Compact linearization for bilinear mixed-integer problems

Leo Liberti

DEI, Politecnico di Milano, P.zza L. da Vinci 32, 20133 Milano, Italy

(liberti@elet.polimi.it)

April 27, 2005

Abstract

We present a compact linearization for a broad class of bilinear 0-1 mixed-integer problems subject to assignment constraints. We apply the linearization to three classes of problems: quadratic assignment, multiprocessor scheduling with communication delays, and graph partitioning, and show that it yields faster solution times.

1 Introduction

Solving Bilinear Mixed-Integer Problems (BMIP)s of combinatorial nature is a very hard problem (although polynomially solvable instance classes exist, see e.g. [AFLS01]), yet has a wealth of applications [PR96, HR68]. Usually, 0-1 BMIPs can be reformulated exactly to MILPs with a large number of variables and constraints [For60, Bea98] as shown in Sect. 2. A complete treatment of linearizations for some classes of BMIPs can be found in [PR96], where a very detailed polyhedral study of the problem polytope is performed for some classes of BMIPs.

In this paper, we present a compact linearization (Sect. 3) for a class of BMIPs subject to various subsets of assignment constraints. This linearization arose over twenty years ago as a reformulation for the Quadratic Assignment Problem [FY83]. We show how to extend it to more general classes of BMIPs. We show the usefulness of our linearization by applying it to three problems in the literature (quadratic assignment, multiprocessor scheduling with communication delays and graph partitioning) in Sect. 4.

We remark that although Padberg et al. warn against blindly looking for compact formulations ([PR96], Sect. 3.1, p. 64) with no other merit than the small cardinality of their constraints set, we have three common-sense reasons for considering our compact linearization worthwhile:

- its ease and extent of application;
- branch-and-bound algorithms require solving an LP relaxation at each step, and the speed of solution of an LP depends largely on the number of constraints;
- computational experience shows that it often gives rise to tighter LP relaxations than the usual linearization (most commonly this happens in branch-and-bound tree nodes deeper than the root node — a detailed computational study of this phenomenon has not been performed yet).

Although we do not think that our result can in any way supplant a sound polyhedral analysis of the problem polytope, we nonetheless believe that our method can be used to quickly derive compact linearizations for BMIPs for which such an analysis is not yet available.

2 Usual linearization

Let $G = (V, E)$ be an undirected graph ($|V| = n, |E| = m$), possibly with self-loops but without isolated vertices, where the nodes represent the problem variables and the edges represent the bilinear products between variables (G encodes the bilinear structure of the problem). Let $x \in \{0,1\}^n$ be the problem variables, and w a set of non-negative linearization variables satisfying $w_{ij} = x_i x_j$ for all $\{i,j\} \in E$. By commutativity of product, the following also hold:

$$\forall \{i,j\} \in E \quad (w_{ij} = w_{ji}). \tag{1}$$

Let $N = \{1, \ldots, n\}$ and consider a subset of indices $K \subseteq N$. Let $\{I_k \mid k \in K\}$ be a covering of N, i.e.

$$N \subseteq \bigcup_{k \in K} I_k.$$

We consider the following problem P:

$$\min_{x,w} c^\mathsf{T} x + b^\mathsf{T} w \tag{2}$$

$$g(x,w) \leq 0 \tag{3}$$

$$\forall k \in K \quad \sum_{i \in I_k} x_i = 1 \tag{4}$$

$$\forall \{i,j\} \in E \quad w_{ij} = x_i x_j \tag{5}$$

$$\forall i \in N \quad x_i \in \{0,1\} \tag{6}$$

$$\forall \{i,j\} \in E \quad w_{ij} \geq 0 \tag{7}$$

where $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$, and g is a q-vector of functions of x, w.

The linearization below, which we call *usual linearization*, was proposed originally in [For60] and discussed in [HR68], Sect. 7, Thm. 4. It is possibly the best known linearization for bilinear products of binary variables. We reformulate problem P exactly to a MILP by introducing the following linearization constraints:

$$\forall \{i,j\} \in E \quad (w_{ij} \leq x_i) \tag{8}$$

$$\forall \{i,j\} \in E \quad (w_{ij} \leq x_j) \tag{9}$$

$$\forall \{i,j\} \in E \quad (w_{ij} \geq x_i + x_j - 1), \tag{10}$$

and removing the bilinear constraints (5). If either x_i or x_j is zero, then by (8) or (9) we have $w_{ij} = 0$. If $x_i = x_j = 1$ then by (10) $w_{ij} = 1$. Hence this is an exact reformulation, which we indicate with $L_1(P)$. Notice that this linearization implies the addition of $(3m)$ constraints to the model: in practice constraints (1) can be removed from the formulation by substituting each w_{ji} with w_{ij}, so they do not actually add to the number of constraints; rather, they decrease the number of variables in the problem. Computationally, this is usually carried out automatically by the presolver code, present in most good quality MILP solvers.

3 Compact linearization

Let $\{J_k \subseteq N \mid k \in K\}$ be a cover of N. We assume that the following edge-covering condition holds:

$$E \subseteq \bigcup_{k \in K} (I_k \times J_k). \tag{11}$$

We multiply (4) by the x_j variables (where $j \in J_k$) to form the following system:

$$\forall k \in K, j \in J_k \quad (\sum_{i \in I_k} x_i x_j = x_j),$$

then substitute $w_{ij} = x_i x_j$ to get the following linear system:

$$\forall k \in K, j \in J_k \quad (\sum_{i \in I_k} w_{ij} = x_j). \tag{12}$$

Up to this point, the process is similar to that described in the Reformulation-Linearization Technique [SA86]. We then form the problem $L_2(P)$ by substituting constraints (8)-(10) in $L_1(P)$ with (1) and (12). We claim that $L_2(P)$ is an exact linearization of P. This is shown in the following theorem by deriving (8)-(10) from (1) and (12).

Note that condition (11) means in practice that we need a "large enough" set of assignment constraints and multiplying variables.

3.1 Theorem
Provided condition (11) holds, $L_2(P)$ is an exact reformulation of $L_1(P)$.

Proof. By (12), since the sum of the w variables is equal to one of the x variables, we infer that just one of the w variables in the sum must be less than or equal to the x variable. Furthermore, by the covering condition (11), all pairs $\{i, j\}$ are considered. Hence,

$$\forall \{i, j\} \in E \quad (w_{ij} \leq x_j), \tag{13}$$

which establishes (8). By (13) and (1), with a swap of i and j, we have:

$$\forall \{i, j\} \in E \quad (w_{ij} \leq x_i), \tag{14}$$

which establishes (9). Now, any sum of (14) preserves the ordering relation, so:

$$\forall k \in K, j \in J_k, i \in I_k \quad (\sum_{f \in I_k \setminus \{i\}} w_{fj} \leq \sum_{f \in I_k \setminus \{i\}} x_f). \tag{15}$$

We add and subtract w_{ij} from LHS of (15):

$$\forall k \in K, j \in J_k, i \in I_k \quad (\sum_{f \in I_k} w_{fj} - w_{ij} \leq \sum_{f \in I_k \setminus \{i\}} x_f). \tag{16}$$

We substitute $x_j = \sum_{f \in I_k} w_{fj}$ (which holds by (12)) in (16):

$$\forall k \in K, j \in J_k, i \in I_k \quad (x_j - w_{ij} \leq \sum_{f \in I_k \setminus \{i\}} x_f). \tag{17}$$

We add and subtract x_i from from RHS of (17):

$$\forall k \in K, j \in J_k, i \in I_k \quad (x_j - w_{ij} \leq \sum_{f \in I_k} x_f - x_i). \tag{18}$$

Finally, we substitute $1 = \sum_{f \in I_k} x_f$ (which holds by 4) in (17):

$$\forall k \in K, j \in J_k, i \in I_k \quad (x_j - w_{ij} \leq 1 - x_i), \tag{19}$$

which, by condition (11), establishes (10). □

The number of added constraints (12) is $R = \sum_{k \in K} |J_k|$. The worst case is $K = \{1, \ldots, n\}, I_k = \{k\}, J_k = \{1, \ldots, n\}$ for all $k \in K$, yielding $R = n^2$. Compared to $3m$ constraints in the usual linearization described in Sect. 2, this does not amount to any significant reduction in problem size (indeed, if the bilinear structure is sparse, this may increase the problem size!). However, in practice it is much more common to find one assignment constraint $\sum_{i=1}^{n} x_i = 1$ and to multiply it by each x_j $(j \leq n)$ in turn,

yielding n linearization constraint and a reduction of one order of magnitude in the number of constraints. This can also be seen from the case studies in Sect. 4.

It should appear clear that the greatest reduction is obtained by minimizing R subject to condition (11). In may sometimes pay to solve the corresponding covering problem (at least heuristically) as a pre-processing step. A similar issue when P is a nonconvex continuous optimization problem is discussed in [Lib04].

4 Applications

In this section we shall present some applications of the compact linearization described in this paper.

4.1 Quadratic Assignment problem

The Quadratic Assignment Problem (QAP) can be formulated as the following bilinear integer problem (BIP):

$$\left.\begin{array}{rrcl} \min\limits_{x} & \sum\limits_{i,j,k,l}^{n} a_{ij}b_{kl}x_{ik}x_{jl} & + & \sum\limits_{i,j}^{n} c_{ij}x_{ij} \\ \forall i \leq n & \sum\limits_{j=1}^{n} x_{ij} & = & 1 \\ \forall j \leq n & \sum\limits_{i=1}^{n} x_{ij} & = & 1 \\ \forall i,j \leq n & x_{ij} & \in & \{0,1\}, \end{array}\right\} \tag{20}$$

where (x_{ij}) is an $n \times n$ array of binary variables and $A = (a_{ij}), B = (b_{ij}), C = (c_{ij})$ are given $n \times n$ matrices. This is known as the Koopmans-Beckmann formulation [KB57]. We multiply both sets of assignment constraints by each problem variable x_{ij} $(i, j \leq n)$. We then linearize the resulting bilinear products by imposing $\forall i, j, k, l \leq n(w_{ijkl} = x_{ij}x_{kl})$ and obtaining

$$\begin{aligned} \forall i,k,l \leq n \ (\sum_{j=1}^{n} w_{ijkl} &= x_{kl}) \\ \forall j,k,l \leq n \ (\sum_{i=1}^{n} w_{ijkl} &= x_{kl}). \end{aligned}$$

By Thm. 3.1, we can use the above constraints to replace the usual linearizing constraints $\forall i, j, k, l \leq n(w_{ijkl} \leq x_{ij} \wedge w_{ijkl} \leq x_{kl} \wedge w_{ijkl} \geq x_{ij} + x_{kl} - 1)$. The resulting linearization is known as the Frieze-Yadegar formulation [FY83], and is one of the most compact (and tight) linearizations of the QAP. In this setting, Thm. 3.1 provides a new proof of the validity of the Frieze-Yadegar formulation.

4.2 Multi-processor Scheduling with Communication Delays

The Multi-processor Scheduling problem with Communication Delays (MSPCD) arises in parallel computing. It consists of scheduling dependent tasks with communication delays (due to data transfer) onto homogeneous, arbitrary connected multiprocessor architecture such that the total completion time is minimum. The model [DMM03] is further complicated by the fact that communication delays between tasks also depend on what processors the tasks are being executed on; namely, we assume that the connections among the processors do not form a complete graph, so transferring data from a processor to an adjacent processor requires less time than between non-adjacent processors. The MSPCD then can be formulated

as follows:

$$\left.\begin{array}{rrcl}
\min\limits_{y,t} & \max\limits_{j\le n}\{t_j+L_j\} & & \\
\forall j\le n & \sum\limits_{k=1}^{p}\sum\limits_{s=1}^{n} y^s_{jk} & = & 1\\
\forall k\le p & \sum\limits_{j=1}^{n} y^1_{jk} & \le & 1\\
\forall k\le p, s\ge 2 & \sum\limits_{j=1}^{n} y^s_{jk}-\sum\limits_{j=1}^{n} y^{s-1}_{jk} & \le & 0\\
\forall j\le n, i\in \mathrm{Pred}(j) & t_i+L_i+\sum\limits_{k=1}^{p}\sum\limits_{s=1}^{n}\sum\limits_{l=1}^{p}\sum\limits_{r=1}^{n}\gamma^{kl}_{ij}y^s_{ik}y^r_{jl} & \le & t_j\\
\forall i,j\le n, k\le p, s\le n-1 & t_i+L_i-\alpha\left[2-\left(y^s_{ik}+\sum\limits_{r=s+1}^{n} y^r_{jk}\right)\right] & \le & t_j\\
\forall j\le n & T^L-L_j & \le & t_j\\
\forall j,s\le n, k\le p & y^s_{jk}\in\{0,1\} & &
\end{array}\right\} \quad (21)$$

where p is the number of processors, n is the number of tasks, $\mathrm{Pred}(j)$ is the set of immediate predecessors of task j, L_j is the length of task j, γ^{kl}_{ij} is the communication delay between tasks i and j when they are executed on processors k and l, α is a sufficiently large penalty coefficient, y^s_{jk} is a binary variable set to 1 if task j is the s-th process to be executed on processor k and t_j is the starting time of process j. T^L is a lower bound on the total completion time calculatd either with Load Balancing or with a Critical Path Method (CPM) applied to the task precedence graph where the arcs are weighted by the running times L_j. The linearization of this formulation is obtained by replacing each bilinear product $y^r_{jl}y^s_{ik}$ by a variable w^{sr}_{ijkl} and adding the usual linearization constraints (8)-(10) to the model.

The sets $\mathrm{Pred}(j)$ define a digraph topology on the tasks. For each pair of tasks $i,j\le n$, we define an arc (i,j) if $i\in\mathrm{Pred}(j)$, obtaining a digraph $G=(V,A)$ where V is the set of tasks and an arc (i,j) is in A if task i has to be completed before task j. We then consider the graph $G'=(V',E')$ associated to G which describes the bilinear products present in the problem. The vertices in V' are triples (i,s,k), where $i,s\le n$ and $k\le p$. There is an edge in E' between (i,s,k) and (j,r,l) if and only if there is an arc (i,j) or (j,i) in A. We define $K=\{1,\dots n\}$, $J_0=\{(j,l,r)\mid l\le p, j,r\le n\}$

$$\begin{array}{rcl}
\forall\, k\in K\ (I_k & = & \{(k,h,s)\mid h\le p\wedge s\le n\})\\
\forall\, k\in K\ (J_k & = & J_0),
\end{array}$$

and note that condition (11) is satisfied by the index sets I_k, J_k. Thus, by Thm. 3.1, on multiplying the assignment constraints $\forall j\le n\sum_{k=1}^{p}\sum_{s=1}^{n} y^s_{jk}=1$ by all problem variables y^r_{il}, we obtain a set of n^4p constraints

$$\forall i,j,s,r\le n, l\le p\ (\sum_{k=1}^{p}\sum_{s=1}^{n} w^{rs}_{ijlk}=y^r_{il})$$

which reformulate the usual $3n^4p^2$ linearization constraints, which gives rise to a more compact formulation. In practice, when solving the problem with a Branch-and-Bound approach, each LP relaxation is solved much faster, thus yielding a significant decrease in solution time. See [DLMM04] for computational results, which indicate user CPU time improvements of up to two orders of magnitude.

4.3 Graph partitioning

This problem, also called Min-k-Cut, is a classic **NP**-hard problem that has many applications in several fields, and in particular in the design of telecommunication networks, where we want to minimize the amount of traffic between clusters of nodes. Many different formulations, of varying degrees of compactness, have recently been proposed in [Bou04]. Given an undirected graph $H=(U,F)$ and an integer $k\le|U|$, the problem consists of finding the k-partition of U which minimizes the number of edges $\{i,j\}$ where i,j are nodes in different sets of the partition (which are called clusters). To each vertex

$i \in U$ and for each cluster $h \leq k$, we associate a binary variable x_{ih} which is 1 if vertex i is in cluster h and 0 otherwise. We formulate the problem as follows:

$$\left.\begin{array}{rrcl} \min\limits_{x} & \frac{1}{2} \sum\limits_{h \neq l \leq k} \sum\limits_{\{i,j\} \in F} x_{ih} x_{jk} & & \\ \forall i \in U & \sum\limits_{h=1}^{k} x_{ih} & = & 1 \\ \forall h \leq k & \sum\limits_{i \in U} x_{ih} & \geq & 1 \\ \forall i \in U, h \leq k & x_{ih} & \in & \{0,1\}. \end{array}\right\} \quad (22)$$

The usual linearization of this BIP involves the addition of $k|F|$ continuous variables $0 \leq w_{ijhl} \leq 1$ to the formulation, as well as the introduction of $3|F|\frac{k(k-1)}{2}$ linearization constraints of the form (8)-(10) (one for each bilinear product $w_{ijhl} = x_{ih}x_{jl}$, for all $h \neq l, \{i,j\} \in F$ — keeping in mind that $w_{ijhl} = w_{jilh}$).

We consider the graph $G = (V, E)$ associated to H which encodes the bilinear structure of the problem. The set V consists of all pairs of indices (i,h) with $i \in U, h \leq k$. There is an edge in E between (i,h) and (j,l) if $h \neq l \leq k$ and $\{i,j\} \in F$. We define:

$$\begin{array}{rcl} \forall\, i \in U\ (I_i & = & \{(i,h) \mid h \leq k\}) \\ \forall\, j \in U\ (J_j & = & \{(j,l) \mid l \leq k\}), \end{array}$$

and now apply Thm. 3.1 to obtain the following $k|F|$ linearization constraints:

$$\forall \{i,j\} \in F, l \leq k\ (\sum_{h=1}^{k} w_{ijhl} = x_{jl}),$$

which replace the usual linearization constraints.

Table 1 reports computational results obtained over a few problem instances: `mesh` indicates a $\sqrt{|U|} \times \sqrt{|U|}$ square grid graph; `hypercube` is a $(\log_2 |U|)$-dimensional hypercubic graph; `random-0.5` is a randomly generated graph where each edge has $\frac{1}{2}$ generation probability. These results were obtained by solving the above formulations with CPLEX 8.1 [ILO02] on a 2.66GHz Pentium IV CPU with 1GB RAM running Linux. $|U|$ is the number of nodes and $|F|$ the number of edges in the graph; k is the number of subsets of the partition. The next two columns (labelled u) denote the number of linearization constraints LC and user time of CPU required to reach the optimal solution. The last two columns (labelled c) indicate comparative results on the same quantities for the compact linearization.

Instance	$\|U\|$	$\|F\|$	k	u: LC	CPU	c: LC	CPU
`mesh`	9	12	2	36	0.1s	24	0.03s
			5	360	34.22s	60	3.99s
			8	1008	1498.2s	96	783.48s
`hypercube`	16	32	2	96	0.38s	64	0.13s
			3	288	4.68s	96	2.18s
			5	960	2882.39s	160	125.79s
`random-0.5`	15	64	2	192	1.03s	128	0.23s
			3	576	8.03s	192	2.91s
			5	1920	1496.5s	320	218.2s

Table 1: The effect of compact linearization on the MIN-k-CUT problem.

These computational results show the typical improvements of our compact linearization on the MIN-k-CUT problem.

5 Conclusion

In this paper we derived a compact linearization which is applicable to mixed-integer problems involving bilinear products of binary variables, subject to assignment constraints. Our result is readily applicable to many classes of problems from the literature, and provides a definite improvement over the usual linearization of 0-1 bilinear products. We showed its application to quadratic assignment problems, multiprocessor scheduling problems with communication delays, and graph partitioning problems.

References

[AFLS01] K. Allemand, K. Fukuda, T.M. Liebling, and E. Steiner. A polynomial case of unconstrained zero-one quadratic optimization. *Mathematical Programming*, 91:49–52, 2001.

[Bea98] J.E. Beasley. Heuristic algorithms for the unconstrained binary quadratic programming problem. `http://mscmga.ms.ic.ac.uk/jeb/bqp.pdf`, 1998.

[Bou04] M. Boulle. Compact mathematical formulation for graph partitioning. *Optimization and Engineering*, 5:315–333, 2004.

[DLMM04] T. Davidović, L. Liberti, N. Maculan, and N. Mladenović. Mathematical programming-based approach to scheduling of communicating tasks. *Cahiers de GERAD*, G-2004-99, December 2004.

[DMM03] T. Davidović, N. Maculan, and N. Mladenović. A mathematical programming formulation for the multiprocessor scheduling problem with communication delays. In N. Mladenović and Đ. Dugošija, editors, *SYM-OP-IS Conference Proceedings, Herceg-Novi*, pages 331–334, Beograd, September 2003. Mathematical Institute, Academy of Sciences.

[For60] R. Fortet. Applications de l'algèbre de boole en recherche opérationelle. *Revue Française de Recherche Opérationelle*, 4:17–26, 1960.

[FY83] A.M. Frieze and J. Yadegar. On the quadratic assignment problem. *Discrete Applied Mathematics*, 5:89–98, 1983.

[HR68] P.L. Hammer and S. Rudeanu. *Boolean Methods in Operations Research and Related Areas.* Springer, Berlin, 1968.

[ILO02] ILOG. *ILOG CPLEX 8.0 User's Manual.* ILOG S.A., Gentilly, France, 2002.

[KB57] T.C. Koopmans and M.J. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25:53–76, 1957.

[Lib04] L. Liberti. Linearity embedded in nonconvex programs. *Journal of Global Optimization*, (to appear) 2004.

[PR96] M.W. Padberg and M.P. Rijal. *Location, Scheduling, Design and Integer Programming.* Kluwer, Dordrecht, 1996.

[SA86] H.D. Sherali and W.P. Adams. A tight linearization and an algorithm for 0-1 quadratic programming problems. *Management Science*, 32(10):1274–1290, 1986.