SEMIDEFINITE PROGRAMMING
&
GRAPH BISECTION

Henry Wolkowicz
University of Waterloo

Work with Franz Rendl, Qing Zhao, Stefan Karisch

Semidefinite Programming
looks just like
Linear Programming

$$
\textbf{(P)} \quad
\begin{array}{rll}
p^* = & \sup & c^t x \\
 & \text{s.t.} & Ax \preceq b \quad (b - Ax \in \mathcal{P}) \\
 & & x \in \Re^m
\end{array}
$$

$\preceq$ denotes the Löwner partial order

$A : \Re^m \to \mathcal{S}_n,\ n \times n$ symmetric matrices

$\mathcal{P}$, cone of positive semidefinite matrices

replaces

$\Re^n_+$, the nonnegative orthant

payoff function player X to player Y

$$L(x,U) := \langle U, b\rangle + x^t(c - A^*U)$$

The dual is obtained from the optimal strategy of the competing player

$$p^* = \max_x \min_{U\succeq 0} L(x,U) \leq d^* = \min_{U\succeq 0} \max_x L(x,U)$$

The hidden constraint $c - A^*U = 0$ yields the dual

$$\begin{array}{rcll} & d^* = & \inf & \operatorname{trace} bU \\ (\mathbf{D}) & & \text{s.t.} & A^*U = c \\ & & & U \succeq 0. \end{array}$$

for the primal

$$\begin{array}{rcll} & p^* = & \sup & c^t x \\ (\mathbf{P}) & & \text{s.t.} & Ax \preceq b \\ & & & x \in \Re^m \end{array}$$

Characterization of optimality for the dual pair x, U

$$Ax \preceq b \qquad \text{primal feasibility}$$

$$A^*U = c \qquad \text{dual feasibility}$$

$$U \circ (Ax - b) = 0e \qquad \text{complementary slackness}$$

$$U \circ (Ax - b) = \mu e \qquad \text{perturbed}$$

Forms the basis for:

primal simplex method
dual simplex method
interior point methods

Why use SEMIDEFINITE PROGRAMMING?

Quadratic approximations are better than linear approximations.

Quadratic approximations are too hard to solve!

But, we can solve relaxations of quadratic approximations efficiently using semidefinite programming.

APPLICATIONS

1. Finding bounds and good feasible solutions for hard combinatorial problems such as:
   (a) max-cut;
   (b) graph partitioning;
   (c) quadratic assignment problem;
   (d) max-clique.
2. Unconstrained and constrained optimization techniques, e.g.
   (a) quasi-Newton updates that preserve positive definiteness.
   (b) Trust region algorithms for large scale minimization.

(c) Extended SQP techniques for constrained minimization.

3. Partial Hermitian matrix completion problems.

4. Min-max eigenvalue problems, matrix norm minimization, eigenvalue localization.

How does SDP arise from quadratic approximations?

Let

$$q_i(y) = \frac{1}{2} y^t Q_i y + y^t b_i + c_i, \;\; y \in \Re^n$$

$$(\mathbf{QQP}) \quad \begin{array}{rcc} q^* = & \min & q_0(y) \\ & \text{s.t.} & q_i(y) = 0 \\ & & i = 1, \ldots m \end{array}$$

Lagrangian:

$$\begin{array}{rcl} L(y,x) & = & \frac{1}{2} y^t (Q_0 - \sum_{i=1}^m x_i Q_i) y \\ & & +y^t (b_0 - \sum_{i=1}^m x_i b_i) \\ & & +(c_0 - \sum_{i=1}^m x_i c_i) \end{array}$$

$$q^* = \min_y \max_x L(y,x) \geq d^* = \max_x \min_y L(y,x).$$

homogenize

$$y_0 y^t (b_0 - \sum_{i=1}^m x_i b_i), \quad y_0^2 = 1.$$

$$
\begin{aligned}
d^* &= \\
&= \max_x \min_y \quad L(y,x) \\
&= \max_x \min_{y_0^2=1} \quad \tfrac{1}{2} y^t (Q_0 - \textstyle\sum_{i=1}^m x_i Q_i) y \quad (+t y_0^2) \\
&\qquad + y_0 y^t (b_0 - \textstyle\sum_{i=1}^m x_i b_i) \\
&\qquad + (c_0 - \textstyle\sum_{i=1}^m x_i c_i) \quad (-t)
\end{aligned}
$$

The hidden semidefinite constraint yields the semidefinite program, i.e. we get
$A : \Re^{m+1} \to \mathcal{S}_{n+1}$

$$
B = \begin{pmatrix} 0 & b_0^t \\ b_0 & Q_0 \end{pmatrix}, A \begin{pmatrix} t \\ x \end{pmatrix} = \begin{bmatrix} -t & \sum_{i=1}^m x_i b_i^t \\ \sum_{i=1}^m x_i b_i & \sum_{i=1}^m x_i Q_i \end{bmatrix}
$$

$$
B - A \begin{pmatrix} t \\ x \end{pmatrix} \succeq 0.
$$

The dual program is equivalent to the SDP (with $c_0 = 0$)

$$\begin{array}{lrl} & d^* = \ \sup & -t - \sum_{i=1}^m x_i c_i \\ (\mathbf{D}) & \text{s.t.} & A \begin{pmatrix} t \\ x \end{pmatrix} \preceq B \\ & & x \in \Re^m, t \in \Re \end{array}$$

As in linear programming, the dual of the dual is obtained from the optimal strategy of the competing player:

$$\begin{array}{lrl} & d^* = \ \inf & \operatorname{trace} BU \\ (\mathbf{DD}) & \text{s.t.} & A^* U = \begin{pmatrix} -1 \\ -c \end{pmatrix} \\ & & U \succeq 0. \end{array}$$

QUADRATIC ASSIGNMENT PROBLEM QAP

$$\mu^* := \min_{X \in \Pi} \ \text{trace}\, AXBX^t - 2CX^t$$

A, B and C are real $n \times n$ matrices
Π is the set of permutaion matrices.

Rewrite as

$$(QAP_E) \quad \begin{array}{rll} \mu^* := & \min & \text{trace}\, AXBX^t - 2CX^t \\ & \text{s.t.} & XX^t = I, \left(X^tX = I\right) \\ & & \left(Xe = X^te = e\right) \\ & & X_{ij}^2 - X_{ij} = 0, \quad \forall i, j. \end{array}$$

ignore $Xe = X^te = e$ for now

Find the semidefinite relaxation by taking the dual of the Lagrangian dual.

We first add the (0,1)-constraints to the objective function using Lagrange multipliers W_{ij}

$$\mu_{\mathcal{O}} = \min_{XX^t=X^tX=I} \max_{W} \operatorname{trace} AXBX^t - 2CX^t + \sum_{ij} W_{ij}(X_{ij}^2 - X_{ij}).$$

We now homogenize the objective function by multiplying by a constrained scalar x_0

$$\mu_{\mathcal{O}} \geq \mu_R =$$

$$\max_{W} \min_{\substack{XX^t=X^tX=I \\ x_0^2=1}} \operatorname{trace} \Big[AXBX^t + W(X \circ X)^t - x_0(2C + W)X^t \Big].$$

Introducing a Lagrange multiplier w_0 for the constraint on x_0 and Lagrange multipliers S_b for $XX^t = I$ and S_o for $X^tX = I$ we get

$$\begin{array}{ll} \mu_{\mathcal{O}} \geq \mu_R := & \\ \max\limits_{W} \min\limits_{X,\ x_0} & \text{trace}\left[AXBX^t + W(X \circ X)^t + w_0 x_0^2\right. \\ & \left. + S_b XX^t + S_o X^t X\right] \\ & -\text{trace}\, x_0(2C + W)X^t \\ & -w_0 - \text{trace}\, S_b - \text{trace}\, S_o. \end{array}$$

We have grouped the quadratic, linear, and constant terms together. We now define $x := \text{vec}\, X$, $y^t := (x_0, x^t)$ and $w^t := (w_0, \text{vec}\, W^t)$ and get

$$\begin{array}{ll} \mu_R = & \\ \max\limits_{W} \min\limits_{y} & y^t \left[L_Q + Arrow(w) + \mathrm{B}^0\text{Diag}\,(S_b) + \right. \\ & \left. \mathrm{O}^0\text{Diag}\,(S_o)\right] y \\ & -w_0 - \text{trace}\, S_b - \text{trace}\, S_o \end{array}$$

We used the $(n^2+1) \times (n^2+1)$ matrix

$$L_Q := \begin{bmatrix} 0 & -\mathrm{vec}\,(C)^t \\ -\mathrm{vec}\,(C) & B \otimes A \end{bmatrix},$$

and the (interesting) linear operators

$$\mathrm{Arrow}\,(w) := \begin{bmatrix} w_0 & -\frac{1}{2} w_{1:n^2}^t \\ -\frac{1}{2} w_{1:n^2} & \mathrm{Diag}\,\left(w_{1:n^2}\right) \end{bmatrix},$$

$$\mathrm{B^0Diag}\,(S) := \begin{bmatrix} 0 & 0 \\ 0 & I \otimes S_b \end{bmatrix}$$

and

$$\mathrm{O^0Diag}\,(S) := \begin{bmatrix} 0 & 0 \\ 0 & S_o \otimes I \end{bmatrix}.$$

The hidden semidefinite constraint yields the equivalent SDP:

$$
(D_{\mathcal{O}}) \quad
\begin{array}{ll}
\max & -w_0 - \operatorname{trace} S_b - \operatorname{trace} S_o \\
\text{s.t.} & L_Q + \operatorname{Arrow}(w) + \\
 & \text{B}^0\operatorname{Diag}(S_b) + \text{O}^0\operatorname{Diag}(S_o) \succeq 0,
\end{array}
$$

The dual of this dual yields the semidefinite relaxation.

$Y \succeq 0$ is $(n^2+1) \times (n^2+1)$
the dual matrix variable

$$
(SDP_{\mathcal{O}}) \quad
\begin{array}{llll}
\min & \operatorname{trace} L_Q Y & & \\
\text{s.t.} & \text{b}^0\operatorname{diag}(Y) = I & & \text{o}^0\operatorname{diag}(Y) = I \\
 & \operatorname{arrow}(Y) = e_0 & & Y \succeq 0
\end{array}
$$

adjoint operators are:

$$\text{arrow}\,(Y) := \text{diag}\,(Y) - (0, (Y_{0,1:n^2})^t.$$

$$\text{b}^0\text{diag}\,(Y) := \sum_{k=1}^{n} Y_{(k-1)n+1:kn,(k-1)n+1:kn}$$

$$[\text{o}^0\text{diag}\,(Y)]_{ij} := \text{trace}\, Y_{(i-1)n+1:in,(j-1)n+1:jn}$$

Direct Approach to SDP Relaxation

Let

$X \in \Pi_n$ be a permutation matrix

$x = \text{vec}\,(X),\ c = \text{vec}\,(C)$.

$$\begin{aligned} q(X) &= \text{trace}\, AXBX^t - 2CX^t \\ &= x^t(B \otimes A)x - 2c^t x \\ &= \text{trace}\, xx^t(B \otimes A) - 2c^t x \\ &= \text{trace}\, L_Q Y_X, \end{aligned}$$

where L_Q is as above and

$$Y_X := \begin{bmatrix} 1 & x^t \\ x & xx^t \end{bmatrix}.$$

Adding Generic Inequality Constraints

From the relaxation for the (0,1)-constraints of the original problem:

$$y_{ij} \geq 0 \quad \text{since} \quad x_i x_j \geq 0.$$

We also get so called triangle inequalities

$$y_{ij} + y_{ik} + y_{jk} - \left(y_{ii} + y_{jj} + y_{kk}\right) + 1 \geq 0$$

and

$$y_{ij} - y_{ik} - y_{jk} + y_{kk} \geq 0$$

Therefore the following is a strengthened semidefinite relaxation of QAP

$$
(SDP) \quad
\begin{array}{lll}
\min & \operatorname{trace} L_Q Y & \\
\text{s.t.} & \mathrm{b}^0\mathrm{diag}\,(Y) = I & \mathrm{o}^0\mathrm{diag}\,(Y) = I \\
& T_0 Y T_0^t = E & T_0 Y_{\cdot,0} = e \\
& \operatorname{arrow}(Y) = e_0 & \operatorname{trian}(Y) \geq -g \\
& Y \succeq 0 &
\end{array}
$$

BUT

Slater's condition always fails!!!

How do we start a primal-dual **INTERIOR** point method???

THEOREM
Let $x = vec(X)$ and F_S be the feasible set of (SDP). Define the centroid point

$$\hat{Y} := \frac{1}{n!} \sum_{X \in \Pi_n} \begin{bmatrix} 1 & x^t \\ x & xx^t \end{bmatrix}.$$

Then:

1. $\hat{Y}$ has a 1 in the (1,1) position and n diagonal $n \times n$ blocks with diagonal elements $1/n$. The first row and column equal the diagonal. The rest of the matrix is made up of $n \times n$ blocks with all elements equal to $1/(n(n-1))$ except for the diagonal elements which are 0:

$$\hat{Y} =$$

$$= \left[\begin{array}{c|cccc} 1 & & \frac{1}{n}e^t & & \\ \hline & \operatorname{diag}(\frac{1}{n}e) & (\frac{1}{n(n-1)})(E-I) & \cdots & (\frac{1}{n(n-1)})(E \ldots \\ & \cdots & \cdots & \cdots & \cdots \\ \frac{1}{n}e & \cdots & \cdots & \cdots & \cdots \\ & \cdots & \cdots & \cdots & \cdots \\ & (\frac{1}{n(n-1)})(E-I) & \cdots & (\frac{1}{n(n-1)})(E-I) & \operatorname{diag}(\frac{1}{n} \ldots \end{array}\right.$$

$$= \left[\begin{array}{c|c} 1 & \frac{1}{n}e^t \\ \hline \frac{1}{n}e & E \otimes \left(\frac{1}{n(n-1)}(E-I)\right) - I \otimes \left(\frac{1}{n(n-1)}E - \frac{1}{n-1}I\right) \end{array}\right]$$

2.

$$\operatorname{rank}(\hat{Y}) = (n-1)^2 + 1$$

3.

$$\operatorname{diag} \hat{Y} = \begin{pmatrix} 1 \\ \frac{1}{n}e \end{pmatrix}$$

4.

$$\operatorname{trace}(\hat{Y}) = 1 + n$$

5.
The $n^2 + 1$ eigenvalues of $\hat{Y}$ are given in the vector

$$(2, \frac{1}{n-1} e_{(n-1)^2}, 0 e_{2n-1})$$

6.

$$\mathcal{N}(\hat{Y}) = \left\{ \begin{pmatrix} -\frac{1}{n} e^t u \\ u \end{pmatrix} : u \in \mathcal{R}(T^t) \right\},$$

where T is the assignment constraint matrix.

———————————————————-

We can use the matrix T to project onto the minimal face.

We get a simplified SDP with a positive definite feasible point.

Graph Equi-Partitioning

(special case of QAP)

k and m given integers
G an edgeweighted undirected graph on $n := km$ nodes, given by its adjacency matrix A.
(a_{ij} weight of edge $i \leftrightarrow j$)

a k-partition of $V(G)$ is a partitioning of $V(G)$ into k subsets $(S_1, \ldots, S_k)$ of equal cardinality.

denoted (k-GP)

the columns of $Y \in \Re^{n \times k}$ are the characteristic vectors for the sets S_k

$$\mathcal{F}_k := \{Y : Yu_k = u_n,\ Y^t u_n = m u_k, y_{ij} \in \{0, 1\}\}$$

$L := \operatorname{diag}(Au_n) - A$ *Laplacian matrix associated to* G.

weight of the edges of G, cut by some k-partition $Y \in \mathcal{F}_k$,

$$\frac{1}{2} \operatorname{trace} Y^t L Y$$

THE PROBLEM:

$$(k\!-\!GP) \quad \begin{array}{cc} \min & \frac{1}{2}\operatorname{trace} Y^t L Y \quad (= \operatorname{trace} L Y Y^t) \\ \text{subject to} & Y \in \mathcal{F}_k \end{array}$$

$$\mathcal{T}_k := \{X : X = YY^t \text{ for some } Y \in \mathcal{F}_k\}$$

$$\mathcal{E}_m := \{X : X = X^t, \operatorname{diag}(X) = u_n, Xu_n = mu_n\}$$

note $X \in \mathcal{T}_k$ implies that the only eigenvalues of X are 0 and m
Therefore $mI - X \succeq 0$

THE SDP RELAXATION:

$$\min\{\frac{1}{2}\operatorname{trace} LX : X \in \mathcal{E}_m,\ X \succeq 0,\ mI - X \succeq 0\}.$$

We can add further constraints, e.g. $X \geq 0$ and other 'polyhedral constraints'

Numerical tests by:
Stefan Karisch and Franz Rendl

n	$\|E\|$	$\|E_{cut}\|$	$(k-GP_{R1})$	$(k-GP_{R4})$
36a	297	117	111.76	116.22
36b	303	118	111.63	117.18
36c	316	124	119.48	123.34
60a	885	367	354.46	364.84
60b	863	357	343.24	354.57
60c	841	343	329.43	341.61
84a	1742	742	716.38	734.70
84b	1793	772	744.67	762.38
84c	1753	753	728.01	744.25
108a	2873	1247	1205.80	1231.12
108b	2933	1283	1243.12	1264.06
108c	2871	1240	1201.78	1227.50
132a	4294	1885	1832.76	1861.55
132b	4301	1883	1842.16	1870.04
132c	4257	1854	1808.86	1837.78

Partitioning unweighted random graphs into $k=2$ components.

n	$\|E_{cut}\|$	$(k-GP_{R1})$	$(k-GP_{R2})$		$(k-GP_{R3})$
36a	160	149.02	154.32	(102)	157.36
36b	163	148.84	155.84	(98)	159.11
36c	173	159.31	166.22	(103)	169.40
60a	502	472.61	484.14	(174)	489.88
60b	485	457.65	468.21	(185)	474.03
60c	470	439.24	451.49	(200)	457.79
84a	1011	955.18	972.89	(300)	981.17
84b	1044	992.89	1007.73	(290)	1016.83
84c	1024	970.68	986.38	(276)	993.57
108a	1686	1607.74	1630.80	(399)	1642.86
108b	1738	1657.50	1675.22	(376)	1685.67
108c	1688	1602.38	1628.73	(404)	1639.40
132a	2556	2443.69	2469.33	(498)	2483.07
132b	2569	2456.21	2486.40	(510)	2498.01
132c	2526	2411.82	2442.92	(527)	2456.18

Partitioning unweighted random graphs into $k=3$ components.

n	$\|E_{cut}\|$	$(k-GP_{R1})$	$(k-GP_{R2})$	$(k-GP_{R3})$
36a	186	167.65	179.73	180.74
36b	189	167.45	182.63	183.65
36c	201	179.22	193.99	195.27
60a	577	531.69	556.13	557.92
60b	558	514.86	538.32	540.09
60c	540	494.14	521.45	523.07
84a	1155	1074.57	1112.74	1114.69
84b	1195	1117.00	1152.30	1154.58
84c	1164	1092.01	1126.55	1128.25
108a	1918	1808.71	1860.94	1863.37
108b	1981	1864.68	1907.70	1909.83
108c	1923	1802.68	1858.23	1860.17
132a	2910	2749.15	2810.40	2812.43
132b	2921	2763.24	2827.65	2829.61
132c	2874	2713.30	2781.83	2784.19

Partitioning unweighted random graphs into $k = 4$ components.

n	$\|E_{cut}\|$	$(k-GP_{R1})$	$(k-GP_{R2})$
36a	216	186.27	210.22
36b	221	186.06	215.53
36c	234	199.13	227.62
60a	660	590.77	638.21
60b	640	572.06	618.24
60c	622	549.05	600.66
84a	1313	1193.97	1265.56
84b	1355	1241.11	1310.61
84c	1325	1213.35	1279.39
108a	2184	2009.68	2106.69
108b	2237	2071.87	2158.07
108c	2185	2002.97	2104.26
132a	3290	3054.61	3171.61
132b	3292	3070.26	3187.10
132c	3251	3014.78	3141.52

Partitioning unweighted random graphs into $k = 6$ components.

n	$w(E)$	$w(E_{cut})$	$(k-GP_{R1})$	$(k-GP_{R4})$
36d	3055	1426	1402.76	1425.27
36e	3168	1482	1462.12	1481.12
36f	3134	1454	1435.05	1453.24
60d	8831	4151	4118.59	4150.11
60e	8833	4154	4102.88	4149.89
60f	8777	4132	4078.85	4125.43
84d	17241	8152	8057.77	8132.57
84e	17547	8327	8233.21	8296.74
84f	17473	8264	8166.24	8237.93
108d	29153	13895	13728.41	13825.87
108e	28877	13699	13550.40	13649.46
108f	28892	13709	13588.80	13678.63
132d	43208	20581	20380.59	20514.25
132e	43220	20618	20415.68	20536.67
132f	43391	20707	20488.58	20617.09

Partitioning weighted random graphs into $k = 2$ components.

n	$w(E_{cut})$	$(k-GP_{R1})$	$(k-GP_{R2})$		$(k-GP_{R3})$
36d	1924	1870.35	1891.39	(96)	1904.92
36e	2006	1949.49	1973.55	(102)	1987.21
36f	1974	1913.40	1944.87	(111)	1956.33
60d	5609	5491.45	5533.67	(182)	5558.27
60e	5592	5470.51	5520.86	(195)	5543.31
60f	5555	5438.46	5481.36	(177)	5507.30
84d	10960	10743.70	10828.31	(286)	10860.45
84e	11181	10977.62	11034.14	(322)	11068.14
84f	11105	10888.33	10966.49	(315)	11001.26
108d	18624	18304.55	18398.93	(433)	18438.50
108e	18420	18067.20	18180.35	(429)	18223.56
108f	18442	18118.40	18219.78	(413)	18258.13
132d	27659	27174.12	27325.69	(564)	27374.23
132e	27657	27220.90	27332.27	(506)	27392.74
132f	27768	27318.11	27444.35	(543)	27500.82

Partitioning weighted random graphs into $k = 3$ components.

n	$w(E_{cut})$	$(k-GP_{R1})$	$(k-GP_{R2})$	$(k-GP_{R3})$
36d	2182	2104.14	2152.81	2159.53
36e	2268	2193.18	2243.20	2250.25
36f	2241	2152.57	2212.78	2218.76
60d	6341	6177.88	6288.66	6278.02
60e	6344	6154.33	6259.22	6266.31
60f	6282	6118.27	6214.29	6222.40
84d	12421	12086.66	12260.11	12268.31
84e	12645	12349.82	12482.87	12492.12
84f	12553	12249.37	12411.77	12421.67
108d	21069	20592.62	20796.31	20804.05
108e	20836	20325.60	20562.29	20571.90
108f	20859	20383.20	20599.70	20606.26
132d	31277	30570.89	30888.41	30894.53
132e	31273	30623.52	30885.18	30894.99
132f	31392	30732.87	31016.60	31025.67

Partitioning weighted random graphs into $k=4$ components.

n	$w(E_{cut})$	$(k-GP_{R1})$	$(k-GP_{R2})$
36d	2454	2337.93	2434.02
36e	2552	2436.87	2533.87
36f	2525	2391.75	2502.79
60d	7116	6864.32	7039.53
60e	7115	6838.14	7033.36
60f	7048	6798.08	6985.54
84d	13923	13429.63	13740.85
84e	14171	13722.03	13983.97
84f	14093	13610.41	13909.54
108d	23575	22880.69	23258.89
108e	23303	22584.00	23007.83
108f	23345	22648.00	23043.71
132d	34985	33967.66	34518.13
132e	34965	34026.13	34522.69
132f	35109	34147.64	34663.42

Partitioning weighted random graphs into $k=6$ components.

k	gap_{R1}	gap_{R2}	gap_{R3}	gap_{R4}
2	3.34	–	–	0.73
3	5.54	3.50	2.46	–
4	7.34	3.41	3.17	–
6	9.88	3.29	–	–

Average gaps in percent for unweighted graphs.

k	gap_{R1}	gap_{R2}	gap_{R3}	gap_{R4}
2	1.14	–	–	0.23
3	2.07	1.31	0.95	–
4	2.66	1.22	1.12	–
6	3.54	1.15	–	–

Average gaps in percent for weighted graphs.

n	$\lvert E\rvert$	$\lvert E_{cut}\rvert$	$(k-GP_{R1})$	$(k-GP_{R4})$
124a	149	13 (13)	7.34	12.00
124b	318	63 (63)	46.86	61.01
124c	620	178 (179)	153.01	170.91
124d	1271	449 (449)	418.98	440.06
250a	331	29 (29)	15.44	24.87
250b	612	114 (116)	81.86	100.45
250c	1283	357 (361)	303.53	327.88
250d	2421	828 (833)	747.32	779.55

Partitioning graphs of Johnson et al. into $k = 2$ components.

n	k	$\lvert E_{cut}\rvert$	$(k-GP_{R1})$	$(k-GP_{R2})$	$(k-GP_{R3})$
124a	4	23	11.01	13.65 (638)	17.16
124b	4	107	70.29	81.36 (604)	92.06
124c	4	294	229.52	251.51 (704)	260.65
124d	4	726	628.48	662.64 (749)	669.14
250a	5	63	24.71	31.08	31.08
250b	5	218	130.99	150.32	152.82
250c	5	648	485.66	527.12	530.82
250d	5	1447	1195.72	1268.17	1269.03

Partitioning graphs of Johnson et al. into $k = 4$ or $k = 5$ components.

n	k	$\lvert E_{cut}\rvert$	$(k-GP_{R1})$	$(k-GP_{R2})$	$(k-GP_{R3})$	$(k-GP_{R4})$
120	2	8	1.67	–	–	4.83
120	3	10	2.22	4.64	7.67	–
120	4	22	2.50	9.07	12.07	–
120	5	20	2.67	14.26	16.50	–
120	6	36	2.78	22.38	25.90	–

Partitioning a geometric graph of size $n = 120$ with $\lvert E\rvert = 413$.

n	$\|E\|$	cut	$\mathcal{E}_m \cap \mathcal{P}$
36	305	119	112
60	903	367	355
84	1762	747	717
108	2897	1252	1206
132	4325	1901	1833

Partitioning random graphs into 2 components of equal size. The first two columns describe the size of the graphs, column 3 contains the best equipartition found, column 4 contains the lower bound.

n	cut	$\mathcal{E}_m \cap \mathcal{P}$	$\mathcal{E}_m \cap \mathcal{P} \cap \mathcal{N}$	sign constr.
36	160	149.1	154.3	120
60	506	472.6	484.1	223
84	1014	955.1	972.8	349
108	1693	1607.7	1630.8	469
132	2563	2443.7	2469.3	554

Partitioning random graphs into $k = 3$ components of equal size. The first column identifies the graph, column 2 contains the best 3-partition found, columns 3 and 4 contain lower bounds from semidefinite relaxations, the last column indicates the number of nonnegativity constraints $x_{ij} \geq 0$ used, to insure $X \geq 0$.

n	cut	$\mathcal{E}_m \cap \mathcal{P}$	$\mathcal{E}_m \cap \mathcal{P} \cap \mathcal{N}$	sign constr.
36	186	167.6	179.7	211
60	585	531.7	556.1	405
84	1165	1074.6	1112.7	596
108	1935	1808.7	1860.9	859
132	2915	2749.2	2810.4	993

Partitioning random graphs into $k = 4$ components of equal size. The first column identifies the graph, column 2 contains the best 4-partition found, columns 3 and 4 contain lower bounds from semidefinite relaxations, the last column indicates the number of non-negativity constraints $x_{ij} \geq 0$ used, to insure $X \geq 0$.