Separation Techniques for Constrained Nonlinear 0–1 Programming

Christoph Buchheim

Computer Science Department, University of Cologne
and
DEIS, University of Bologna

MIP 2008, Columbia University, New York City

Topic of this Talk

Solve quadratic (or nonlinear) variants of easy (or well-studied) combinatorial optimization problems.
[e.g., quadratic assignment or quadratic knapsack]

What happens when combining an easy binary IP-formulation with a quadratic (or nonlinear) objective function?

Two reasons to consider such problems:

– natural way to model many applications

– easiest type of constrained nonlinear 0–1 programs
[the building blocks are well-known]

Topic of this Talk

Straightforward approach:

– replace each nonlinear term by a new variable

– add linear constraints linking new variables to old ones

– add the constraints of the original linear problem

Clearly yields a correct ILP for the nonlinear problem variant, but the induced LP-relaxation is very weak in general.

How can this be improved?

Topic of this Talk

1. Reduction to the Quadratic Case
   [joint work with Giovanni Rinaldi]
2. Example: Quadratic Linear Ordering
   [joint work with Angelika Wiegele and Lanbo Zheng]
3. Example: Power Cost Problems
   [joint work with Frank Baumann]

Reduction to the Quadratic Case

Reduction to Quadratic Case

Objective:

Show that for most types of nonlinear objective functions the problem can be reduced efficiently to the quadratic case.

Special case: polynomial 0–1 optimization

B. and Rinaldi: Efficient reduction of polynomial zero-one optimization to the quadratic case, SIAM J Opt [2007]

Unconstrained Pseudo-boolean Optimization

Consider boolean functions build up recursively by arbitrary unary or binary operators $\{0,1\}^2 \rightarrow \{0,1\}$.

Problem:

Maximize a pseudo-boolean function given as a weighted sum of such boolean functions.

Example:

maximize
$2 \cdot (\neg a \vee (b \wedge \neg c \wedge \neg d)) - 4 \cdot (\neg a \vee \neg c) + 3 \cdot (c \wedge d) - 2 \cdot (a \Leftrightarrow \neg(b \wedge c))$
s.t. $a, b, c, d \in \{0,1\}$

Special cases:

– unconstrained polynomial 0–1 optimization
– maximum satisfiability...

Linearization

Standard linearization:

– introduce binary variables for all appearing boolean functions
– add linear constraints linking these variables on each level

Example

$$2\cdot\big(\neg a \vee (b \wedge c \wedge \neg d)\big) - 4\cdot\big(\neg a \vee \neg c\big) + 3\cdot\big(c \wedge d\big) - 2\cdot\big(a \Leftrightarrow \neg(b \wedge c)\big)$$

original variables:

$x_a \quad x_b \quad x_c \quad x_d$

new variables:

$x_{\neg a \vee (b \wedge c \wedge \neg d)} \quad x_{\neg a \vee \neg c} \quad x_{c \wedge d} \quad x_{a \Leftrightarrow \neg(b \wedge c)}$

$x_{\neg a} \quad x_{b \wedge c \wedge \neg d} \quad [x_{\neg a}] \quad x_{\neg c} \quad [x_c] \quad [x_d] \quad [x_a] \quad x_{\neg(b \wedge c)}$

$[x_a] \quad x_{b \wedge c} \quad x_{\neg d} \quad [x_c] \quad [x_{b \wedge c}]$

$[x_b] \quad [x_c] \quad [x_d]$

Quadratic Case

Let F denote the set of all variables after linearization.
Let $P(F) \subseteq \mathbb{R}^F$ be the convex hull of feasible solutions.

Theorem:
Let all boolean functions contain at most one operator.
Then $P(F) \cong \mathcal{C}(G)$ for some graph G on at most $|F|$ edges.

Proof:

$$\begin{aligned} a \circ b \quad = \quad & 1/2(-0 \circ 0 + 0 \circ 1 + 1 \circ 0 - 1 \circ 1) \cdot a \oplus b \\ & +1/2(-0 \circ 0 - 0 \circ 1 + 1 \circ 0 + 1 \circ 1) \cdot a \\ & +1/2(-0 \circ 0 + 0 \circ 1 - 1 \circ 0 + 1 \circ 1) \cdot b \\ & +(0 \circ 0) \end{aligned} \qquad \square$$

Can this be generalized to higher-degree objective functions?

Reduction to Quadratic Case

Recipe:

1. introduce a copy $\tilde{x}_f$ of every connection variable x_f
2. replace x_f by $\tilde{x}_f$ wherever appearing as an operand
3. introduce two more terms $x_f^1 = \tilde{x}_g \wedge \tilde{x}_f$ and $x_f^2 = \tilde{x}_h \wedge \tilde{x}_f$ for every connection variable $x_f = x_g \circ x_h$
4. the result is a quadratic instance with a polytope $P(\tilde{F})$ isomorphic to some cut polytope $\mathcal{C}(G)$
5. intersect $P(\tilde{F})$ with the hyperplane $\tilde{x}_f = x_f$ for all f
6. intersect $P(\tilde{F})$ with the hyperplanes that correctly link both x_f^1 and x_f^2 to x_f, x_g, x_h
7. call the resulting polytope $P^\star(F)$

Clear: $P^\star(F) \subseteq \mathcal{C}(G)$ is a relaxation of $P(F)$

Reduction to Quadratic Case

Theorem:
$P^\star(F)$ is a face of $P(\tilde{F})$, thus $P^\star(F) = P(F)$.

Corollary:
$P(F)$ is a face of $\mathcal{C}(G)$, where G has at most $4|F|$ edges.

Hence

- the separation problem for $P(F)$ reduces to the separation problem for $\mathcal{C}(G)$ (by a very simple transformation)
- in a branch-and-cut approach, separation can be done for $\mathcal{C}(G)$, the rest for $P(F)$

Works very well in practice!

Constraints?

What about constraints in the original nonlinear problem?

If linear, they remain unaffected!

The polytope spanned by all feasible solutions of the linearized linearly constrained **pseudo-boolean** problem becomes a face of a polytope spanned by all feasible solutions of a linearized linearly constrained **quadratic** problem...

In other words:
forget pseudo-boolean objective functions and concentrate on quadratic ones

Example: Quadratic Linear Ordering

Linear Ordering

Linear Ordering problem:

Given a set of elements $\{1, \ldots, n\}$ and costs $c_{ij} \in \mathbb{R}$ for all $i < j$.
Find a permutation $\pi \in S_n$ minimizing

$$\sum_{\pi(i)<\pi(j)} c_{ij}$$

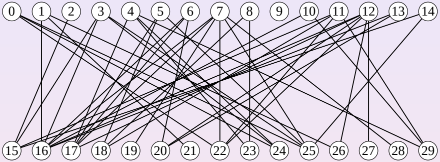
ILP model:

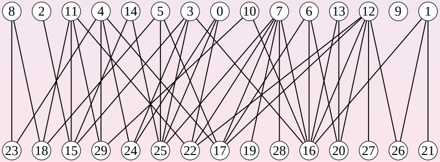
$$\begin{array}{llcll}
\min & c^\top x & & & \\
\text{s.t.} & x_{ij} + x_{jk} - x_{ik} & \geq & 0 & \text{for all } i < j < k \\
 & x_{ij} + x_{jk} - x_{ik} & \leq & 1 & \text{for all } i < j < k \\
 & x_{ij} & \in & \{0,1\} & \text{for all } i < j \; .
\end{array}$$

The 3-dicycle inequalities $0 \leq x_{ij} + x_{jk} - x_{ik} \leq 1$ model transitivity.

Bipartite Crossing Minimization

Bipartite Crossing Minimization:





Find permutations minimizing the number of edge crossings.

Quadratic Linear Ordering Polytope

Let $QLO(n)$ be the polytope corresponding to the linearized quadratic linear ordering problem on n elements.

Lemma
For all $i < j < k$ and all binary x, the two inequalities

$$0 \leq x_{ij} + x_{jk} - x_{ik} \leq 1$$

are equivalent to the single quadratic equation

$$x_{ik} - x_{ij}x_{ik} - x_{ik}x_{jk} + x_{ij}x_{jk} = 0 \; .$$

Lemma
The (linearized) constraints $x_{ik} - x_{ij}x_{ik} - x_{ik}x_{jk} + x_{ij}x_{jk} = 0$ form a minimal equation system for $QLO(n)$.

Quadratic Linear Ordering Polytope

Consider the unconstrained quadratic optimization problem over $x \in \{0,1\}^{\binom{n}{2}}$, and the corresponding polytope BQP.

Theorem

Each (linearized) constraint $x_{ik} - x_{ij}x_{ik} - x_{ik}x_{jk} + x_{ij}x_{jk} = 0$ is face-inducing for BQP. Thus $QLO(n)$ is a face of BQP.

Consequences:

– knowledge of $LO(n)$ is useless for understanding $QLO(n)$

– separation from $QLO(n)$ essentially means separation from the corresponding unconstrained problem

– use an IP-based or an SDP-based approach for max-cut!

Bipartite Crossing Minimization

n	*d*	JM		LIN		MC-ILP		MC-SDP	
		#	time	#	time	#	time	#	time
10	10	10	0.02	10	0.01	10	0.30	10	1.16
10	20	10	0.05	10	0.74	10	1.01	10	2.25
10	30	10	0.15	10	14.95	10	4.55	10	4.77
10	40	10	0.33	10	51.20	10	12.17	10	5.07
10	50	10	0.61	10	180.86	10	18.31	10	4.71
10	60	10	1.14	10	738.58	10	27.34	10	5.35
10	70	10	2.35	8	1225.62	10	33.46	10	6.81
10	80	10	4.05	10	538.68	10	15.64	10	5.15
10	90	10	8.86	10	86.51	10	8.59	10	6.79
12	10	10	0.20	10	0.02	10	8.07	10	9.54
12	20	10	1.52	10	5.93	10	19.00	10	18.36
12	30	10	4.53	10	140.60	10	35.95	10	21.61
12	40	10	16.36	7	1808.35	10	106.01	10	25.29
12	50	10	57.05	0	—	10	440.96	10	44.84
12	60	10	102.15	0	—	10	622.10	10	48.26
12	70	10	211.37	0	—	10	607.73	10	40.31
12	80	10	527.75	0	—	10	273.39	10	28.71
12	90	10	1036.30	6	1693.75	10	73.60	10	22.21

[Running times on Intel Xeon processor with 2.33 GHz, limit 1h, 10 instances/row]

Bipartite Crossing Minimization

n	d	JM		LIN		MC-ILP		MC-SDP	
		#	time	#	time	#	time	#	time
14	10	10	15.68	10	0.33	10	19.02	10	41.03
14	20	10	110.83	10	102.07	10	155.14	10	89.61
14	30	10	747.49	4	1267.86	10	688.01	10	132.72
14	40	9	1432.45	0	—	8	1667.63	10	144.03
14	50	2	2718.05	0	—	1	1453.35	10	180.49
14	60	0	—	0	—	1	2594.94	10	141.93
14	70	0	—	0	—	5	2177.86	10	149.68
14	80	0	—	0	—	7	1829.18	10	145.97
14	90	0	—	0	—	10	398.75	10	81.27
16	10	8	328.92	10	2.77	10	190.83	10	124.57
16	20	5	2220.12	7	809.30	9	882.19	10	309.31
16	30	0	—	0	—	4	2112.61	10	630.77
16	40	0	—	0	—	0	—	9	800.87
16	50	0	—	0	—	0	—	7	451.09
16	60	0	—	0	—	0	—	9	403.82
16	70	0	—	0	—	0	—	8	789.62
16	80	0	—	0	—	0	—	10	568.55
16	90	0	—	0	—	7	2373.15	10	362.29

[Running times on Intel Xeon processor with 2.33 GHz, limit 1h, 10 instances/row]

Bipartite Crossing Minimization

Typical situation:

– knowing the original polytope usually doesn't help at all

– standard linearization without separation performs poorly

– quadratic reformulation usually yields stronger constraints

– sometimes this reformulation yields faces of cut polytopes

– even without reformulation, max-cut separation is useful

Example: Power Cost Problems

Power Cost Problems

Ad-hoc networks

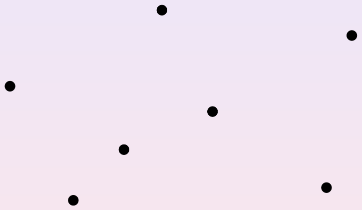
Given n points in the plane and, for every pair of points (i,j), the power c_{ij} that is necessary to transmit data from i to j.

[usually $c_{ij} = d(i,j)^\kappa$, with $\kappa > 1$]

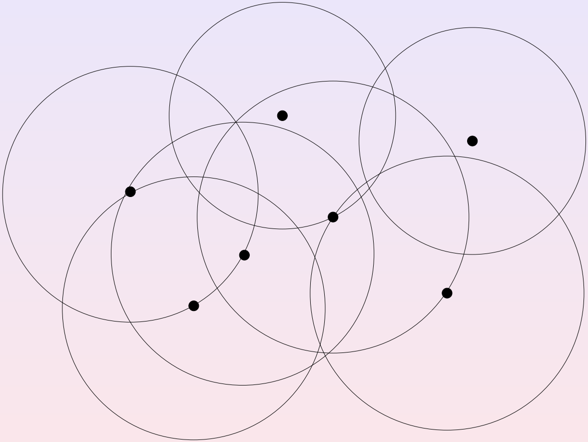
Let $V = \{1,\ldots,n\}$. Any power assignment $p\colon V \to \mathbb{R}$ defines a graph on the nodes V, by setting

$$(i,j) \in E \quad \Longleftrightarrow \quad p(i), p(j) \geq c_{ij} \ .$$

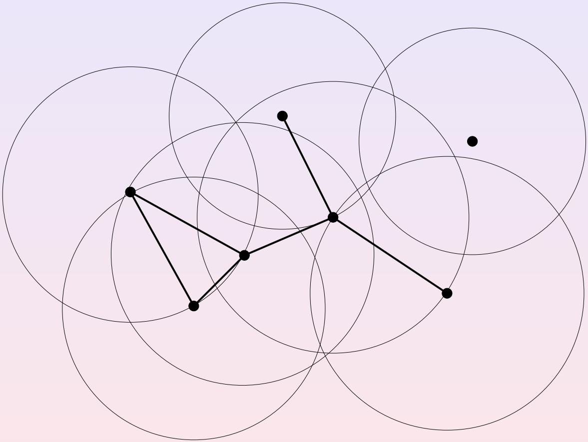
Power Cost Problems



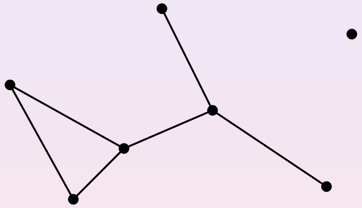
Power Cost Problems



Power Cost Problems



Power Cost Problems



Power Cost Problems

The aim is to **minimize** the total power consumption such that the resulting graph has certain connectivity properties (connected, k-connected, s-t-path...)

Nonlinear model:

$$\begin{array}{ll} \min & \sum_{i \in V} \max\{c_{ij} x_{ij} \mid j \neq i\} \\ \text{s.t.} & x \in X \end{array}$$

where $X = \{\text{incidence vectors of feasible graphs on } V\}$.
[e.g., $X = \{\text{spanning trees of } K_n\}$.]

Linearize by introducing power variables $y_i \in \mathbb{R}$...

Linearization

Nonlinear model:

$$\begin{array}{ll} \min & \sum_{i \in V} \max\{c_{ij} x_{ij} \mid j \neq i\} \\ \text{s.t.} & x \in X \end{array}$$

Linearized model:

$$\begin{array}{llll} \min & \sum_{i \in V} y_i & & \\ \text{s.t.} & x \in X & & \\ & y \in \mathbb{R}^n & & \\ & y_i \geq c_{ij} x_{ij} & \text{for all } i \in V \text{ and } j \neq i \end{array}$$

Linearization

Same situation as always:

– standard linearization yields very weak LP-relaxation
– crucial improvement by addressing unconstrained problem!

Unconstrained problem: replace X by $\{0,1\}^{\binom{n}{2}}$

$$\begin{array}{llcll} \text{min} & \sum_{i \in V} y_i & & & \\ \text{s.t.} & x & \in & \{0,1\}^{\binom{n}{2}} & \\ & y & \in & \mathbb{R}^n & \\ & y_i & \geq & c_{ij} x_{ij} & \text{for all } i \in V \text{ and } j \neq i \end{array}$$

Can be reduced to the case of a single power variable...

Modeling Weighted Maxima

Theorem

If the c_i are pairwise distinct, then the polyhedron

$$P = \text{conv}\big\{(x, y) \in \{0,1\}^k \times \mathbb{R} \mid y \geq \max\{c_1 x_1, \ldots, c_k x_k\}\big\}$$

has 2^{k-1} facets, which can be separated in $O(k \log k)$ time.

<u>Solution approach:</u>

– solve the linearized problem with a branch-and-cut algorithm
– add inequalities necessary to describe X
– add separation algorithm for P

Preliminary computational results show that this approach outperforms the currently best problem-specific algorithms.

Much more flexible than other approaches, works for any X.

Conclusions & Experimental Experience

When combining linear constraints with nonlinear objective functions, the most important task is to address the nonlinear structure itself.

– for quadratic problems, try max-cut separation

– for pseudo-boolean objective function, try the reduction to the quadratic case

– for other types of nonlinearity, try to understand the unconstrained problem