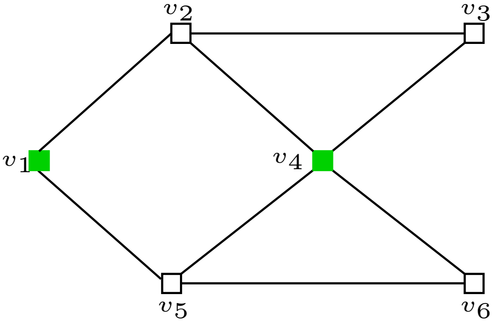
Goal

The UNACND problem consists in determining the number of modules that have to be installed on each arc of G, so that all the commodities can be routed and the total cost is minimum

- commodities are unsplittable (even between two modules of the same arc !)
- the capacities of the modules are non-additive

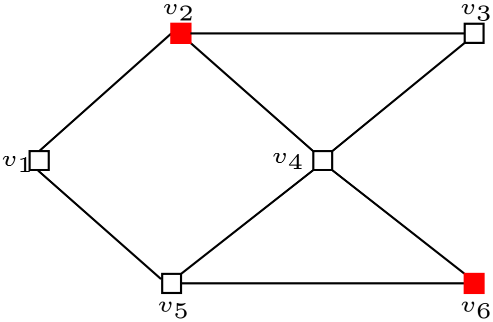
UNACND - example with C = 10

$D^1 = 6$



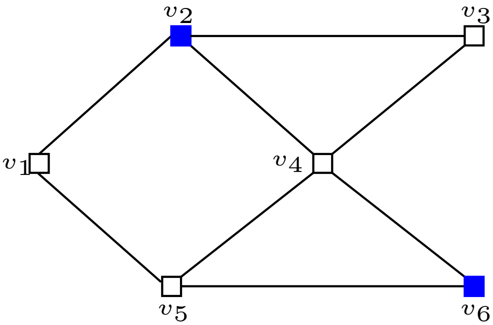
UNACND - example with C = 10

$D^1 = 6$, $D^2 = 6$



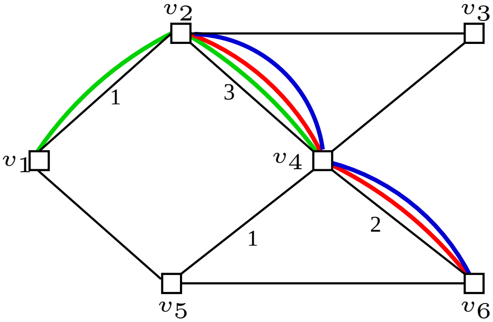
UNACND - example with C = 10

$D^1 = 6$, $D^2 = 6$, $D^3 = 6$



UNACND - example with C = 10

$D^1 = 6$, $D^2 = 6$, $D^3 = 6$



Compact formulation for UNACND

$$y_{ij}^{w} = \begin{cases} 1, & \text{if } w \text{ is installed on } ij, \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{ij}^{kw} = \begin{cases} 1, & \text{if } k \text{ uses the module } w \textit{ on arc } ij \text{ for its routing}, \\ 0, & \text{otherwise.} \end{cases}$$

Compact formulation for UNACND

$$\min \sum_{ij \in A} \sum_{w \in W} c_{ij} y_{ij}^w$$

$$\sum_{w \in W} \sum_{j \in V} x_{ji}^{kw} - \sum_{w \in W} \sum_{j \in V} x_{ij}^{kw} = \begin{cases} 1, & \text{if } i = d_k, \\ -1, & \text{if } i = o_k, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall k \in K, \forall i \in V,$$

$$\sum_{k \in K} D^k x_{ij}^{kw} \leq C y_{ij}^w, \qquad \forall w \in W, \forall ij \in A,$$

$$0 \leq x_{ij}^{kw} \leq 1, x_{ij}^{kw} \in \{0,1\}, \qquad \forall k \in K, \forall w \in W, \forall ij \in A,$$

$$0 \leq y_{ij}^w \leq 1, y_{ij}^w \in \{0,1\}, \qquad \forall w \in W, \forall ij \in A.$$

Outline

Compact formulation for UNACND

The UNACND problem, restricted to a single arc ij is to determine the number of modules to install over ij so that

- each commodity is assigned at most one module
- the total cost is minimum

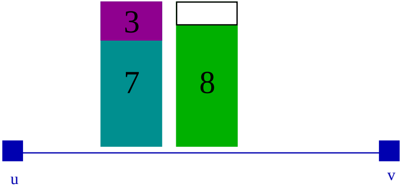
$$P_{ij} := conv\{(x, y) \in \{0,1\}^{|K| \times |W|} \times \{0,1\}^{|W|} :$$

$$\sum_{k \in K} D^k x_{ij}^{kw} \leq C y_{ij}^w, \forall w \in W,$$

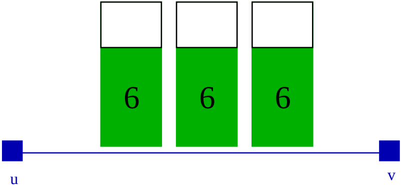
$$\sum_{w \in W} x_{ij}^{kw} \leq 1, \forall k \in K\}$$

UNACND and Bin-Packing problem

- three commodities going from u to v
- $D^1 = 8,\ D^2 = 7,\ D^3 = 3$
- $C = 10$



- three commodities going from u to v
- $D^1 = 6,\ D^2 = 6,\ D^3 = 6$
- $C = 10$



Aggregated formulation for UNACND

$y_{ij} \in \mathbb{Z}^+$ is the number of modules installed on the arc $ij \in A$,

$$x_{ij}^k = \begin{cases} 1 & \text{if } k \text{ uses some module installed on } ij, \\ 0 & \text{otherwise} \end{cases}$$

Aggregated formulation for UNACND

$$\min \sum_{ij \in A} c_{ij} y_{ij}$$

$$\sum_{j \in V} x^k_{ji} - \sum_{j \in V} x^k_{ij} = \begin{cases} 1, & \text{if } i = d_k, \\ -1, & \text{if } i = o_k, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall k \in K, \forall i \in V,$$

$$\sum_{k \in K} D^k x^k_{ij} \leq C y_{ij}, \qquad \forall ij \in A,$$

$$(x^k_{ij}, y_{ij}) \in Q_{ij}, \qquad \forall ij \in A, \forall k \in K.$$

where

$$Q_{ij} := conv\{(x, y) \in \{0,1\}^{|K|} \times \mathbb{Z}^+ : x^k_{ij} = \sum_{w \in W} x^{kw}_{ij}, y_{ij} \geq \sum_{w \in W} y^w_{ij},$$
$$\sum_{k \in K} D^k x^{kw}_{ij} \leq C y^w_{ij}, x^{kw}_{ij} \in \{0,1\}, y^w_{ij} \in \{0,1\}, \forall k \in K, \forall w \in W\}$$

Bin-Packing Function

Given m items (demands) and n bins. We denote by D^k the weight of the item k, $k \in \{1, \ldots, m\}$ and C is the capacity of each bin. The Bin-Packing problem consists in assigning each item to one bin so that the total weight does not exceed C and the number of bins used is minimum.

$$\min \sum_{j \in N} y_j$$

$$s.t:$$

$$\sum_{k \in K} D^k x_j^k \leq C y_j, \qquad \forall j \in N, \qquad (1)$$

$$\sum_{j \in N} x_j^k = 1, \qquad \forall k \in K, \qquad (2)$$

$$0 \leq y_j \leq 1, \quad y_j \in \{0,1\}, \qquad \forall j \in N, \qquad (3)$$

$$0 \leq x_j^k \leq 1, \quad x_j^k \in \{0,1\}, \forall k \in K, \forall j \in N. \qquad (4)$$

Bin-Packing Function

$$P := conv\{(x, y) \in \{0,1\}^{m \times n} \times \{0,1\}^{n} : (1) - (2) \text{ are satisfied}\}$$

$y \in \mathbb{Z}^{+}$ number of bins used to pack the items of K,

$$x^k = \begin{cases} 1 & \text{if } k \text{ is assigned to some bin,} \\ 0 & \text{otherwise} \end{cases}$$

Bin-Packing Function - associated polyhedron

- subset of items $S \subseteq K$
- $BP(S)$ is the smallest number of bins required for S
- $S(x)$ is the subset of items induced by x ($S(x) = \{k \in K : x^k = 1\}$)

$$Q := conv\{(x, y) \in \{0,1\}^{|K|} \times \mathbb{Z}^+ : y \geq BP(S(x))\}$$

Theorem 1.

Q is full dimensional.

Theorem 2.

(i) *For $k \in K$, inequalities $x^k \geq 0$ define facets of Q,*

(ii) *For $k \in K$, inequalities $x^k \leq 1$ define facets of Q.*

Min Set I inequalities

- a subset $S \subseteq K$
- a non negative integer $p \in \mathbb{Z}^+$

$$\sum_{k \in S} x^k \leq y + p,$$

is valid for Q if and only if $p \geq |S| - BP(S)$

Theorem 3.

Let S be a subset of K and p a non negative integer parameter. Inequality

$$\sum_{k \in S} x^k \leq y + p$$

defines a facet of Q if and only if the following conditions hold

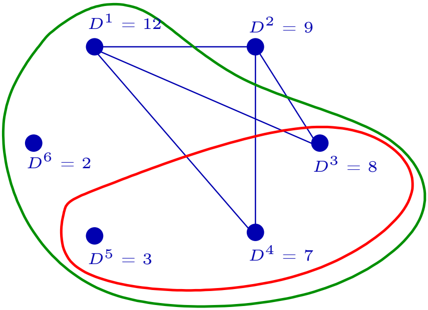
(i) $BP(S) = |S| - p$,

(ii) $BP(S \cup \{\widetilde{s}\}) = |S| - p$, *where $\widetilde{s}$ is the largest element of $K \setminus S$,*

(iii) $BP(S \setminus \{\overline{s}\}) \leq |S| - p - 1$, *where $\overline{s}$ is the smallest element of S.*

Min Set I inequalities

Example with C = 15



Examples of valid inequalities

$$x^1 + x^2 + x^3 \leq y, \quad p = 0$$
$$x^3 + x^4 + x^5 \leq y + 1, \quad p = 1$$
$$x^1 + x^3 + x^4 + x^5 + x^6 \leq y + 2, \quad p = 2$$

Min Set II inequalities

- Let S be a subset of K
- p and q, two non negative integer, $q \geq 2$

$$\sum_{k \in S} x^k \leq qy + p,$$

is valid for Q if and only if $p \geq (|S'| - qBP(S'))$, for all $S' \subseteq S$

Theorem 4.

Given a subset of elements $S \subseteq K$, two non negative integers $q \geq 2$ and p. The inequality

$$\sum_{k \in S} x^k \leq qy + p$$

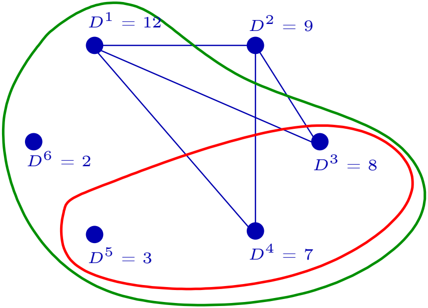
defines a facet of Q, if the following hold.

(i) *There exists an integer $r \in \mathbb{Z}_+$, $p \leq r \leq |S| - 1$, such that for all $S' \subseteq S$ with $|S'| = r$, $BP(S') = \frac{|S'|-p}{q}$,*

(ii) *for all $s \in K \setminus S$, there exists $S' \subseteq S$ such that $BP(S') = \frac{|S'|-p}{q} = BP(S' \cup \{s\})$,*

Min Set II inequalities

Example with C = 15



Examples of valid inequalities

$$x^1 + x^5 + x^6 \leq 2y, \qquad q = 2, p = 0$$
$$x^1 + x^2 + x^3 + x^4 + x^6 \leq 2y, \qquad q = 2, p = 0$$

UNACND using Bin-Packing Function

$$Q_{ij} := conv\{(x,y) \in \{0,1\}^{|K|} \times \mathbb{Z}^+ : x_{ij}^k = \sum_{w \in W} x_{ij}^{kw}, y_{ij} \geq \sum_{w \in W} y_{ij}^w,$$

$$\sum_{k \in K} D^k x_{ij}^{kw} \leq C y_{ij}^w, x_{ij}^{kw} \in \{0,1\}, y_{ij}^w \in \{0,1\}, \forall k \in K, \forall w \in W\}$$

- the UNACND problem restricted to an arc reduces to the Bin-Packing problem,
- Q_{ij}, $ij \in A$ is equivalent to the Bin-Packing func polyhedron Q

Outline

Branch-and-Cut algorithm for UNACND problem

- we start the optimization with the following linear program

$$\min \sum_{(i,j)\in A} c_{ij} y_{ij}$$

$$\sum_{j\in V} x_{ji}^k - \sum_{j\in V} x_{ij}^k = \begin{cases} 1, & if i = d_k, \\ -1, & if i = o_k, \\ 0, & otherwise, \end{cases} \qquad \forall k \in K, \forall i \in V,$$

$$\sum_{k\in K} D^k x_{ij}^k \leq C y_{ij}, \qquad \forall ij \in A,$$

$$0 \leq x_{ij}^k \leq 1, \qquad \forall k \in K, \forall ij \in A,$$

$$0 \leq y_{ij} \leq 1, \qquad \forall ij \in A.$$

- for each arc $ij \in A$, we solve the separation problem for Min Set I and Min Set II
- the valid inequalities are generated in the following order
  1. Min Set I inequalities
  2. Min Set II inequalities

Separation of Min Set I inequalities

For every arc a of the graph

- we consider a relaxed version of Min Set I inequalities,

$$\sum_{k \in S_a} x^k \leq y + |S_a| - \frac{\sum_{k \in S_a} D^k}{C}$$

- we solve the associated separation problem by using a greedy algorithm, and get the subset S_a,
- we use a strong lower bound denoted $L_*^2(S_a)$ [1], instead of computing $BP(S_a)$,
- we check if the corresponding Min Set I inequality is violated or not (with $p = |S_a| - L_*^2(S_a)$).

1. S. P. Fekete, J. Schepers, *New classes of fast lower bounds for bin packing problems*, Mathematical Programming 91 (1) (2001) 11-31

Separation of Min Set II inequalities

For every arc a of the graph

- we compute a subset S_a of commodities (similarly to Min Set I separation)
- for every subset S' of S_a, we compute $L^2_*(S')$
- if the inequality induced by S_a is valid, then we verify if it is violated by the current solution and add it to the current LP if so.

Branch-and-Cut algorithm for UNACND problem

- CPLEX 12.5 Callable library for the Branch-and-Cut framework and as a linear solver,
- C++
- PC with Bi-Xeon quad-core E5507 2.27GHz with 8Go of RAM
- CPU time limit fixed to 5 hours

- random instances
  - instances based on data from SNDlib networks
  - several graph topologies : sparse to strongly meshed, with euclidean distances
  - randomly generated commodities (origin, destination and traffic)
  - $|W| = 10$
- realistic instances
  - SNDlib network topologies
  - K most important traffic demands
  - $|W| = 5$

Compact formulation vs Aggregated formulation

				Compact formulation (B&B)			Aggregated formulation (B&C)				
Instance	|V|	|A|	|K|	Gap %	nodes	TT	Gap %	nodes	TT	#msI	#msII
nobel_us_2_1	14	42	2	22.37	182	0 :00 :09	58.83	18	0 :00 :00	46	0
nobel_us_4_1	14	42	4	30.98	1734	0 :02 :23	39.57	14	0 :00 :00	46	0
nobel_us_6_1	14	42	6	0.00	1	0 :00 :01	25.55	17	0 :00 :00	109	0
nobel_us_8_1	14	42	8	13.29	4978	0 :23 :03	22.35	64	0 :00 :03	169	0
nobel_us_10_1	14	42	10	*29.08*	82768	5 :00 :00	29.02	114	0 :00 :01	158	0
nobel_us_12_1	14	42	12	*11.75*	71534	5 :00 :00	28.90	278	0 :00 :26	291	2
nobel_us_14_1	14	42	14	*27.47*	55221	5 :00 :00	22.50	302	0 :00 :48	556	14
nobel_us_16_1	14	42	16	*6.57*	74429	5 :00 :00	20.28	281	0 :00 :46	580	21
nobel_us_18_1	14	42	18	*7.48*	64164	5 :00 :00	22.34	1001	0 :03 :45	757	27
nobel_us_20_1	14	42	20	*10.79*	13291	5 :00 :00	22.92	295	0 :00 :49	434	6

Computational results - random instances

Instance	|V|	|A|	|K|	#msI	#msII	Gap %	Opt	nodes	TT
polska_10	12	36	10	247.6	3.2	14.61	5/5	66.4	0 :00 :03
polska_15	12	36	15	533.2	3.6	16.63	5/5	205.8	0 :00 :25
polska_20	12	36	20	970.8	39.8	17.78	5/5	835.2	0 :03 :37
polska_30	12	36	30	3294.2	149.2	15.06	4/5	4818	1 :14 :15
polska_40	12	36	40	7596.8	388.4	16.43	1/5	17778.6	4 :02 :08
newyork_10	16	98	10	271.2	2.4	10.61	5/5	110.6	0 :00 :10
newyork_15	16	98	15	598	11.4	12.66	5/5	527.4	0 :01 :33
newyork_20	16	98	20	1993.6	28.2	14.09	5/5	3778.4	0 :34 :21
newyork_30	16	98	30	4683.4	101.2	*15.78*	0/5	17894.6	5 :00 :00
newyork_40	16	98	40	8994.8	99.4	*60.53*	0/5	10812.2	5 :00 :00
geant_10	22	72	10	155.6	0.6	16.65	5/5	35.6	0 :00 :03
geant_15	22	72	15	312.8	3	14.27	5/5	98.2	0 :00 :17
geant_20	22	72	20	353	1	13.11	5/5	98.2	0 :00 :23
geant_30	22	72	30	1496.6	21.6	12.34	5/5	686.8	0 :07 :15
geant_40	22	72	40	3111.2	37.2	12.60	5/5	2096.4	0 :54 :45
ta1_10	24	110	10	415.6	7.6	21.03	5/5	778	0 :02 :04
ta1_15	24	110	15	1120.4	78.4	27.32	4/5	8788.4	1 :08 :46
ta1_20	24	110	20	1920	49.4	25.25	3/5	10870	2 :45 :12
ta1_30	24	110	30	4570	69.6	*25.88*	0/5	9886.8	5 :00 :00
ta1_40	24	110	40	9187.2	117.8	*52.85*	0/5	13739.8	5 :00 :00
pioro_10	40	178	10	884	9	51.72	5/5	5719.6	0 :10 :36
pioro_15	40	178	15	2471.4	82.8	56.60	2/5	46315.2	3 :51 :36
pioro_20	40	178	20	3426.4	87.4	*55.55*	0/5	41249.8	5 :00 :00

Computational results - realistic instances

Instance	|V|	|A|	|K|	#msI	#msII	Gap %	nodes	TT	TT(sep)
nobel_germany	17	52	10	37	1	1.52	8	0 :00 :00	0
nobel_germany	17	52	20	400	10	13.47	80	0 :00 :04	0
nobel_germany	17	52	25	632	20	9.31	123	0 :00 :11	0
nobel_germany	17	52	30	325	18	33.06	345	0 :00 :19	0
nobel_germany	17	52	40	1088	130	30.57	232	0 :45 :03	6
nobel_germany	17	52	45	703	44	34.00	721	0 :01 :01	2
france	25	90	10	296	10	39.43	229	0 :00 :11	0
france	25	90	20	1223	137	27.80	4610	0 :14 :57	8
france	25	90	25	1748	68	17.06	3903	0 :19 :21	13
france	25	90	30	4585	951	35.04	26412	5 :00 :00	64
france	25	90	40	5763	1154	33.06	18025	5 :00 :00	69
france	25	90	45	7865	1497	61.83	18521	5 :00 :00	84
cost266	37	102	10	168	10	36.40	106	0 :00 :04	0
cost266	37	102	20	414	751	37.67	4808	1 :19 :30	35
cost266	37	102	25	2680	780	38.9	11252	5 :00 :00	91
cost266	37	102	30	2523	646	42.48	9710	5 :00 :00	85
cost266	37	102	40	4224	689	58	6505	5 :00 :00	164
cost266	37	102	45	3599	794	68.62	6439	5 :00 :00	168
zib54	54	160	10	869	111	46.62	975	0 :03 :83	4
zib54	54	160	20	4050	913	60.52	17227	5 :00 :00	49
zib54	54	160	25	5474	558	66.46	13841	5 :00 :00	57
zib54	54	160	30	4816	565	64.2	11809	5 :00 :00	58
zib54	54	160	40	3264	246	81.46	10289	5 :00 :00	85
zib54	54	160	45	5245	367	68.89	8446	5 :00 :00	93

Outline

Concluding remarks

- We have considered the UNACND problem
- In particular, we focused on the arc-set polyhedron associated with this problem
- We studied a more general family of polyhedra defined by a larger class of functions called unitary step monotonically increasing set functions
- By considering one of those functions, the Bin Packing function, we have derived valid inequalities that we applied for UNACND problem
- The Bin Packing function defines an NP-hard problem, but Min Set I and Min Set II inequalities could be efficiently separated by using fast and effective lower bounds.

Perspectives

- Extend the results concerning Min Set I and Min Set II inequalities in the context of a cutset relaxation of UNACND problem
- Improve the effectiveness of the proposed algorithms
- Use similar approaches in other Capacitated Network Design variants ...

Thank you !

Questions ... ?