Solving Lift-and-Project Relaxations of Binary Integer Programs

Sam Burer
University of Iowa
Dieter Vandenbussche
University of Illinois U-C

Workshop on Integer Programming and Continuous Optimization
Chemnitz University of Technology
November 8, 2004

Outline

- Lift-and-project relaxations
- Lagrangian methods
- Augmented Lagrangian approach
- Computational results
  - Quadratic assignment
  - Erdös-Turán

Lifting (Lovász and Schrijver)

$$\min \; c^T x \quad \text{s.t.} \quad Ax \le b, \; x \in \{0,1\}^n$$

Need tighter/stronger relaxations than LP

Lift using matrix variable: $Y = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix}$

Homogenized version of LP relaxation

$$K := \left\{ \begin{pmatrix} x_0 \\ x \end{pmatrix} \in \Re^{1+n} \; : \; \begin{array}{c} Ax - bx_0 \le 0 \\ 0 \le x \le x_0 e, \; x_0 \le 1 \end{array} \right\}$$

Properties of Y (Lovász and Schrijver)

1. $Y e_i \in K \qquad \forall\, i \geq 0$
2. $Y(e_0 - e_i) \in K \quad \forall\, i \geq 1$
3. $Y e_0 = \text{diag}(Y)$
4. $Y = Y^T$
5. $Y_{00} = 1$
6. $Y \succeq 0$

$$Y = \begin{pmatrix} 1 & x^T \\ x & xx^T \end{pmatrix}$$

Projection (Lovász and Schrijver)

Relaxations of the convex hull of integer solutions:

$$N(K) := \left\{ x \in \Re^n \ : \ \begin{array}{c} (1, x) = \text{diag}(Y) \\ Y \text{ satisfies } 1 - 5 \end{array} \right\}$$

$$N_+(K) := \left\{ x \in \Re^n \ : \ \begin{array}{c} (1, x) = \text{diag}(Y) \\ Y \text{ satisfies } 1 - 6 \end{array} \right\}$$

Linear Optimization

$$\min \; c^T x \quad \text{s.t.} \quad x \in N(K)$$

$$\bar{c} = \begin{pmatrix} 0 \\ c \end{pmatrix}$$

$$\min \; \bar{c}^T Y e_0$$

$$\text{s.t.} \quad Y = Y^T \qquad Y e_0 = \text{diag}(Y)$$

$$Y e_i \in K \qquad \forall \; i \geq 0$$

$$Y(e_0 - e_i) \in K \quad \forall \; i \geq 1$$

$$Y_{00} = 1$$

Very large-scale LP: Can we decompose it somehow?

Semidefinite Optimization

$$\min \; c^T x \quad \text{s.t.} \quad x \in N_+(K)$$

$$\begin{aligned} \min \quad & \bar{c}^T Y e_0 \\ \text{s.t.} \quad & [\text{ same constraints as for } N(K)\ ] \\ & Y \succeq 0 \end{aligned}$$

Very large-scale SDP: Can we decompose it somehow?

Decomposition

Introduce matrix variable $Z \in \Re^{(1+n)\times n}$:

$$Ze_i = Y(e_0 - e_i) \quad \forall\ i \geq 1$$

$$Y = Y^T \qquad Ye_0 = \text{diag}(Y)$$

represent as

$$Ye_i \in K \quad \forall\ i \geq 0$$

$$Ze_i \in K \quad \forall\ i \geq 1$$

$$Y_{00} = 1$$

$$[Y \succeq 0]$$

$$h(Y, Z) = 0$$

Lagrangian relaxation of *N(K)*

$$L(\lambda) := \min \quad \bar{c}^T Y e_0 + \lambda^T h(Y, Z)$$

$$\text{s.t.} \quad \boxed{\begin{array}{ll} Y e_i \in K & \forall\ i \geq 0 \\ Z e_i \in K & \forall\ i \geq 1 \end{array}}$$

$$Y_{00} = 1$$

$L(\lambda)$ requires $2n + 1$

linear optimizations over K

Lagrangian relaxation of $N_+(K)$

Relax $Y \succeq 0$ with dual multiplier $S \succeq 0$:

$$L(\lambda, S) := \min \quad \bar{c}^T Y e_0 + \lambda^T h(Y, Z) - S \bullet Y$$
$$\text{s.t.} \quad [\text{ same constraints as } L(\lambda) \text{ }]$$

$L(\lambda, S)$ requires $2n + 1$
linear optimizations over K

Solving the Lagrangian Dual

$$\max_{\lambda} L(\lambda) \quad \text{or} \quad \max_{\lambda, S \succeq 0} L(\lambda, S)$$

Subgradient Algorithm (semidefinite case)

$\lambda = 0,\ S = 0$

loop

Find $(Y, Z) \in \text{Argmin}\, L(\lambda, S)$

$\lambda \leftarrow \lambda + \alpha\, h(Y, Z)$

$S \leftarrow \text{proj}_{\succeq}(S - \alpha\, Y)$

end loop

Comments on Subgradient Method

- Difficult to choose stepsize in update formulae
- Initial experiments indicated slow convergence
- Can use ergodic recovery scheme to get approximate primal solutions
- In the semidefinite case, must project S onto cone of positive semidefinite matrices in each iteration

Augmented Lagrangian Method

- Combination of Lagrangian and quadratic penalty methods
- Relaxed constraints typically in equality form
- Standard textbook method for nonlinear programs
- Not typically used for linear conic programs

Augmented Lagrangian for *N(K)*

$$A_\sigma(\lambda) := \min \; \bar{c}^T Y e_0 + \lambda^T h(Y,Z) + \frac{\sigma}{2}\|h(Y,Z)\|^2$$

$$\text{s.t.} \quad Ye_i \in K \quad \forall\, i \geq 0$$

$$Ze_i \in K \quad \forall\, i \geq 1$$

$$Y_{00} = 1$$

Unlike before, optimization is *not* separable due to quadratic term in objective.

Augmented Lagrangian for $N_+(K)$

Aug Lag not immediately applicable to constraint $Y \succeq 0$. Consider alternative optimization over $N_+(K)$:

$$\min \quad \bar{c}^T Y e_0$$

s.t. [same constraints as for $N(K)$]

$$Y = U, \quad U \succeq 0$$

Augmented Lagrangian for $N_+(K)$

$$A_\sigma(\lambda, S) := \min \; \bar{c}^T Y e_0 + \lambda^T h(Y,Z) + \frac{\sigma}{2}\|h(Y,Z)\|^2 + S \bullet (U - Y) + \frac{\sigma}{2}\|U - Y\|_F^2$$

s.t. [same constraints as $A_\sigma(\lambda)$]

$$U \succeq 0$$

Unlike before, optimization is *not* separable due to quadratic term in objective.

Solving the Augmented Lagrangian Subproblems

- To take advantage of structure in constraints, use coordinate descent
- Iteratively solve over the columns of *Y*, *Z* and over *U*
- Each step in coordinate descent is a convex QP
- For *Y* and *Z*, use simplex-type QP algorithm, taking advantage of advance basis information to hotstart
- For *U*, use projection on positive semidefinite cone

$$\mathrm{proj}_{\succeq}\left(Y - \frac{1}{\sigma}S\right)$$

Augmented Lagrangian Method

Aug Lagrangian Algorithm (semidefinite case)

$\lambda = 0,\ S = 0,\ \sigma = 1$

loop

Find $(Y, Z, U) \in \text{Argmin}\, A_\sigma(\lambda, S)$

$\lambda \leftarrow \lambda + \sigma\, h(Y, Z)$

$S \leftarrow \text{proj}_{\succeq}(S - \sigma\,(Y - U))$

$\sigma \leftarrow \eta\,\sigma$, where $\eta \geq 1$

end loop

Comments on Augmented Lagrangian Method

- No difficulty in choosing stepsize
- Much better convergence than subgradient method
- Computations show it does not pay to solve subproblems to optimality, i.e., only a few iterations of coordinate descent are required
- Finds primal and dual solutions
- Closely related to bundle method

Computational results

- Computations done on a 2.4 GHz Pentium 4
- Projections onto cone done with LAPACK
- QPs solved with CPLEX 8.1 simplex-type algorithm
- Steadily increase penalty parameter
- One iteration of coordinate descent each time

Quadratic Assignment

$$\min \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} a_{ij} b_{kl} x_{ik} x_{jl}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{ik} = 1 \quad \forall\, k = 1, \ldots, n$$

$$\sum_{k=1}^{n} x_{ik} = 1 \quad \forall\, i = 1, \ldots, n$$

$$x_{ik} \in \{0, 1\} \quad \forall\, i, k = 1, \ldots, n$$

Quadratic Assignment

Comparison of gap (%) with other bounds:

Problem	GLB	QPB1	RS3	*N(K)*	$N_+(K)$
nug12	14.7	16.6	3.6	9.5	1.7
nug14	16.0	12.1	2.2	9.0	0.4
nug15	16.3	13.4	2.4	9.5	0.8
nug16a	18.4	10.1	2.5	11.5	0.8
nug16b	17.6	13.6	4.2	12.3	1.7
nug17	19.9	11.7	3.6	13.1	1.4
nug18	19.5	11.7	4.0	13.9	1.9
nug20	20.0	12.3	4.6	15.1	2.5
nug21	24.8	15.7	4.7	17.1	2.5
nug22	31.0	14.4	4.3	20.9	2.3
nug24	23.3	13.2	5.1	17.8	2.6
nug25	23.4	12.6	5.6	17.8	3.3
nug27	29.3	*n/a*	5.1	21.6	2.3
nug28	26.7	*n/a*	5.1	21.6	2.9
nug30	25.4	8.0	5.2	21.6	3.1

GLB = Gilmore, Lawler
QPB1 = Anstreicher, Brixius
RS3 = Rendl, Sotirov

Quadratic Assignment

Problem	GLB	QPB1	RS3	*N(K)*	$N_+(K)$
rou12	14.1	12.5	5.0	4.8	0.1
rou15	15.7	14.2	5.9	8.3	1.1
rou20	17.3	16.2	8.5	11.4	4.2
scr12	11.3	72.7	6.7	5.1	0.0
scr15	12.5	75.6	4.5	3.7	0.0
scr20	30.2	78.2	13.7	13.6	3.9
esc16a	44.1	19.1	13.2	29.4	5.9
esc16b	24.7	14.4	1.37	4.8	0.7
esc16c	48.1	40.6	11.3	26.3	3.8
esc16d	81.3	218.8	50.0	75.0	18.8
esc16e	57.1	78.6	17.9	50.0	3.6
esc16g	53.9	65.4	23.1	46.2	3.9
esc16h	37.3	28.9	2.6	29.3	1.9
esc16i	100.0	278.6	35.7	100.0	14.3
esc16j	87.5	175.0	12.5	75.0	0.0

Quadratic Assignment

Problem	GLB	QPB1	RS3	$N(K)$	$N_+(K)$
had12	7.0	3.6	0.5	1.8	0.0
had14	8.5	3.5	0.3	2.1	0.0
had16	9.7	3.4	0.6	4.3	0.1
had18	10.9	4.0	0.8	5.1	0.1
had20	10.9	3.5	0.5	5.0	0.2
kra30a	23.1	22.9	12.9	14.5	2.5
kra30b	24.5	24.5	11.2	16.1	4.1
tai12a	12.7	11.1	0.7	1.0	0.0
tai15a	15.6	14.9	6.0	9.1	2.9
tai17a	16.1	15.4	8.2	10.0	3.1
tai20a	17.5	16.8	9.4	12.1	4.5
tai25a	17.6	15.8	10.8	14.3	4.7
tai30a	17.2	16.5	9.1	13.7	6.1

Quadratic Assignment

Problem	HH	$N_+(K)$
had16	0.0	0.1
had18	0.0	0.1
had20	0.0	0.2
kra30a	3.0	2.5
kra30b	4.7	4.1
nug12	0.0	1.7
nug15	0.0	0.8
nug20	3.2	2.5
nug30	6.1	3.1
rou15	0.0	1.1
rou20	3.6	4.2
tai20a	3.9	4.5
tai25a	6.5	4.7
tai30a	7.3	6.1
tho30	8.8	4.8

HH =
Hahn-Hightower
level-2 RLT

Problem of Erdös-Turán

Find largest subset of {1,...,n} s.t. no three numbers are in arithmetic progression:

$$\max \sum_{i=1}^{n} x_i$$

$$\text{s.t.} \quad x_{i_1} + x_{i_2} + x_{i_3} \leq 2 \quad \text{if } i_1 + i_2 = 2i_3$$

$$x \in \{0,1\}^n$$

	Upper Bound		Time (sec)	
n	***N(K)***	**CPLEX**	***N(K)***	**CPLEX**
60	34.34	34.29	628.48	2349.68
70	40.08	40.00	1176.53	10625.88
80	45.84	45.85*	2064.92	15000.00*
90	51.45	52.93*	3652.16	15000.00*
100	57.45	60.38*	4994.79	15000.00*

* Time limit reached

Standard SDP methods could not solve $N_+(K)$ for n > 30

Inexact computation for $N_+(K)$ by Dash

Future Work

- Specialized algorithms for solving QP subproblems
- Other types of large-scale LPs
- Relaxations of problems with a mix of binary and continuous variables
- Relaxations for nonconvex QPs

Relationship with Bundle

$$\min\{c^T y \mid Ay = b, y \in X\}$$

Given before any iteration:

Bundle: $\{y^k\} \subset X$ Current best dual: $\bar{\lambda}$

Define: $\tilde{X} = \mathrm{conv}(\{y^k\})$

Solve:

$$\max_{\lambda} \min_{y \in \tilde{X}} \; c^T y + \lambda^T (b - Ay) - \frac{\rho}{2} \|\lambda - \bar{\lambda}\|^2$$

$$= \min_{y \in \tilde{X}} \max_{\lambda} \; c^T y + \lambda^T (b - Ay) - \frac{\rho}{2} \|\lambda - \bar{\lambda}\|^2$$

Relationship with Bundle

Optimal λ: $\boxed{\lambda = \bar{\lambda} + \frac{1}{\rho}(b - Ay)}$

$$\therefore \quad \min_{y \in \tilde{X}} \; c^T y + \bar{\lambda}^T (b - Ay) + \frac{1}{2\rho} \|b - Ay\|^2$$

Substituting $\sigma \leftrightarrow 1/\rho$ and $X \leftrightarrow \tilde{X}$:

$$\boxed{\min_{y \in X} \; c^T y + \bar{\lambda}^T (b - Ay) + \frac{\sigma}{2} \|b - Ay\|^2}$$