# BiqCrunch: a semidefinite-based solver for binary quadratic problems

Nathan Krislock

PIMS Postdoctoral Fellow, Computer Science
University of British Columbia, Canada

Canadian Discrete and Algorithmic Mathematics Conference
Memorial University of Newfoundland
June 10–13, 2013

Joint work with Jérôme Malick (CNRS)
and Frédéric Roupin (Université Paris 13)

# BiqCrunch

## Binary Quadratic Problems

- BiqCrunch is a branch & bound solver for binary quadratic problems:

$$
\begin{array}{ll}
\text{maximize} & z^T A_0 z + b_0^T z + c_0 \\
\text{subject to} & z^T A_i z + b_i^T z = c_i, \quad i \in \{1, \ldots, m_E\} \\
& z^T A_j z + b_j^T z \leq c_j, \quad j \in \{1, \ldots, m_I\} \\
& z \in \{0, 1\}^n
\end{array}
$$

- examples of binary quadratic problems:
  - Max-Cut / Unconstrained Binary Quadratic
  - Maximum $k$-Cluster
  - Maximum Independent Set / Quadratic Stable Set
  - Quadratic Knapsack
  - Quadratic Assignment
  - ...

# BiqCrunch

## BiqCrunch

- uses our improved semidefinite bounding procedure
- written in C and Fortran and uses:
  - L-BFGS-B, a library for quasi-Newton bound-constrained optimization
  - BOB, a branch-and-bound framework
- uses our `BC` (BiqCrunch) format – is very similar to the `SDPA` format
- available as an online solver

# BiqCrunch

## Vector form

$$
\begin{array}{ll}
\text{maximize} & z^T A_0 z + b_0^T z + c_0 \\
\text{subject to} & z^T A_i z + b_i^T z = c_i, \quad i \in \{1, \ldots, m_E\} \\
& z^T A_j z + b_j^T z \leq c_j, \quad j \in \{1, \ldots, m_I\} \\
& z \in \{0,1\}^n
\end{array}
$$

## Matrix form

$$
\begin{array}{ll}
\text{maximize} & \langle Q_0, Z \rangle \\
\text{subject to} & \langle Q_i, Z \rangle = c_i, \quad i \in \{1, \ldots, m_E\} \\
& \langle Q_j, Z \rangle \leq c_j, \quad j \in \{1, \ldots, m_I\} \\
& Z = \begin{bmatrix} zz^T & z \\ z^T & 1 \end{bmatrix}, \quad z \in \{0,1\}^n
\end{array}
$$

- $Q_0 = \begin{bmatrix} A_0 & \frac{1}{2} b_0 \\ \frac{1}{2} b_0^T & c_0 \end{bmatrix}$, $Q_i = \begin{bmatrix} A_i & \frac{1}{2} b_i \\ \frac{1}{2} b_i^T & 0 \end{bmatrix}$, $Q_j = \begin{bmatrix} A_j & \frac{1}{2} b_j \\ \frac{1}{2} b_j^T & 0 \end{bmatrix}$

# BiqCrunch

*A semidefinite branch-and-bound method for solving binary quadratic problems*



## BiqCrunch

**BiqCrunch** is a semidefinite-based solver for binary quadratic problems. It uses a branch-and-bound method featuring an improved semidefinite bounding procedure [3], mixed with a polyhedral approach (see [1,2] for details). BiqCrunch uses a particular *BC* input file format that is similar to the *SDPA* format to describe the combinatorial problems. Documentation is available here.

You can use BiqCrunch online to solve any 0-1 quadratic problem to optimality, or simply to get an SDP-quality bound [3]. Specific versions are provided for solving *Max-Cut*, *k-cluster* and *Max-independent set* problems. We also provide conversion tools for these problems and a large set of BC input files.

## Papers


**[1] (2012) N. Krislock, J. Malick, F. Roupin.** Improved semidefinite branch-and-bound algorithm for k-cluster. Submitted to JOCO.

**[2] (2012) N. Krislock, J. Malick, F. Roupin:** Improved semidefinite bounding procedure for solving Max-Cut problems to optimality. Mathematical Programming A

**[3] (2011) J. Malick, F. Roupin:** On the bridge between combinatorial optimization and nonlinear optimization: a family of semidefinite bounds for 0-1 quadratic problems leading to quasi-Newton methods. To appear in Mathematical Programming B.


## Acknowledgments


**BiqCrunch** is the outcome of several research works and collaborations. This work was supported by the LIPN lab, CNRS (through "GdR Recherche Opérationnelle"), Grenoble University (Université Joseph Fourier, through "Pôle Math-STIC"). We thank: Quentin Monnet and Lise-Marie Veillon (students of ENSIIE, Evry, France in 2008) for their help in developing and testing parts of a preliminary version of the solver, and Geoffrey Kozak (student of Sup Galilée, Villetaneuse, France in 2012) for his help in developing several tools and for testing the solver.


This website was created by Marco Casazza. Please feel free to contact F. Roupin with any questions or suggestions.

### CONTACTS

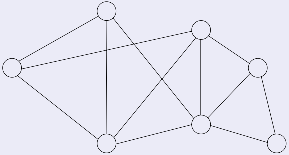

**Nathan Krislock:**
nathan.krislock@inria.fr

**Jérôme Malick:**
jerome.malick@inria.fr

**Frédéric Roupin:**
Frederic.Roupin@lipn.univ-paris13.fr

# Max-Cut

## Max-Cut



$$\begin{array}{ll} \text{maximize} & \sum_{ij \in E} w_{ij} \left( \frac{1 - x_i x_j}{2} \right) \\ \text{subject to} & x \in \{-1, 1\}^n \end{array}$$

$G = (V, E)$ $\quad$ $(n = |V|)$

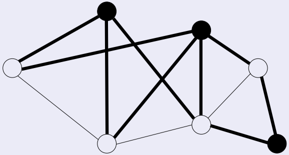
## Equivalent formulation

$$\text{(MC)} \quad \begin{array}{ll} \text{maximize} & x^T Q x \\ \text{subject to} & x \in \{-1, 1\}^n \end{array}$$

- $Q = \frac{1}{4} L \in \mathcal{S}^n$ $\quad$ ($L$ is the Laplacian matrix of $G$)

# Max-Cut

## Max-Cut



$G = (V, E)$

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{ij \in E} w_{ij} \left( \frac{1 - x_i x_j}{2} \right) \\ \text{subject to} & x \in \{-1, 1\}^n \end{array}$$

$(n = |V|)$

## Equivalent formulation

$$\text{(MC)} \quad \begin{array}{ll} \text{maximize} & x^T Q x \\ \text{subject to} & x \in \{-1, 1\}^n \end{array}$$

- $Q = \frac{1}{4} L \in \mathcal{S}^n$ ($L$ is the Laplacian matrix of $G$)

# Max-Cut

## Max-Cut

$$(\text{MC}) \quad \begin{array}{|ll|} \text{maximize} & x^T Q x \\ \text{subject to} & x \in \{-1,1\}^n \end{array}$$

$X = xx^T,\ x \in \{-1,1\}^n \iff \operatorname{diag}(X) = e,\ X \succeq 0,\ \operatorname{rank}(X) = 1$

$$(\text{MC}) \quad \begin{array}{|ll|} \text{maximize} & \langle Q, X \rangle \\ \text{subject to} & \operatorname{diag}(X) = e,\ X \succeq 0 \\ & \operatorname{rank}(X) = 1 \end{array}$$

- $\langle Q, X \rangle = \sum_{ij} Q_{ij} X_{ij} = \operatorname{trace}(QX)$
- $\operatorname{diag}(X) = (X_{11}, \ldots, X_{nn})^T,\ e = (1, \ldots, 1)^T$
- $X \succeq 0$ : $X$ is symmetric and positive semidefinite

# Semidefinite Relaxation of Max-Cut

## Semidefinite relaxation

$$\text{(SDP)} \quad \boxed{\begin{array}{ll} \text{maximize} & \langle Q, X \rangle \\ \text{subject to} & \operatorname{diag}(X) = e,\ X \succeq 0 \end{array}}$$

- gives an upper bound on Max-Cut: (MC) $\leq$ (SDP)
- tight bounds needed for B&B methods to solve Max-Cut efficiently
- Goemans and Williamson (1995): (SDP) $<$ 1.14(MC)
- can be solved very efficiently by a primal-dual interior-point method
- bound is too weak to be used to solve Max-Cut efficiently, so we add some triangle inequalities

# Semidefinite Relaxation of Max-Cut

## Enhancing the SDP bound

There are $4\binom{n}{3}$ triangle inequalities (for $1 \leq i < j < k \leq n$):

$$\begin{aligned} X_{ij} + X_{ik} + X_{jk} \geq -1, &\quad -X_{ij} + X_{ik} - X_{jk} \geq -1 \\ X_{ij} - X_{ik} - X_{jk} \geq -1, &\quad -X_{ij} - X_{ik} + X_{jk} \geq -1 \end{aligned}$$

Choosing a subset $\mathcal{I}$ of the inequalities, we have

$$(\text{SDP}_{\mathcal{I}}) \quad \boxed{\begin{array}{ll} \text{maximize} & \langle Q, X \rangle \\ \text{subject to} & \operatorname{diag}(X) = e,\ X \succeq 0 \\ & \mathcal{A}_{\mathcal{I}}(X) \geq -e \end{array}}$$

- can greatly enhance the bound:

$$(\text{MC}) \leq (\text{SDP}_{\mathcal{I}}) \leq (\text{SDP})$$

# Our Semidefinite Bounds

## A simple fact

$$\operatorname{diag}(X) = e,\ X \succeq 0 \quad \Longrightarrow \quad \|X\|_F^2 \leq n^2$$

## Our semidefinite bounds

$$(\mathrm{SDP}_{\mathcal{I}}^{\alpha}) \quad \begin{array}{ll} \text{maximize} & \langle Q, X\rangle + \frac{\alpha}{2}\left(n^2 - \|X\|_F^2\right) \\ \text{subject to} & \operatorname{diag}(X) = e,\ X \succeq 0 \\ & \mathcal{A}_{\mathcal{I}}(X) \geq -e \end{array}$$

- for $\alpha \geq 0$: $(\mathrm{SDP}_{\mathcal{I}}) \leq (\mathrm{SDP}_{\mathcal{I}}^{\alpha})$
- smaller $\alpha$ gives tighter upper bounds:

$$\alpha < \alpha' \quad \Longrightarrow \quad (\mathrm{SDP}_{\mathcal{I}}^{\alpha}) < (\mathrm{SDP}_{\mathcal{I}}^{\alpha'})$$

- $\lim_{\alpha \to 0}(\mathrm{SDP}_{\mathcal{I}}^{\alpha}) = (\mathrm{SDP}_{\mathcal{I}})$

# Our Semidefinite Bounds

## Dual function $F_{\mathcal{I}}^{\alpha}(y, z)$

- The Lagrangian is ($y \in \mathbb{R}^n$, $z \in \mathbb{R}_+^{|\mathcal{I}|}$):

$$\mathcal{L}(X; y, z) := \langle Q, X\rangle + \frac{\alpha}{2}\left(n^2 - \|X\|_F^2\right) + \langle y, e - \operatorname{diag}(X)\rangle + \langle z, e + \mathcal{A}_{\mathcal{I}}(X)\rangle$$

- The dual function is:

$$F_{\mathcal{I}}^{\alpha}(y, z) := \max_{X \succeq 0} \mathcal{L}(X; y, z) = \frac{1}{2\alpha}\left\| X_{\mathcal{I}}(y, z) \right\|_F^2 + e^T y + e^T z + \frac{\alpha}{2} n^2$$

where $$\boxed{X_{\mathcal{I}}(y, z) := \left[Q - \operatorname{Diag}(y) + \mathcal{A}_{\mathcal{I}}^*(z)\right]_+}$$

# Our Semidefinite Bounds

## Dual function $F_{\mathcal{I}}^{\alpha}(y, z)$

- The Lagrangian is ($y \in \mathbb{R}^n$, $z \in \mathbb{R}_+^{|\mathcal{I}|}$):

$$\mathcal{L}(X; y, z) := \langle Q - \mathrm{Diag}(y) + \mathcal{A}_{\mathcal{I}}^*(z), X \rangle - \frac{\alpha}{2} \|X\|_F^2 + e^T y + e^T z + \frac{\alpha}{2} n^2$$

- The dual function is:

$$\begin{aligned} F_{\mathcal{I}}^{\alpha}(y, z) &:= \max_{X \succeq 0} \mathcal{L}(X; y, z) \\ &= \frac{1}{2\alpha} \left\| X_{\mathcal{I}}(y, z) \right\|_F^2 + e^T y + e^T z + \frac{\alpha}{2} n^2 \end{aligned}$$

where $$\boxed{X_{\mathcal{I}}(y, z) := \left[ Q - \mathrm{Diag}(y) + \mathcal{A}_{\mathcal{I}}^*(z) \right]_+}$$

# Our Semidefinite Bounds

## Generic semidefinite relaxation

$$(\text{SDP}) \quad \boxed{\begin{array}{ll} \text{maximize} & \langle Q, X \rangle \\ \text{subject to} & \mathcal{A}X \leq a \\ & \mathcal{B}X = b \\ & X \succeq 0 \end{array}}$$

- $\operatorname{diag}(X) = e$ constraints included in the equality constraints $\mathcal{B}X = b$

## Generic semidefinite bounds

$$F^\alpha(y,z) = \frac{1}{2\alpha}\left\| X(y,z) \right\|_F^2 + b^T y + a^T z + \frac{\alpha}{2} n^2$$

$$\boxed{X(y,z) = \left[ Q - \mathcal{B}^*(y) + \mathcal{A}^*(z) \right]_+}$$

# Our Semidefinite Bounds

## Computing the best bound $F_{\mathcal{I}}^{\alpha}(y,z)$

$$(\text{DSDP}_{\mathcal{I}}^{\alpha}) \quad \boxed{\begin{array}{cc} \text{minimize} & F_{\mathcal{I}}^{\alpha}(y,z) \\ \text{subject to} & z \geq 0 \end{array}}$$

## $(\text{DSDP}_{\mathcal{I}}^{\alpha})$ is a smooth convex optimization problem

$$\nabla_y F_{\mathcal{I}}^{\alpha}(y,z) = e - \operatorname{diag}\left(\frac{1}{\alpha} X_{\mathcal{I}}(y,z)\right)$$

$$\nabla_z F_{\mathcal{I}}^{\alpha}(y,z) = e + \mathcal{A}_{\mathcal{I}}\left(\frac{1}{\alpha} X_{\mathcal{I}}(y,z)\right)$$

- can be solved with a quasi-Newton method like L-BFGS-B
- ill-conditioned for very small $\alpha$

# Efficiently Computing the Bounds

## Algorithm: Improved semidefinite bounding procedure

Given:

- Initial vectors and inequalities: $y^0 = 0 \in \mathbb{R}^n$, $\mathcal{I}^0 = \emptyset$, $z^0 = 0 \in \mathbb{R}^0$
- Scalars: $\alpha^1, \varepsilon^1 > 0$
- Parameters: $0 < \theta \leq 1$ and $0 < \rho \leq 1$

For $k = 1, 2, \ldots$ do:

1. Starting from $(y^{k-1}, z^{k-1})$, compute $(y^k, \hat{z}^k)$ such that

$$\max\left\{\left\|\nabla_y F_{\mathcal{I}^{k-1}}^{\alpha^k}(y^k, \hat{z}^k)\right\|_\infty, \left\|\left[\nabla_z F_{\mathcal{I}^{k-1}}^{\alpha^k}(y^k, \hat{z}^k)\right]_-\right\|_\infty\right\} < \varepsilon^k$$

2. Update $\mathcal{I}$, $\alpha$, $\varepsilon$:
   - Add / remove triangle inequalities to get $\mathcal{I}^k$ and $z^k$
   - Reduce $\alpha$: $\alpha^{k+1} \leftarrow \theta\alpha^k$
   - Reduce $\varepsilon$: $\varepsilon^{k+1} \leftarrow \rho\varepsilon^k$

# Efficiently Computing the Bounds

## Convergence Theorem

If:

- $(X^k, y^k, z^k, \mathcal{I}^k)$ generated by Algorithm, where $X^k := \frac{1}{\alpha^k} X_{\mathcal{I}^k}(y^k, z^k)$
- $\alpha^k \to 0$ and $\varepsilon^k \to 0$
- $(\bar{X}, \bar{y}, \bar{z}, \bar{\mathcal{I}})$ is an accumulation point of $(X^k, y^k, z^k, \mathcal{I}^k)$

Then:

- $(\bar{X}, \bar{y}, \bar{z})$ is a primal-dual optimal solution for $(\mathrm{SDP}_{\bar{\mathcal{I}}})$
- the bounds converge to $(\mathrm{SDP}_{\bar{\mathcal{I}}})$:

$$\lim_{k \to \infty} F_{\mathcal{I}^k}^{\alpha^k}(y^k, z^k) = (\mathrm{SDP}_{\bar{\mathcal{I}}})$$

Furthermore, if all violated inequalities are added (in the limit), we have

$$\lim_{k \to \infty} F_{\mathcal{I}^k}^{\alpha^k}(y^k, z^k) = (\mathrm{SDP}_{\mathcal{I}_{\text{all}}})$$

# Comparing to Biq Mac

## Our bounds: $\mathcal{F}_{\mathcal{I}}^{\alpha}(y,z)$

$$F_{\mathcal{I}}^{\alpha}(y,z) = \frac{1}{2\alpha}\left\| \left[ Q - \mathrm{Diag}(y) + \mathcal{A}_{\mathcal{I}}^{*}(z) \right]_{+} \right\|_{F}^{2} + e^{T}y + e^{T}z + \frac{\alpha}{2}n^{2}$$

- $F_{\mathcal{I}}^{\alpha}$ is convex and smooth
- evaluate $F_{\mathcal{I}}^{\alpha}(y,z)$ and $\nabla F_{\mathcal{I}}^{\alpha}(y,z)$ by computing a partial eigenvalue decomposition
- minimize $F_{\mathcal{I}}^{\alpha}(y,z)$ by a quasi-Newton method

## Biq Mac bounds: $\theta_{\mathcal{I}}(z)$

$$\theta_{\mathcal{I}}(z) = e^{T}z \quad + \quad \begin{array}{ll} \text{minimize} & \langle Q + \mathcal{A}_{\mathcal{I}}^{*}(z), X \rangle \\ \text{subject to} & \mathrm{diag}(X) = e,\ X \succeq 0 \end{array}$$

- $\theta_{\mathcal{I}}$ is convex and nonsmooth
- evaluate $\theta_{\mathcal{I}}(z)$ and $\phi \in \partial\theta_{\mathcal{I}}(z)$ by solving an SDP
- minimize $\theta_{\mathcal{I}}(z)$ by a bundle method

# Comparing to Biq Mac

## Our bounds: $\mathcal{F}_{\mathcal{I}}^{\alpha}(y,z)$

$$F_{\mathcal{I}}^{\alpha}(y,z) = \frac{1}{2\alpha}\left\| \left[ Q - \mathrm{Diag}(y) + \mathcal{A}_{\mathcal{I}}^{*}(z) \right]_{+} \right\|_F^2 + e^T y + e^T z + \frac{\alpha}{2} n^2$$

- $F_{\mathcal{I}}^{\alpha}$ is convex and smooth
- evaluate $F_{\mathcal{I}}^{\alpha}(y,z)$ and $\nabla F_{\mathcal{I}}^{\alpha}(y,z)$ by computing a partial eigenvalue decomposition
- minimize $F_{\mathcal{I}}^{\alpha}(y,z)$ by a quasi-Newton method

## Biq Mac bounds: $\theta_{\mathcal{I}}(z)$

$$\theta_{\mathcal{I}}(z) = e^T z \quad + \quad \begin{array}{ll} \text{minimize} & \langle Q + \mathcal{A}_{\mathcal{I}}^{*}(z), X \rangle \\ \text{subject to} & \mathrm{diag}(X) = e,\ X \succeq 0 \end{array}$$

- $\theta_{\mathcal{I}}$ is convex and nonsmooth
- evaluate $\theta_{\mathcal{I}}(z)$ and $\phi \in \partial\theta_{\mathcal{I}}(z)$ by solving an SDP
- minimize $\theta_{\mathcal{I}}(z)$ by a bundle method

# Comparing to Biq Mac

## Our bounds: $\mathcal{F}_{\mathcal{I}}^{\alpha}(y,z)$

$$F_{\mathcal{I}}^{\alpha}(y,z) = \frac{1}{2\alpha}\left\| \left[ Q - \mathrm{Diag}(y) + \mathcal{A}_{\mathcal{I}}^{*}(z) \right]_{+} \right\|_F^2 + e^T y + e^T z + \frac{\alpha}{2} n^2$$

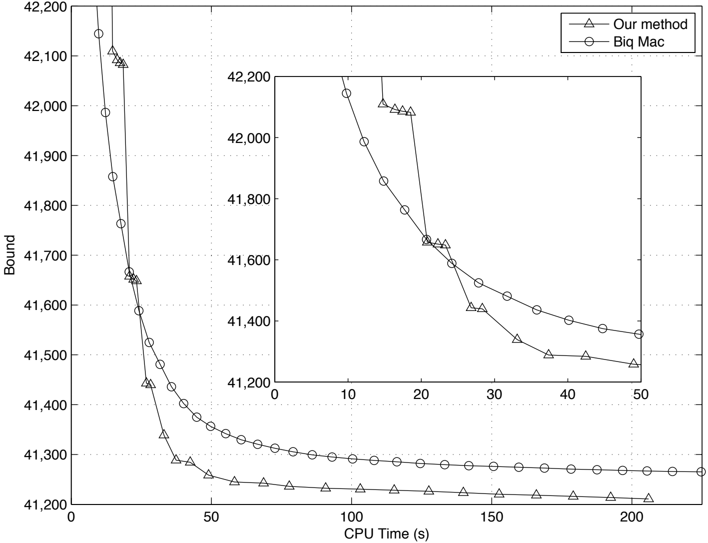
- $F_{\mathcal{I}}^{\alpha}$ is convex and smooth
- evaluate $F_{\mathcal{I}}^{\alpha}(y,z)$ and $\nabla F_{\mathcal{I}}^{\alpha}(y,z)$ by computing a partial eigenvalue decomposition
- minimize $F_{\mathcal{I}}^{\alpha}(y,z)$ by a quasi-Newton method

## Biq Mac bounds: $\theta_{\mathcal{I}}(z)$

$$\theta_{\mathcal{I}}(z) = e^T z \quad + \quad \begin{array}{ll} \text{minimize} & \langle Q + \mathcal{A}_{\mathcal{I}}^{*}(z), X \rangle \\ \text{subject to} & \mathrm{diag}(X) = e,\ X \succeq 0 \end{array}$$

- $\theta_{\mathcal{I}}$ is convex and nonsmooth
- evaluate $\theta_{\mathcal{I}}(z)$ and $\phi \in \partial\theta_{\mathcal{I}}(z)$ by solving an SDP
- minimize $\theta_{\mathcal{I}}(z)$ by a bundle method

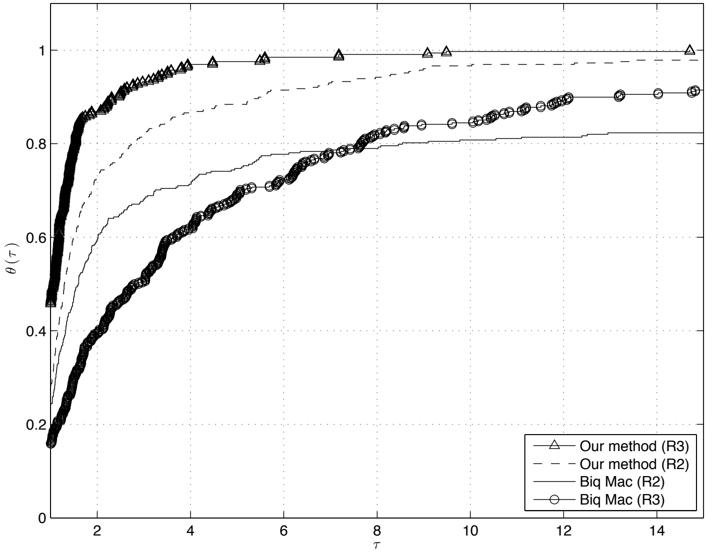
# Comparing to Biq Mac





Nathan Krislock, Jérôme Malick, and Frédéric Roupin. (2012)
Improved semidefinite bounding procedure for solving Max-Cut problems to optimality. *Mathematical Programming*.

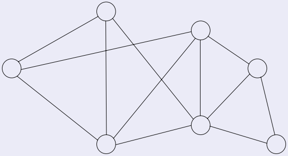
# Comparing to Biq Mac




Nathan Krislock, Jérôme Malick, and Frédéric Roupin. (2012)
Improved semidefinite bounding procedure for solving Max-Cut problems to optimality. *Mathematical Programming*.

# Maximum $k$-cluster

## Maximum $k$-cluster



$$\begin{array}{ll} \text{maximize} & \sum_{ij \in E} w_{ij} z_i z_j \\ \text{subject to} & \sum_{i=1}^n z_i = k \\ & z \in \{0,1\}^n \end{array}$$

$G = (V, E)$ $\qquad$ $(n = |V|)$

## Reinforcing equality constraints

$$\text{(KC)} \quad \begin{array}{ll} \text{maximize} & \frac{1}{2} z^T W z \\ \text{subject to} & \sum_{i=1}^n z_i = k \\ & \sum_{i=1}^n z_i z_j = k z_j, \quad j \in \{1, \ldots, n\} \\ & z \in \{0,1\}^n \end{array}$$


Alain Faye and Frédéric Roupin. (2007)
Partial Lagrangian relaxation for general quadratic programming. *4OR: A Quarterly Journal of Operations Research*.

# Results for maximum $k$-cluster

## Medium problems

| $n$ | $k$ | $d(\%)$ | nodes | time (s) |
|---|---|---|---|---|
| 100 | 25 | 25 | 20.6 | 31 |
| | | 50 | 35.0 | 42 |
| | | 75 | 30.6 | 32 |
| | 50 | 25 | 3.4 | 5 |
| | | 50 | 25.4 | 46 |
| | | 75 | 1.4 | 2 |
| | 75 | 25 | 1.0 | 2 |
| | | 50 | 1.8 | 4 |
| | | 75 | 1.0 | 2 |

| $n$ | $k$ | $d(\%)$ | nodes | time (s) |
|---|---|---|---|---|
| 120 | 30 | 25 | 119.4 | 316 |
| | | 50 | 194.2 | 425 |
| | | 75 | 422.2 | 890 |
| | 60 | 25 | 59.8 | 199 |
| | | 50 | 85.8 | 263 |
| | | 75 | 43.0 | 143 |
| | 90 | 25 | 1.8 | 7 |
| | | 50 | 22.2 | 97 |
| | | 75 | 1.0 | 3 |

- on average 8 times faster than previous best methods
- up to more than 60 times faster than previous best methods

Nathan Krislock, Jerome Malick, and Frédéric Roupin. (2012)
Improved semidefinite branch-and-bound algorithm for $k$-cluster. Technical Report, INRIA Grenoble Rhône-Alpes.

# Results for maximum $k$-cluster

## Large probems

| $n$ | $k$ | $d(\%)$ | nodes | time (s) |
|---|---|---|---|---|
| 140 | 35 | 25 | 366.2 | 1166 |
| | | 50 | 1063.4 | 2889 |
| | | 75 | 1558.6 | 4080 |
| | 70 | 25 | 134.2 | 543 |
| | | 50 | 780.6 | 3035 |
| | | 75 | 52.2 | 203 |
| | 105 | 25 | 2.6 | 14 |
| | | 50 | 11.0 | 62 |
| | | 75 | 6.6 | 35 |

| $n$ | $k$ | $d(\%)$ | nodes | time (s) |
|---|---|---|---|---|
| 160 | 40 | 25 | 744.6 | 2856 |
| | | 50 | 11325.4 | 37565 |
| | | 75 | 8050.6 | 26303 |
| | 80 | 25 | 395.4 | 1835 |
| | | 50 | 993.4 | 4655 |
| | | 75 | 3829.0 | 18654 |
| | 120 | 25 | 31.4 | 220 |
| | | 50 | 17.4 | 143 |
| | | 75 | 9.8 | 82 |

- first time problems of size $n = 140$ and $n = 160$ solved

Nathan Krislock, Jerome Malick, and Frédéric Roupin. (2012)
Improved semidefinite branch-and-bound algorithm for $k$-cluster. Technical Report, INRIA Grenoble Rhône-Alpes.

# Summary & Future Work

## Summary

- BiqCrunch available as an online solver
- able to solve 75% of the test problems faster than the leading Biq Mac method
- able to solve $k$-cluster problems much faster than previously possible
- more numerical results on BiqCrunch website

## Future Work

- make a public release of the BiqCrunch code
- make a version of BiqCrunch to solve Quadratic Assignment Problems
- investigate facial reduction to handle problems with a semidefinite relaxation that is not strictly feasible