A Multilevel Vertex Separator Algorithm Based on the Solution of Bilinear Programs

William Hager [1] James Hungerford [2] Ilya Safro [3]

[1]University of Florida

[2]M.A.I.O.R. Srl, Lucca, Italy

[3]Clemson University

October 12th, 2016
CSC16
Albuquerque, New Mexico

Outline

Outline

Notation

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ simple undirected graph

- $\mathcal{V} = \{1, 2, \ldots, n\}$
- $\mathcal{E} \subseteq (\mathcal{V} \times \mathcal{V})$; $\quad (i,j) \in \mathcal{E} \; \Leftrightarrow \; (j,i) \in \mathcal{E}$; $\quad (i,i) \notin \mathcal{E} \; \forall \, i \in \mathcal{V}$

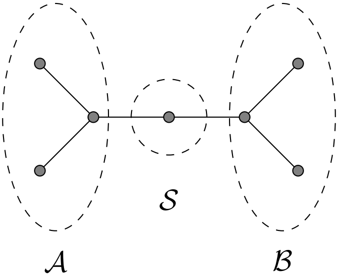
- Vertex costs: $c_i \in \mathbb{R} \; \forall \, i \in \mathcal{V}$
- Vertex weights: $w_i > 0 \; \forall \, i \in \mathcal{V}$
- For any $\mathcal{Z} \subset \mathcal{V}$,

$$\mathcal{C}(\mathcal{Z}) = \sum_{i \in \mathcal{Z}} c_i \quad \text{and} \quad \mathcal{W}(\mathcal{Z}) = \sum_{i \in \mathcal{Z}} w_i$$

Definition

Let $\mathcal{A}, \mathcal{B}, \mathcal{S} \subseteq \mathcal{V}$ be a partition of $\mathcal{V}$ such that $\mathcal{A}, \mathcal{B} \neq \emptyset$. Then $(\mathcal{A}, \mathcal{S}, \mathcal{B})$ is a *vertex separator* if

$$(\mathcal{A} \times \mathcal{B}) \cap \mathcal{E} = \emptyset.$$



Definition

Let $\mathcal{A}, \mathcal{B}, \mathcal{S} \subseteq \mathcal{V}$ be a partition of $\mathcal{V}$ such that $\mathcal{A}, \mathcal{B} \neq \emptyset$. Then $(\mathcal{A}, \mathcal{S}, \mathcal{B})$ is a *vertex separator* if

$$(\mathcal{A} \times \mathcal{B}) \cap \mathcal{E} = \emptyset.$$

Definition (Vertex Separator Problem (VSP))

Given $\ell_a, \ell_b, u_a, u_b \in \mathbb{R}_+$, find a vertex separator $(\mathcal{A}, \mathcal{S}, \mathcal{B})$ such that $\ell_a \leq \mathcal{W}(\mathcal{A}) \leq u_a$, $\ell_b \leq \mathcal{W}(\mathcal{B}) \leq u_b$, and $\mathcal{C}(\mathcal{S})$ is minimized:

$$\min_{\mathcal{A},\, \mathcal{S},\, \mathcal{B} \subseteq \mathcal{V}} \quad \mathcal{C}(\mathcal{S})$$

$$\text{subject to} \quad \mathcal{S} = \mathcal{V} \setminus (\mathcal{A} \cup \mathcal{B}), \quad \mathcal{A} \cap \mathcal{B} = \emptyset, \quad (\mathcal{A} \times \mathcal{B}) \cap \mathcal{E} = \emptyset,$$

$$\ell_a \leq \mathcal{W}(\mathcal{A}) \leq u_a, \text{ and } \ell_b \leq \mathcal{W}(\mathcal{B}) \leq u_b.$$

NP Hard [1]

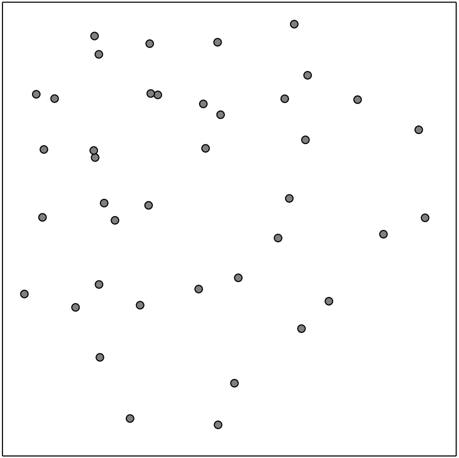
[1] T. Bui and C. Jones, *Finding good approximate vertex and edge partitions is NP-hard*, IPL 42 (1992), pp. 153–159.

Applications

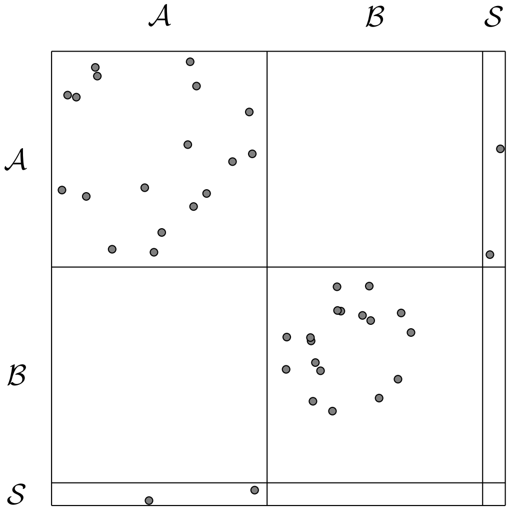
- Sparse matrix factorizations

- Cyber-security, social network analysis

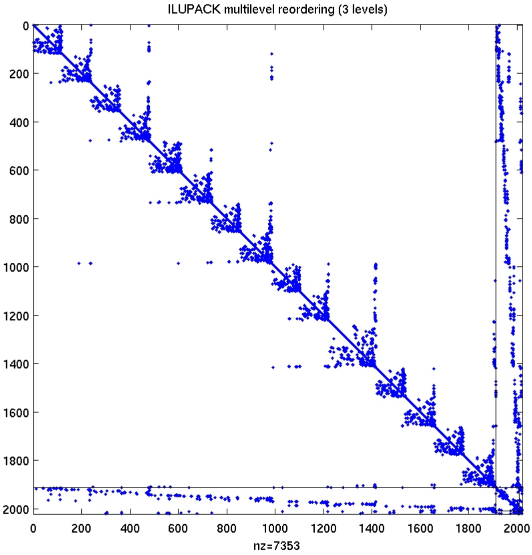
Applications



Applications



Applications



Applications

Applications

- Sparse matrix factorizations

  speed vs. solution quality

- Cyber-security, social network analysis

  speed vs. solution quality

Edge-separator based methods:

A. Pothen, H. D. Simon, and K. Liou, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 430–452.

J. R. Gilbert and E. Zmijewski, *A parallel graph partitioning algorithm for a message-passing multiprocessor*, Intl. J. Parallel Programming, 16 (1987), pp. 498–513.

S. Acer, E. Kayaaslan, and C. Aykanat, *A recursive bipartitioning algorithm for permuting sparse square matrices into block diagonal form with overlap*, SIAM Journal on Scientific Computing, 35 (2013), pp. C99–C121.

Software/related work

Direct methods:

C. C. Ashcraft and J. W. H. Liu, *A partition improvement algorithm for generalized nested dissection*, Tech. Rep. BCSTECH-94-020, Boeing Computer Services, Seattle, WA, 1994.

E. Balas and C. C. de Souza, *The vertex separator problem: a polyhedral investigation*, Math. Program., 103 (2005), pp. 583–608.

U. Benlic and J. Hao, *Breakout local search for the vertex separator problem*, in Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, 2013.

U. Feige, M. Hajiaghayi, and J. Lee, *Improved approximation algorithms for vertex separators*, SIAM J. Comput., 38 (2008), pp. 629–657.

C. Leiserson and J. Lewis, *Orderings for parallel sparse symmetric factorization*, in Third SIAM Conference on Parallel Processing for Scientific Computing, SIAM Publications, 1987, pp. 27–31.

Software/related work

- MeTiS
- BEND
- SCOTCH
- KaHIP
- Hypergraphs: hMeTiS, PaToH, Mondriaan, Parkway, Zoltan, ...

Outline

Combinatorial formulation

$$\min_{\mathcal{A},\,\mathcal{S},\,\mathcal{B}\subseteq\mathcal{V}} \quad \mathcal{C}(\mathcal{S})$$

$$\text{subject to} \quad \mathcal{S} = \mathcal{V} \setminus (\mathcal{A} \cup \mathcal{B}), \quad \mathcal{A} \cap \mathcal{B} = \emptyset, \quad (\mathcal{A} \times \mathcal{B}) \cap \mathcal{E} = \emptyset,$$

$$\ell_a \leq \mathcal{W}(\mathcal{A}) \leq u_a, \;\; \text{and} \;\; \ell_b \leq \mathcal{W}(\mathcal{B}) \leq u_b$$

$$\max_{\mathcal{A},\, \mathcal{B} \subseteq \mathcal{V}} \quad \mathcal{C}(\mathcal{A} \cup \mathcal{B})$$

$$\text{subject to} \quad \mathcal{A} \cap \mathcal{B} = \emptyset, \quad (\mathcal{A} \times \mathcal{B}) \cap \mathcal{E} = \emptyset,$$

$$\ell_a \leq \mathcal{W}(\mathcal{A}) \leq u_a, \ \text{ and } \ \ell_b \leq \mathcal{W}(\mathcal{B}) \leq u_b$$

$$\max_{\mathcal{A},\, \mathcal{B} \subseteq \mathcal{V}} \quad \mathcal{C}(\mathcal{A} \cup \mathcal{B})$$

$$\text{subject to} \quad \mathcal{A} \cap \mathcal{B} = \emptyset, \quad (\mathcal{A} \times \mathcal{B}) \cap \mathcal{E} = \emptyset,$$

$$\ell_a \le \mathcal{W}(\mathcal{A}) \le u_a, \;\; \text{and} \;\; \ell_b \le \mathcal{W}(\mathcal{B}) \le u_b$$

$$x_i = \begin{cases} 1, & \text{if } i \in \mathcal{A} \\ 0, & \text{if } i \notin \mathcal{A} \end{cases} \quad \text{and} \quad y_i = \begin{cases} 1, & \text{if } i \in \mathcal{B} \\ 0, & \text{if } i \notin \mathcal{B} \end{cases} \quad \forall\, i \in \mathcal{V}$$

$$\ell_a \le \mathbf{w}^\mathsf{T}\mathbf{x} \le u_a, \quad \ell_b \le \mathbf{w}^\mathsf{T}\mathbf{y} \le u_b$$

$$\mathbf{c}^\mathsf{T}(\mathbf{x} + \mathbf{y}) \;\leftarrow\; \mathcal{C}(\mathcal{A} \cup \mathcal{B})$$

$$\mathbf{A}_{ij} \in \{0,1\}, \quad \mathbf{A}_{ij} = 1 \;\Leftrightarrow\; (i,j) \in \mathcal{E}$$

$$\begin{aligned}
\mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{y} &= \sum_{i=1}^{n}\sum_{j=1}^{n} x_i a_{ij} y_j = \sum_{x_i=1}\sum_{y_j=1} a_{ij} \\
&= \sum_{i\in\mathcal{A}}\sum_{j\in\mathcal{B}} a_{ij} = |(\mathcal{A}\times\mathcal{B})\cap\mathcal{E}|.
\end{aligned}$$

$$\begin{aligned}
\mathbf{x}^{\mathsf{T}}\mathbf{y} &= \sum_{i=1}^{n} x_i y_i = \sum_{x_i=y_i=1} 1 \\
&= |\mathcal{A}\cap\mathcal{B}|
\end{aligned}$$

$$\mathbf{x}^{\mathsf{T}}(\mathbf{A}+\mathbf{I})\mathbf{y} = 0 \quad\Leftrightarrow\quad \left\{ \begin{array}{c} \mathcal{A}\cap\mathcal{B} = \emptyset \\ \text{and} \\ (\mathcal{A}\times\mathcal{B})\cap\mathcal{E} = \emptyset \end{array} \right.$$

Integer formulation

$$\begin{aligned} \max \quad & \mathbf{c}^{\mathsf{T}}(\mathbf{x}+\mathbf{y}) \\ \text{subject to} \quad & \mathbf{x}^{\mathsf{T}}(\mathbf{A}+\mathbf{I})\mathbf{y} = 0, \\ & \ell_a \le \mathbf{w}^{\mathsf{T}}\mathbf{x} \le u_a, \quad \ell_b \le \mathbf{w}^{\mathsf{T}}\mathbf{y} \le u_b, \\ & \mathbf{x}, \mathbf{y} \in \{0,1\}^n \end{aligned}$$

Integer formulation

$\gamma \in \mathbb{R}$ (penalty parameter)

$$\begin{aligned} \max \quad & f(\mathbf{x}, \mathbf{y}) := \mathbf{c}^{\mathsf{T}}(\mathbf{x} + \mathbf{y}) - \gamma \mathbf{x}^{\mathsf{T}}(\mathbf{A} + \mathbf{I})\mathbf{y} \\ \text{subject to} \quad & \ell_a \le \mathbf{w}^{\mathsf{T}}\mathbf{x} \le u_a, \quad \ell_b \le \mathbf{w}^{\mathsf{T}}\mathbf{y} \le u_b, \\ & \mathbf{x}, \mathbf{y} \in \{0, 1\}^n \end{aligned} \tag{1}$$

Integer formulation

Proposition

If $\mathbf{w} \geq \mathbf{1}$ *and* $\gamma > 0$ *with* $\gamma \geq \max\{c_i : i \in \mathcal{V}\}$, *then for any feasible point* $(\mathbf{x}, \mathbf{y})$ *in* (1) *satisfying*

$$f(\mathbf{x}, \mathbf{y}) \geq \gamma(\ell_a + \ell_b),$$

there is a feasible point $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ *in* (1) *such that*

$$f(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \geq f(\mathbf{x}, \mathbf{y}) \quad \text{and} \quad \bar{\mathbf{x}}^{\mathsf{T}}(\mathbf{A} + \mathbf{I})\bar{\mathbf{y}} = 0.$$

Hence, if the optimal objective value in (1) *is at least* $\gamma(\ell_a + \ell_b)$, *then there exists an optimal solution* $(\mathbf{x}^*, \mathbf{y}^*)$ *to* (1) *such that an optimal vertex separator is given by*

$$\mathcal{A} = \{i : x_i^* = 1\}, \ \ \mathcal{B} = \{i : y_i^* = 1\}, \ \ \text{and} \ \ \mathcal{S} = \{i : x_i^* = y_i^* = 0\}.$$

Continuous formulation

$$\begin{array}{lll} \max & f(\mathbf{x},\mathbf{y}) & \left[= \mathbf{c}^{\mathsf{T}}(\mathbf{x}+\mathbf{y}) - \gamma \mathbf{x}^{\mathsf{T}}(\mathbf{A}+\mathbf{I})\mathbf{y}\right] \\ \text{subject to} & \ell_a \leq \mathbf{w}^{\mathsf{T}}\mathbf{x} \leq u_a, \quad \ell_b \leq \mathbf{w}^{\mathsf{T}}\mathbf{y} \leq u_b, & \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{1}, \quad \mathbf{0} \leq \mathbf{y} \leq \mathbf{1}, \quad \mathbf{x},\mathbf{y} \in \mathbb{R}^n & \end{array} \tag{2}$$

Theorem (Theorem 2.1, Part 1 [1])

Suppose $\mathbf{w} = \mathbf{1}$, $\mathbf{c} \geq \mathbf{0}$, $\gamma \geq \max\{c_i : i \in \mathcal{V}\} > 0$, *and*

$$\mathcal{C}(\mathcal{S}^*) \leq \mathcal{C}(\mathcal{V}) - \gamma(\ell_a + \ell_b).$$

Then (2) *has a binary optimal solution* $(\mathbf{x}^*, \mathbf{y}^*) \in \{0,1\}^{2n}$ *satisfying* $\mathbf{x}^{*\mathsf{T}}(\mathbf{A}+\mathbf{I})\mathbf{y}^* = 0$. *Hence, an optimal vertex separator is given by*

$$\mathcal{A} = \{i : x_i^* = 1\}, \;\; \mathcal{B} = \{i : y_i^* = 1\}, \;\; \text{and} \;\; \mathcal{S} = \{i : x_i^* = y_i^* = 0\}.$$

[1] W. W. Hager and J.T. Hungerford, "Continuous quadratic programming formulations of optimization problems on graphs", *European Journal of Operations Research, 2014.*

Continuous formulation

$$\begin{aligned} \max \quad & f(\mathbf{x},\mathbf{y}) \quad \left[= \mathbf{c}^{\mathsf{T}}(\mathbf{x}+\mathbf{y}) - \gamma \mathbf{x}^{\mathsf{T}}(\mathbf{A}+\mathbf{I})\mathbf{y}\right] \\ \text{subject to} \quad & \ell_a \le \mathbf{w}^{\mathsf{T}}\mathbf{x} \le u_a, \quad \ell_b \le \mathbf{w}^{\mathsf{T}}\mathbf{y} \le u_b, \\ & \mathbf{0} \le \mathbf{x} \le \mathbf{1}, \quad \mathbf{0} \le \mathbf{y} \le \mathbf{1}, \quad \mathbf{x},\mathbf{y} \in \mathbb{R}^n \end{aligned} \tag{2}$$

Definition

A point $(\mathbf{x},\mathbf{y}) \in \mathbb{R}^{2n}$ is called *mostly binary* if $\mathbf{x}$ and $\mathbf{y}$ each have at most one non-binary component.

Proposition

For any $\mathbf{w} > \mathbf{0}$, $\mathbf{c} \in \mathbb{R}$, *and* $\gamma \in \mathbb{R}$, (2) *has a mostly binary optimal solution.*

Local optimality

Definition

Let $(\mathcal{A}, \mathcal{S}, \mathcal{B})$ be a feasible vertex separator.

1. For every $i \in \mathcal{V}$, let

$$\mathrm{gain}(i, \mathcal{A}) = \begin{cases} 0, & \text{if } i \in \mathcal{A} \\ c_i - \sum_{j \in \overline{\mathcal{N}}(i) \cap \mathcal{B}} c_j, & \text{if } i \in \mathcal{S} \cup \mathcal{B} \end{cases},$$

$$\mathrm{gain}(i, \mathcal{B}) = \begin{cases} 0, & \text{if } i \in \mathcal{B} \\ c_i - \sum_{j \in \overline{\mathcal{N}}(i) \cap \mathcal{A}} c_j. & \text{if } i \in \mathcal{S} \cup \mathcal{A} \end{cases}.$$

2. $(\mathcal{A}, \mathcal{S}, \mathcal{B})$ is strongly FM-optimal if $\mathrm{gain}(i, \mathcal{A}) \leq 0$ and $\mathrm{gain}(i, \mathcal{B}) \leq 0$ for every $i \in \mathcal{V}$.

Local optimality

Proposition

Suppose that $\mathbf{c} \geq \mathbf{0}$ *and* $\gamma \geq \max\{c_i : i \in \mathcal{V}\}$, *and let* $(\mathcal{A}, \mathcal{S}, \mathcal{B})$ *be a feasible vertex separator. If* $(\mathcal{A}, \mathcal{S}, \mathcal{B})$ *is strongly FM-optimal, then the pair of incidence vectors* $(\mathbf{x}, \mathbf{y})$ *for* $\mathcal{A}$ *and* $\mathcal{B}$ *is locally optimal in* (2).

Outline

Mountain Climbing Algorithm

Input: A feasible point $(\mathbf{x}, \mathbf{y})$ and $\eta > 0$.
while ($(\mathbf{x}, \mathbf{y})$ not a first-order maximizer)
  $\hat{\mathbf{x}} \leftarrow$ argmax $\{f(\mathbf{x}, \mathbf{y}) : \mathbf{x} \in \mathcal{P}_a\}$
  $\hat{\mathbf{y}} \leftarrow$ argmax $\{f(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in \mathcal{P}_b\}$
  if ($f(\hat{\mathbf{x}}, \hat{\mathbf{y}}) > \max\{f(\hat{\mathbf{x}}, \mathbf{y}), f(\mathbf{x}, \hat{\mathbf{y}})\} + \eta$)
    $(\mathbf{x}, \mathbf{y}) \leftarrow (\hat{\mathbf{x}}, \hat{\mathbf{y}})$
  else if ($f(\hat{\mathbf{x}}, \mathbf{y}) > f(\mathbf{x}, \hat{\mathbf{y}})$)
    $\mathbf{x} \leftarrow \hat{\mathbf{x}}$
  else
    $\mathbf{y} \leftarrow \hat{\mathbf{y}}$
  end if
end while
return $(\mathbf{x}, \mathbf{y})$

Algorithm : **MCA:** A modified version of Konno's Mountain Climbing Algorithm.

Mountain Climbing Algorithm

Proposition (Konno [1])

Let $(\mathbf{x}, \mathbf{y})$ *be a feasible initial guess. Then* MCA *terminates in a finite number of iterations, returning a point which satisfies the first-order optimality conditions.*

[1] H. Konno, *A cutting plane algorithm for solving bilinear programs*, Mathematical Programming, 11 (1976), pp. 14–27.

Escaping first-order optima

Escaping first-order optima

A feasible point $(\mathbf{x}, \mathbf{y})$ is a first-order maximizer if and only if there exist multipliers

$$\boldsymbol{\mu}^a \in \mathcal{M}(\mathbf{x}),\ \boldsymbol{\mu}^b \in \mathcal{M}(\mathbf{y}),\ \lambda^a \in \mathcal{L}(\mathbf{x}, \ell_a, u_a),\ \lambda^b \in \mathcal{L}(\mathbf{y}, \ell_b, u_b)$$

such that

$$\begin{bmatrix} \nabla_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) \\ \nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}) \end{bmatrix} + \begin{bmatrix} \boldsymbol{\mu}^a \\ \boldsymbol{\mu}^b \end{bmatrix} + \begin{bmatrix} \lambda^a \mathbf{w} \\ \lambda^b \mathbf{w} \end{bmatrix} = \mathbf{0},$$

where

$$\begin{aligned} \mathcal{M}(\mathbf{z}) &= \{\boldsymbol{\mu} \in \mathbb{R}^n : \mu_i z_i \le \min\{\mu_i, 0\} \text{ for all } 1 \le i \le n\} \quad \text{and} \\ \mathcal{L}(\mathbf{z}, \ell, u) &= \{\lambda \in \mathbb{R} : \lambda \mathbf{w}^\mathsf{T} \mathbf{z} \le \min\{\lambda u, \lambda \ell\}\}. \end{aligned}$$

Escaping first-order optima

Proposition

Suppose $\gamma \in \mathbb{R}$, and let $(\mathbf{x}, \mathbf{y})$ be a feasible point which satisfies the first-order optimality conditions. Let $\epsilon > 0$ and define $\tilde{\mathbf{c}}$ in any of the following ways:

1. *For any $i \neq j$ such that $\mu_i^a = \mu_j^a = 0$, $x_i < 1$, and $x_j > 0$, take*
$$\tilde{c}_k = \begin{cases} c_k + \epsilon, & \text{if } k = i, \\ c_k - \epsilon, & \text{if } k = j, \\ c_k, & \text{otherwise} \end{cases}.$$
2. *If $\lambda^a = 0$ and $\mathbf{w}^\mathsf{T}\mathbf{x} < u_a$, then for any i such that $\mu_i^a = 0$ and $x_i < 1$, take $\tilde{c}_i = c_i + \epsilon$ and $\tilde{c}_k = c_k$ for $k \neq i$.*
3. *If $\lambda^a = 0$ and $\mathbf{w}^\mathsf{T}\mathbf{x} > \ell_a$, then for any j such that $\mu_j^a = 0$ and $x_j > 0$, take $\tilde{c}_j = c_j - \epsilon$ and $\tilde{c}_k = c_k$ for $k \neq j$.*

Then, there exists a feasible direction $\mathbf{d} \in \cup_{i,j=1}^n \{\pm\mathbf{e}_i, \mathbf{e}_i - \mathbf{e}_j\}$ such that
$$f_{\tilde{\mathbf{c}}}(\mathbf{x} + \mathbf{d}, \mathbf{y}) > f_{\tilde{\mathbf{c}}}(\mathbf{x}, \mathbf{y}).$$

Input: A feasible point $(\mathbf{x}, \mathbf{y})$.
$(\mathbf{x}, \mathbf{y}) \leftarrow$ MCA $(\mathbf{x}, \mathbf{y})$
loop
  $\tilde{\mathbf{c}} \leftarrow$ perturb $(\mathbf{c})$
  $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \leftarrow$ MCA $(\mathbf{x}, \mathbf{y}, \tilde{\mathbf{c}})$
  $(\mathbf{x}^*, \mathbf{y}^*) \leftarrow$ MCA $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \mathbf{c})$
  if ($f(\mathbf{x}^*, \mathbf{y}^*) > f(\mathbf{x}, \mathbf{y})$)
    $(\mathbf{x}, \mathbf{y}) \leftarrow (\mathbf{x}^*, \mathbf{y}^*)$
  else
    break
  end if
end loop
return $(\mathbf{x}, \mathbf{y})$

Algorithm : **MCA_CP:** A modification of MCA incorporating $\mathbf{c}$-perturbations.

Escaping first-order optima

Proposition

Let $\gamma \in \mathbb{R}$, $(\mathbf{x}, \mathbf{y})$ *be a feasible point in* (2) *which satisfies the first-order optimality condition, and let* $\tilde{\gamma} \leq \gamma$. *Then* $(\mathbf{x}, \mathbf{y})$ *is first-order optimal w.r.t.* $f_{\tilde{\gamma}}$ *if and only if*

$$\tilde{\gamma} \;\geq\; \alpha \;:=\; \begin{cases} \alpha_1, & \text{if } \ell_a < \mathbf{w}^\mathsf{T}\mathbf{x} < u_a, \\ \alpha_2, & \text{if } \mathbf{w}^\mathsf{T}\mathbf{x} = u_a \end{cases} \quad ,$$

where

$$\alpha_1 \;=\; \max \left\{ \frac{c_j}{(\mathbf{A}+\mathbf{I})_j \mathbf{y}} \,:\; x_j < 1 \text{ and } (\mathbf{A}+\mathbf{I})_j \mathbf{y} > 0 \right\}$$

$$\alpha_2 \;=\; \inf \left\{ \alpha \in \mathbb{R} : \frac{1}{w_i} \frac{\partial f_\alpha}{\partial x_i}(\mathbf{x}, \mathbf{y}) \leq \frac{1}{w_j} \frac{\partial f_\alpha}{\partial x_j}(\mathbf{x}, \mathbf{y}) \;\forall\; x_i < 1 \text{ and } x_j > 0 \right\}.$$

Escaping first-order optima

Input: A feasible point $(\mathbf{x}, \mathbf{y})$.
$(\mathbf{x}, \mathbf{y}) \leftarrow$ MCA_CP $(\mathbf{x}, \mathbf{y})$
$\tilde{\gamma} \leftarrow \alpha_1$
while ($\tilde{\gamma} > 0$)
    $\tilde{\gamma} \leftarrow$ reduce $(\tilde{\gamma})$
    $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \leftarrow$ MCA_CP $(\mathbf{x}, \mathbf{y}, \tilde{\gamma})$
    $(\mathbf{x}^*, \mathbf{y}^*) \leftarrow$ MCA_CP $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \gamma)$
    if ($f(\mathbf{x}^*, \mathbf{y}^*) > f(\mathbf{x}, \mathbf{y})$)
        $(\mathbf{x}, \mathbf{y}) \leftarrow (\mathbf{x}^*, \mathbf{y}^*)$
        $\tilde{\gamma} \leftarrow \alpha_1$
end while
return $(\mathbf{x}, \mathbf{y})$

Algorithm : **MCA_GR:** MCA + $\mathbf{c}$-perturbations and γ-refinements.

Converting to a vertex separator

Converting to a vertex separator

Input: A feasible point $(\mathbf{x}, \mathbf{y})$.
while ($\mathbf{x}$ has at least 2 nonbinary components)
    Choose $i, j \in \mathcal{V}$ such that $x_i, x_j \in (0, 1)$.
    Update $\mathbf{x} \leftarrow \mathbf{x} + t(\frac{1}{w_i}\mathbf{e}_i - \frac{1}{w_j}\mathbf{e}_j)$, choosing t to ensure that:
        (a) $f(\mathbf{x}, \mathbf{y})$ does not decrease,
        (b) either $x_i \in \{0, 1\}$ or $x_j \in \{0, 1\}$,
        (c) $\mathbf{x}$ feasible.
end while
while ($\mathbf{y}$ has at least 2 nonbinary components)
    Choose $i, j \in \mathcal{V}$ such that $y_i, y_j \in (0, 1)$.
    Update $\mathbf{y} \leftarrow \mathbf{y} + t(\frac{1}{w_i}\mathbf{e}_i - \frac{1}{w_j}\mathbf{e}_j)$, choosing t to ensure that:
        (a) $f(\mathbf{x}, \mathbf{y})$ does not decrease,
        (b) either $y_i \in \{0, 1\}$ or $y_j \in \{0, 1\}$,
        (c) $\mathbf{y}$ feasible.
end while

Algorithm : **C2B**: Convert a feasible point into a mostly binary feasible point without decreasing the objective value.

Converting to a vertex separator

Input: A binary feasible point $(\mathbf{x}, \mathbf{y})$ satisfying $f(\mathbf{x}, \mathbf{y}) \geq \gamma(\ell_a + \ell_b)$
while ($\mathbf{x}^\mathsf{T}(\mathbf{A} + \mathbf{I})\mathbf{y} > 0$)
  if ($\mathbf{1}^\mathsf{T}\mathbf{x} > \ell_a$)
    Choose i such that $x_i = 1$ and $(\mathbf{A} + \mathbf{I})_i\mathbf{y} \geq 1$.
    Set $x_i = 0$.
  else if ($\mathbf{1}^\mathsf{T}\mathbf{y} > \ell_b$)
    Choose i such that $y_i = 1$ and $(\mathbf{A} + \mathbf{I})_i\mathbf{x} \geq 1$.
    Set $y_i = 0$.
  end if
end while

Algorithm : **B2S**: Convert a binary feasible point into a vertex separator without decreasing the objective function value.

Outline

Multilevel Scheme

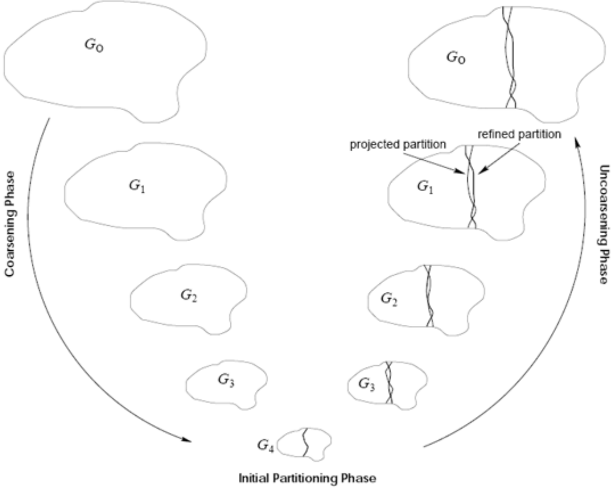
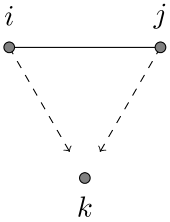


Figure : From *A fast and high quality multilevel scheme for partitioning irregular graphs*, Karypis and Kumar, S.J.S.C., 1998

Coarsening

- Vertices visited in sequence, matched to an unmatched neighbor
- Random matching, heavy-edge matching, algebraic dist. matching
- Multiple edges combined, assigned edge weights
- Degree $0/1$ vertices always matched
- Coarsen until: < 75 nodes or < 10 edges



Solving

- Compute initial guess:

$$x_i = \frac{u_a}{\mathcal{W}(\mathcal{V})},\ y_i = \frac{u_b}{\mathcal{W}(\mathcal{V})} \quad \forall\, i \in \mathcal{V}$$

- Refine:

$$(\mathbf{x}, \mathbf{y}) \ \leftarrow\ \mathrm{MCA\_GR}(\mathbf{x}, \mathbf{y})$$

- Obtain binary solution:

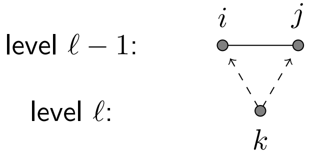
$$(\mathbf{x}, \mathbf{y}) \ \leftarrow\ \mathrm{C2B}(\mathbf{x}, \mathbf{y})$$

- Convert to vertex separator:

$$(\mathbf{x}, \mathbf{y}) \ \leftarrow\ \mathrm{B2S}(\mathbf{x}, \mathbf{y})$$

Uncoarsening and refinement

- Unmatch vertices



- Preserve memberships in sets $\mathcal{A}$, $\mathcal{S}$, and $\mathcal{B}$:

$$(x_i^{\ell-1}, y_i^{\ell-1}) = (x_j^{\ell-1}, y_j^{\ell-1}) = (x_k^{\ell}, y_k^{\ell})$$

- Refine:

$$(\mathbf{x}^{\ell-1}, \mathbf{y}^{\ell-1}) \leftarrow \text{MCA\_GR}(\mathbf{x}^{\ell-1}, \mathbf{y}^{\ell-1})$$

- Convert to vertex separator (optional):

$$(\mathbf{x}, \mathbf{y}) \leftarrow \text{B2S}(\text{C2B}(\mathbf{x}, \mathbf{y}))$$

Outline

Implementation (BLP)

- C++ (6762 lines)
- Test set: 59 sparse graphs ($n \in [1,000,\ 1,965,206]$)
  UFSML, SNAP, "Hard", peer-to-peer, Konect
- Node labels permuted
- $c_i = 1\ \forall\ i \in \mathcal{V}$, $\ell_a = \ell_a = 1$, and $u_a = u_b = \lfloor 0.6n \rfloor$
- Comparisons made with MeTiS 5.1.0.
  - Routine: METIS_ComputeVertexSeparator
  - Options: METIS_IPTYPE_NODE, METIS_RTYPE_SEP2SIDED

Test instances

Graph	$\|\mathcal{V}\|$	$\|\mathcal{E}\|/2$	Sparsity	Degree Min	Degree Max	Degree Ave
bcspwr09	1723	2394	1.61E-03	1	14	2.78
bcsstk17	10974	208838	3.47E-03	0	149	38.06
c-38	8127	34781	1.05E-03	1	888	8.56
c-43	11125	56275	9.09E-04	1	2619	10.12
crystm01	4875	50232	4.23E-03	7	26	20.61
delaunay_n13	8192	24547	7.32E-04	3	12	5.99
Erdos992	6100	7515	4.04E-04	0	61	2.46
fxm3_6	5026	44500	3.52E-03	3	128	17.71
G42	2000	11779	5.89E-03	4	249	11.78
jagmesh7	1138	3156	4.88E-03	3	6	5.55
lshp3466	3466	10215	1.70E-03	3	6	5.89
minnesota	2642	3303	9.47E-04	1	5	2.50
nasa4704	4704	50026	4.52E-03	5	41	21.27
net25	9520	195840	4.32E-03	2	138	41.14
netscience	1589	2742	2.17E-03	0	34	3.45
netz4504	1961	2578	1.34E-03	2	8	2.63
sherman1	1000	1375	2.75E-03	0	6	2.75
sstmodel	3345	9702	1.73E-03	0	17	5.80
USpowerGrid	4941	6594	5.40E-04	1	19	2.67
yeast	2361	6646	2.39E-03	0	64	5.63

Table : UFSML Graphs [1].

[1] T. A. Davis, *University of Florida sparse matrix collection*, 1994. NA Digest, vol 92, no 42.

Test instances

				Degree		
Graph	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert/2$	Sparsity	Min	Max	Ave
ca-HepPh	7241	202194	7.71E-03	2	982	55.85
email-Enron	9660	224896	4.82E-03	2	2532	46.56
email-EuAll	16805	76156	5.39E-04	1	3360	9.06
oregon2_010505	5441	19505	1.32E-03	1	1888	7.17
soc-Epinions1	22908	389439	1.48E-03	1	3026	34
web-NotreDame	56429	235285	1.48E-04	1	6852	8.34
web-Stanford	122749	1409561	1.87E-04	1	35053	22.97
wiki-Vote	3809	95996	1.32E-02	1	1167	50.40

Table : Heavy-tailed degree distribution graphs from the SNAP database [1].

[1] J. Leskovec and A. Krevl, *SNAP Datasets: Stanford large network dataset collection*. http://snap.stanford.edu/data, June 2014.

Test instances

				Degree		
Graph	$\|\mathcal{V}\|$	$\|\mathcal{E}\|/2$	Sparsity	Min	Max	Ave
barth5_1Ksep_50in_5Kout	32212	101805	1.96E-04	1	22	6.32
bcsstk30_500sep_10in_1Kout	58348	2016578	1.18E-03	0	219	69.12
befref_fxm_2_4_air02	14109	98224	9.87E-04	1	1531	13.92
bump2_e18_aa01_model1_crew1	56438	300801	1.89E-04	1	604	10.66
c-30_data_data	11023	62184	1.02E-03	1	2109	11.28
c-60_data_cti_cs4	85830	241080	6.55E-05	1	2207	5.62
data_and_seymourl	9167	55866	1.33E-03	1	229	12.19
finan512_scagr7-2c_rlfddd	139752	552020	5.65E-05	1	669	7.90
mod2_pgp2_slptsk	101364	389368	7.58E-05	1	1901	7.68
model1_crew1_cr42_south31	45101	189976	1.87E-04	1	17663	8.42
msc10848_300sep_100in_1Kout	21996	1221028	5.05E-03	1	722	111.02
p0291_seymourl_iiasa	10498	53868	9.78E-04	1	229	10.26
sctap1-2b_and_seymourl	40174	140831	1.75E-04	1	1714	7.01
south31_slptsk	39668	189914	2.41E-04	1	17663	9.58
vibrobox_scagr7-2c_rlfddd	77328	435586	1.46E-04	1	669	11.27

Table : "Hard" graphs [1].

[1] I. Safro, P. Sanders, and C. Schulz, *Advanced coarsening schemes in graph partitioning*, in Symposium on Experimental Algorithms, LNCS, vol. 7276, 2012, pp. 369–380.

Test instances

				Degree		
Graph	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert/2$	Sparsity	Min	Max	Ave
p2p-Gnutella04	10879	39994	6.76E-04	0	103	7.35
p2p-Gnutella05	8846	31839	8.14E-04	1	88	7.20
p2p-Gnutella06	8717	31525	8.30E-04	1	115	7.23
p2p-Gnutella08	6301	20777	1.05E-03	1	97	6.59
p2p-Gnutella09	8114	26013	7.90E-04	1	102	6.41
p2p-Gnutella24	26518	65369	1.86E-04	1	355	4.93
p2p-Gnutella25	22687	54705	2.13E-04	1	66	4.82
p2p-Gnutella30	36682	88328	1.31E-04	1	55	4.82
p2p-Gnutella31	62586	147892	7.55E-05	1	95	4.73

Table : Peer-to-peer networks [1].

				Degree		
Graph	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert/2$	Sparsity	Min	Max	Ave
out.as-skitter	1696415	11095298	7.71E-06	1	35455	13.08
out.com-amazon	334863	925872	1.65E-05	1	549	5.53
out.com-dblp	317080	1049866	2.09E-05	1	343	6.62
out.com-youtube	1134890	2987624	4.64E-06	1	28754	5.27
out.roadNet-CA	1965206	2766607	1.43E-06	1	12	2.82
out.roadNet-PA	1088092	1541898	2.60E-06	1	9	2.83
out.roadNet-TX	1379917	1921660	2.02E-06	1	12	2.79

Table : Konect graphs [2].

[1] M. Ripeanu, I. Foster, and A. Iamnitchi, *Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design*, IEEE Internet Computing Journal, 6 (2002), pp. 50–57.

[2] *The Koblenz Network collection*. Online reference at http://konect.uni-koblenz.de/

1. Refinement experiment

Part A:

$(\mathbf{x}, \mathbf{y}) \leftarrow$ METIS_ComputeVertexSeparator

$(\mathbf{x}, \mathbf{y}) \leftarrow$ B2S (C2B (MCA_GR $(\mathbf{x}, \mathbf{y})$))

Part B:

$(\mathbf{x}, \mathbf{y}) \leftarrow$ BLP

$(\mathbf{x}, \mathbf{y}) \leftarrow$ METIS_NodeRefine $(\mathbf{x}, \mathbf{y})$

Graph Type	A: MCA_GR			B: METIS_NodeRefine		
	avg	min	max	avg	min	max
UF	0.02	0.00	0.22	0.03	0.00	0.38
HTDD	0.36	0.00	2.06	0.06	0.00	0.55
Hard	0.07	0.00	0.54	0.16	0.00	1.64
p2p	0.49	0.03	1.20	0.27	0.04	0.58
Total	0.16	0.00	2.06	0.12	0.00	1.64

Table : Percent improvement in separator sizes

2. Multilevel experiment

		% Improvement		
Graph Type	% BLP Wins	avg	min	max
UF	35.00	0.10	-0.62	2.00
p2p	100.00	3.52	2.13	4.39
HTDD	62.50	0.84	-0.20	2.80
Hard	73.33	0.96	-0.63	8.33
Konect	14.29	-0.09	-0.60	0.96
Total	55.93	0.92	-0.63	8.33

Table : BLP_RM vs. METIS_RM.

		% Improvement		
Graph Type	% BLP Wins	avg	min	max
UF	30.00	-0.02	-0.57	1.16
p2p	100.00	2.59	0.65	3.44
HTDD	50.00	0.67	-0.30	2.41
Hard	46.67	0.38	-1.55	5.58
Total	50.00	0.65	-1.55	5.58

Table : BLP_HE vs. METIS_HE.

	BLP_RM				METIS_RM			
Graph Type	avg	geomean	min	max	avg	geomean	min	max
UF	0.56	0.34	0.03	3.08	0.01	0.00	0.00	0.12
p2p	36.08	14.99	1.48	276.09	0.16	0.13	0.05	0.49
HTDD	18.34	7.64	0.64	104.82	0.13	0.08	0.01	0.55
Hard	88.59	32.23	1.58	719.48	0.28	0.21	0.05	0.84
Konect	9258.19	5989.74	689.58	27888.17	334.55	4.97	0.63	2702.44
Total	1264.62	10.44	0.03	27888.17	44.72	0.00	0.00	2702.44

Table : CPU times (in seconds).

2. Multilevel experiment

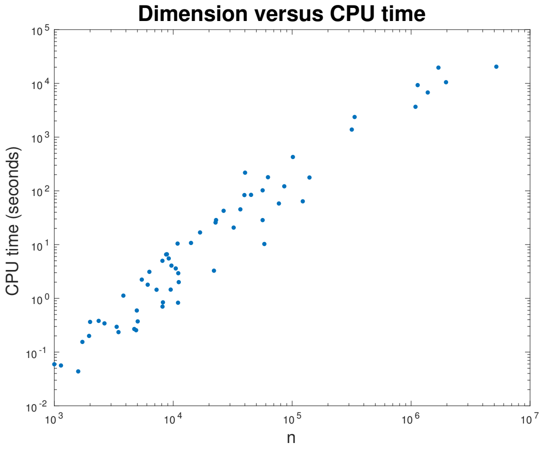


Figure : Number of vertices n versus CPU time for BLP_RM.

Possible speed-ups

1. Instead of resolving the linear program in MCA from scratch, we could exploit the structure of the previously computed solution to update it.
2. In each iteration of the Mountain Climbing Algorithm, we need the products $\mathbf{A}\mathbf{x}_k$ and $\mathbf{A}\mathbf{y}_k$ between the matrix and a vector. We could save the previous products $\mathbf{A}\mathbf{x}_{k-1}$ and $\mathbf{A}\mathbf{y}_{k-1}$ and only recompute the parts of the products that change.

Thank you for your attention