On the solution of a graph partitioning problem under capacity constraints

Viet Hung Nguyen

LIP6 - University Pierre et Marie Curie Paris 6

joint work with M. Minoux, A. Guilbert, P. Bonami

PGMO's days 2013 - ENSTA ParisTech

Outline of the talk

1. Introduction
2. 0/1 quadratic models for GPCC
3. Linearization du quadratic term $x_{ui}x_{vi}$
4. Conclusions

What we are going to talk about...

1. Introduction
2. 0/1 quadratic models for GPCC
3. Linearization du quadratic term $x_{ui}x_{vi}$
4. Conclusions

Graph Partitioning under Capacity Constraints (GPCC)

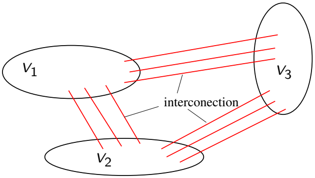
- Given a graph $G = (V, E)$ of n nodes and m edges. The edges are weighted by a vector $t \in \mathbb{R}_+^E$.

Graph Partitioning under Capacity Constraints (GPCC)

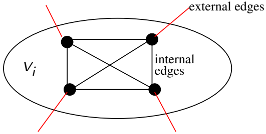
- Given a graph $G = (V, E)$ of n nodes and m edges. The edges are weighted by a vector $t \in \mathbb{R}_+^E$.
- *The Graph Partitioning Problem* to find a partition $\mathcal{P} = \{V_1, V_2, \ldots, V_p\}$ of V minimizing the total weight of the edges in the interconnection. Classical versions impose some constraints on the cardinality of the cluster.

Graph Partitioning under Capacity Constraints (GPCC)

- Given a graph $G = (V, E)$ of n nodes and m edges. The edges are weighted by a vector $t \in \mathbb{R}_+^E$.
- *The Graph Partitioning Problem* to find a partition $\mathcal{P} = \{V_1, V_2, \ldots, V_p\}$ of V minimizing the total weight of the edges in the interconnection. Classical versions impose some constraints on the cardinality of the cluster.
- *Capacity constraints.* The capacity of a cluster V_i in the partition is the sum of its internal edges and external edges. The capacity constraints bound the capacity of every cluster to a constant B.

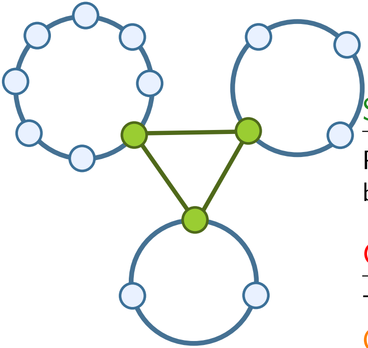


A partition $\mathcal{P} = \{V_1, V_2, V_3\}$.



Capacity of a cluster $V_i \leq B$

Application of GPCC : SONET/SDH network design



Datas

- A set of nodes $V = \{1, \ldots, n\}$ ($n \approx 50$),
- The volume of traffic t_{uv} between every pair u, v of nodes

Solutions

Partition V into rings (*local rings*) connected by a secondary federal ring.

Constraints

The capacity of local ring are bounded.

Objective

We aim to minimize the volume of traffics which transit in the interconnection.

What we are going to talk about...

Node-Cluster model [Goldschmidt et al. 2003]

The variables

For every node u and n possible clusters $i = 1, \ldots, n$,

$$x_{ui} = \begin{cases} 1, & \text{if the node } u \text{ is assigned to cluster } i \\ 0, & \text{otherwise} \end{cases}$$

Objective

Maximizing the internal traffics in clusters,

$$\max \sum_{i=1}^{n} \sum_{u=1}^{n-1} \sum_{v=u+1}^{n} t_{uv} x_{ui} x_{vi}$$

Note that the objective is quadratic

Node-Cluster model : The constraints

The capacity constraints

For all clusters $i = 1, \ldots, n$,

$$\sum_{u=1}^{n} x_{ui} W_u - \sum_{u=1}^{n-1} \sum_{v=u+1}^{n} x_{ui} x_{vi} t_{uv} \leq B$$

W_u is the total traffic incident to the node u.

The assignment constraints

Node-Cluster model : The constraints

The capacity constraints

For all clusters $i = 1, \dots, n$,

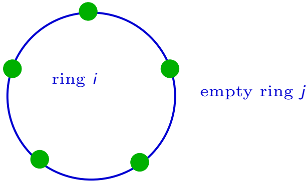
$$\sum_{u=1}^{n} x_{ui} W_u - \sum_{u=1}^{n-1} \sum_{v=u+1}^{n} x_{ui} x_{vi} t_{uv} \leq B$$

W_u is the total traffic incident to the node u.

The assignment constraints

For every node u,

$$\sum_{i=1}^{n} x_{ui} = 1$$



Drawback

High degree of symmetry : the same solution for GPCC may correspond to several distinct solutions for the model.

Node-Cluster model : The constraints

The capacity constraints

For all clusters $i = 1, \ldots, n$,

$$\sum_{u=1}^{n} x_{ui} W_u - \sum_{u=1}^{n-1} \sum_{v=u+1}^{n} x_{ui} x_{vi} t_{uv} \leq B$$

W_u is the total traffic incident to the node u.

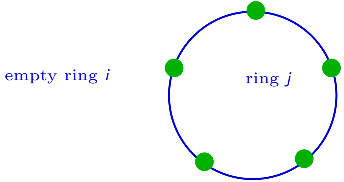
The assignment constraints

For every node u,

$$\sum_{i=1}^{n} x_{ui} = 1$$

Drawback

High degree of symmetry : the same solution for GPCC may correspond to several distinct solutions for the model.



Node-Cluster model : Breaking the symmetry

We propose two constraints which oblige that a solution for GPCC correspond to an unique solution for the model.

The order constraints

For every indices $u, i = 1 \ldots, n$,

$$x_{ui} \leq x_{ii} \;\; u = i + 1, \ldots, n$$

$$x_{ui} = 0 \;\; u = 1, \ldots, i - 1$$

Summary

The Node-Cluster model has n^2 variables, n quadratic constraints, $O(n^2)$ linear constraints and the objective is quadratic.

Node-Node model

This formulation for GPCC is quite similar to the model of Grötschel et Wakabayashi (89) for the clique partitioning problem.

The variables

Node-Node model

This formulation for GPCC is quite similar to the model of Grötschel et Wakabayashi (89) for the clique partitioning problem.

The variables

For every pairs of nodes u and v,

$$x_{uv} = \begin{cases} 0, & \text{if } u, v \text{ are assigned to the same cluster} \\ 1, & \text{otherwise} \end{cases}$$

Objective

Minimizing the traffics in the interconnection,

$$\min \sum_{e \in \mathcal{P}} t_e x_e$$

Note that the objective is linear

Node-Node model : Constraints

The transitivity (or triangle) constraints

For all triplets of nodes u, v and i,

$$x_{(u,i)} + x_{(v,i)} \geq x_{(u,v)}$$

The capacity constraints

For each node i, the capacity of the cluster containing i is bounded by B,

$$\sum_{(u,v)} t_{uv} - \sum_{(u,v)} t_{u,v} x_{ui} x_{vi} \leq B$$

Summary

The Node-Node model has n^2 variables, n quadratic constraints, $O(n^3)$ linear constraints and the objective is linear.

Summary on the models

Improved Node-Cluster model

$$\max \sum_{i=1}^{n}\sum_{u=1}^{n-1}\sum_{v=u+1}^{n} t_{uv}x_{ui}x_{vi}$$

$$\sum_{i=1}^{n} x_{ui} = 1 \ \forall \text{ nodes } u, \text{ clusters } i$$

$$x_{ui} \leq x_{ii} \ \ u = i+1,\dots,n$$

$$x_{ui} = 0 \ \ u = 1,\dots,i-1$$

$$\sum_{u=1}^{n} x_{ui}W_u - \sum_{u=1}^{n-1}\sum_{v=u+1}^{n} t_{uv}x_{ui}x_{vi} \leq B$$

n^2 variables, n quadratic constraints, $O(n^2)$ linear constraints and quadratically objective.

Node-Node model

$$\min \sum_{e\in E} t_e x_e$$

$$x_{(u,i)} + x_{(v,i)} \geq x_{(u,v)}$$

$$(u,v,i) \in \mathcal{T}$$

$$\sum_{(u,v)} t_{uv} - \sum_{(u,v)} t_{uv}x_{ui}x_{vi} \leq B$$

$$x_e \in \{0,1\} \ e \in E.$$

n^2 variables, n quadratic constraints, $O(n^3)$ linear constraints and linear objective.

What we are going to talk about...

Solution techniques for 0/1 quadratically constrained programs (I)

Fortet's linearization (FL) :

Introducing

- $O(n^3)$ continuous variables y_{uvi} to represent each product $x_{ui}x_{vi}$:
- $O(n^3)$ additional constraints.

$$\max\{0, x_{ui} + x_{vi} - 1\} \leq y_{uvi} \leq \min\{x_{ui}, x_{vi}\}, \qquad 1 \leq i < j \leq n,$$

- A drawback of Fortet's linearization is the total number of variables raises from $O(n^2)$ to $O(n^3)$. For $n \approx 50$, the number of variables grows from 2500 to 125000.

Linearization by Projection :

- Saxena et al. (2011) propose a technique which (in a simplified way)
  - dertermine if for a point x^*, there exists a y^* for that (x^*, y^*) feasible for the linear relaxation of the Fortet's linearization.
  - if such y^* does not exist, it generates valid inequalities violated by x^*.

For that, it consists in solving a linear program.

Linearization by Projection :

- Saxena et al. (2011) propose a technique which (in a simplified way)
  - dertermine if for a point x^*, there exists a y^* for that (x^*, y^*) feasible for the linear relaxation of the Fortet's linearization.
  - if such y^* does not exist, it generates valid inequalities violated by x^*.

  For that, it consists in solving a linear program.
- This technique adapted to the models of GPCC is particularly simple, i.e. not have to solve linear program, cuts generation can be done combinatorially in linear time.

Solution techniques for 0/1 quadratically constrained programs (III)

Projection technique :

For $x_{ui}, x_{vi} \in \{0,1\}$, $y_{uvi} = x_{ui}x_{vi} = \min\{x_{ui}, x_{vi}\} = \max\{x_{ui} + x_{vi} - 1, 0\}$. Hence the quadratic capacity constraint can be rewritten as :

$$\phi^i(x) = \sum_{(u,v)} t_{uv} - \sum_{(u,v)} t_{uv} \min(x_{ui}, x_{vi}) \leq B$$

$\phi^i(x)$ is a convex piecewise linear function with at most $2^{\frac{n(n-1)}{2}}$ pieces.

- No additional variables, but exponential number of linear constraints. We prove that it is as strong as the Fortet's linearization in term of continuous relaxation.
- Continuous relaxation can be solved by cutting-planes. Separation procedure is particularly easy.

Comparing the strength of the continuous relaxations

In this first experiment, we only seek to compute the continuous relaxations of the Fortet's linearization of the two models. For Node-Node model, we also implement the projection technique.

Comparing continuous relaxations

	Improved N/C		Fortet NN		proj. N/N	
n	CPU	value	CPU	value	CPU	value
16	0.05	1.13E+5	1.24	2.26E+6	0.53	2.26E+6
17	0.05	1.93E+5	1.98	3.33E+6	0.53	3.33E+6
18	0.06	1.93E+5	2.72	1.14E+6	1.10	1.14E+6
19	0.06	2.07E+5	3.90	1.88E+6	1.58	1.88E+6
20	0.06	2.26E+5	5.40	3.90E+6	2.61	3.90E+6
21	0.07	3.24E+5	7.25	5.52E+6	3.01	5.52E+6
22	0.07	4.67E+5	9.11	7.02E+6	3.84	7.02E+6
25	0.07	3.87E+5	29.3	6.13E+6	8.37	6.13E+6
all	0.06	2.64E+5	7.62	3.90E+6	2.69	3.90E+6

Three methods on Node-Node model

We compare the three methods of convexification/linearization for Node-Node models included in a branch-and-bound/cut algorithm for GPCC.

		quad. N/N			N/N			proj. N/N		
n	#in	#sol	CPU	Nodes	#sol	CPU	Nodes	#sol	CPU	Nodes
16	6	2	7979	75233	6	243	3178	6	52	6060
17	6	1	9011	244233	6	412	5827	6	82	6137
18	6	1	8919	45945	6	594	2981	6	117	10993
19	6	2	7617	26129	6	4885	22044	6	900	66267
20	5	0	10800	29919	5	1142	2454	5	336	4542
21	4	0	10800	41544	3	5083	6176	4	1388	5047
22	2	0	10800	42341	1	8683	6584	2	6175	128823
25	2	0	10800	16921	1	9254	1342	2	2869	23982
all	37	6	9585	65283	34	3787	6323	37	1490	31481

Exact Solutions : Improved Node-Cluster model vs Node-Node model

		Improved N/C			proj. N/N		
n	#inst	# sol	CPU	Nodes	# sol	CPU	Nodes
16	6	6	25	6550	6	52	6060
17	6	6	88	21910	6	82	6137
18	6	6	46	6188	6	117	10993
19	6	6	147	20385	6	900	66267
20	5	5	459	44284	5	336	4542
21	4	4	434	29571	4	1388	5047
22	2	2	2467	161504	2	6175	128823
25	2	2	498	13071	2	2869	23982
all	37	37	520	37933	37	1490	31481

TABLE: Statistics on complete solution by branch-and-cut.

Observations

- Numerical results show :
  - For Node-Nodel model : minimum eigenvalue technique $<$ Fortet's linearization $<$ projection technique.
  - The latter $<$ Fortet's linearization for Improved Node-Cluster model in exact solutions by branch-and-bound algorithms.
- Branch-and-bound because it is better to deactivate all the cut's generation subroutines of CPLEX.

Observations

- Numerical results show :
  - For Node-Nodel model : minimum eigenvalue technique $<$ Fortet's linearization $<$ projection technique.
  - The latter $<$ Fortet's linearization for Improved Node-Cluster model in exact solutions by branch-and-bound algorithms.
- Branch-and-bound because it is better to deactivate all the cut's generation subroutines of CPLEX.
- The explanation is that branching is very efficient for Node-Cluster model, the lower bound in B&B reaches very quickly to the one for Node-Node model.

Observations

- Numerical results show :
  - For Node-Nodel model : minimum eigenvalue technique $<$ Fortet's linearization $<$ projection technique.
  - The latter $<$ Fortet's linearization for Improved Node-Cluster model in exact solutions by branch-and-bound algorithms.
- Branch-and-bound because it is better to deactivate all the cut's generation subroutines of CPLEX.
- The explanation is that branching is very efficient for Node-Cluster model, the lower bound in B&B reaches very quickly to the one for Node-Node model.
- To do better than Fortet's linearization, we may need to speed up the solution of continuous relaxation at each node of B&B tree search. In other word, the "lightness" of a linearization is perhaps more important than its quality.

Sherali-Smith's linearization [Optimization Letters 2007]

- Recall of our capacity constraints :

$$\sum_{u=1}^{n} x_{ui} W_u - \sum_{u=1}^{n-1} \sum_{v=u+1}^{n} t_{uv} x_{ui} x_{vi} \leq B$$

Sherali-Smith's linearization [Optimization Letters 2007]

- Recall of our capacity constraints :

$$\sum_{u=1}^{n} x_{ui} W_u - \sum_{u=1}^{n-1} x_{ui} \sum_{v=u+1}^{n} t_{uv} x_{vi} \leq B$$

Sherali-Smith's linearization [Optimization Letters 2007]

- Recall of our capacity constraints :

$$\sum_{u=1}^{n} x_{ui} W_u - \sum_{u=1}^{n-1} x_{ui} \sum_{v=u+1}^{n} t_{uv} x_{vi} \leq B$$

- Set $\gamma_{ui} = \sum_{v=u+1}^{n} t_{uv} x_{vi}$ and we have

$$\sum_{u=1}^{n} x_{ui} W_u - \sum_{u=1}^{n-1} x_{ui} \gamma_{ui} \leq B$$

- Linearize the term $x_{ui}\gamma_{ui}$ by setting $z_{ui} = x_{ui}\gamma_{ui}$ and by determining B_{ui} over a suitable relaxation such that $0 \leq z_{ui} \leq B_{ui}$.
- Only $O(n^2)$ continuous variables added.

Computing the upper bounds B_{ui}

- $z_{ui} = x_{ui}\gamma_{ui}$ and $0 \leq z_{ui} \leq B_{ui} \Rightarrow$ more B_{ui} is tight, more the continuous relaxation is strong.
- We reduce of n^2 computation of the terms B_{ui} into only n computations.

Computing the upper bounds B_{ui}

- $z_{ui} = x_{ui}\gamma_{ui}$ and $0 \leq z_{ui} \leq B_{ui} \Rightarrow$ more B_{ui} is tight, more the continuous relaxation is strong.
- We reduce of n^2 computation of the terms B_{ui} into only n computations.
- Simple estimation of B_{ui} gives us a MIP, weaker than Fortet's linearization but more efficient when integrated in B&B algorithm.
- Improvement. B_{ui} involves only in one quadratic constraint ! $\Rightarrow$ estimation of B_{ui} can be improved by solving n problems of linear objective and one quadratic constraint.
  - A stronger MIP than Fortet's linearization.
  - Time devoted to find B_{ui} turns out to be insignificant to the total computing time particularly for the difficult instances.

Numerical results

Instance	Improved Sherali-Smith		Fortet's linearization	
	CPU	Nodes B&B	CPU	Nodes B&B
30v1	8.3	2211	19.4	5593
30v2	14.8	16860	105.2	9654
30v3	7.6	2076	21.0	733
40v1	202.6	72989	2235.0	14906
40v2	366.9	188515	7200*	48172
40v3	209.8	151315	1835.4	21647
50v1	221.2	36636	4056.6	9671
50v2	156.2	14202	19.8	1081
50v3	646.9	85575	7200*	10018

(*) Unsolved instances, limit CPU time reached.

Conclusions

- An improved Sherali-Smith linearization method for (NC) :
  - Stronger than the Fortet's linearization in term of continuous relaxation.
  - No additional continuous variables $\Rightarrow$ keeping this advantage over the Fortet's linearization.
- Computing time is reduced by 10 to 20 times in compared to the previous best known method which is the Fortet's linearization for (NC).
- Interesting case study where the model with strongest continuous relaxation is not the most efficient when integrated in Branch-and-Bound algorithms.

Conclusions

- An improved Sherali-Smith linearization method for (NC) :
  - Stronger than the Fortet's linearization in term of continuous relaxation.
  - No additional continuous variables $\Rightarrow$ keeping this advantage over the Fortet's linearization.
- Computing time is reduced by 10 to 20 times in compared to the previous best known method which is the Fortet's linearization for (NC).
- Interesting case study where the model with strongest continuous relaxation is not the most efficient when integrated in Branch-and-Bound algorithms.
- For very difficult instances, tailing-off effect is observed, branching becomes inefficient from about 15% of gap. Need cuts to break more symmetries.

Thank you for your attention !