Combining QCR and CHR

for Convex Quadratic MINLP Problems

with Linear Constraints

Aykut Ahlatçıoğlu[1]
Michael Bussieck[2]
Mustafa Esen[3]
Monique Guignard[4,5]
Jan Jagla[2]
Alexander Meeraus[2]

Abstract

The convex hull relaxation (CHR) method (Albornoz 1998, Ahlatçıoğlu 2007, Ahlatçıoğlu and Guignard 2010) provides lower bounds and feasible solutions (thus upper bounds) on convex 0-1 nonlinear programming problems with linear constraints. In the quadratic case, these bounds may often be improved by a preprocessing step that adds to the quadratic objective function terms which are equal to 0 for all 0-1 feasible solutions yet increase its continuous minimum. Prior to (or in addition to) computing CHR bounds, one may use Plateau's quadratic convex reformulation (QCR) method (2006), or one of its weaker predecessors designed for unconstrained problems, the eigenvalue method of Hammer and Rubin (1970) or the method of Billionnet and Elloumi (2007). In this paper, we first describe the CHR method, and then present the essentials of the QCR reformulation methods. We present computational results for convex GQAP problems using CHR, preceded or not by Plateau's QCR.

Keywords: Nonlinear integer programming, bounds, relaxation, convex hull, CHR, QCR, SDP, GQAP.

[1] Princeton University; aahlatci@princeton.edu

[2] GAMS Dev. Corporation; mbussieck@gams.com, jhjagla@gams.com, ameeraus@gams.com,

[3] mustafa.esen2009@googlemail.com

[4] Corresponding author. University of Pennsylvania. Email address: guignard_monique@yahoo.fr,

[5] Research partially supported under NSF Grant DMI-0400155.

1. Introduction

In this paper we show how combining two recently developed techniques for mixed-integer nonlinear (resp. quadratic) optimization problems with linear constraints may substantially improve bounds over what each technique alone could produce.

The first technique, called Convex Hull Relaxation, or CHR for short, is a special case of the primal relaxation introduced in Guignard (1994) for nonlinear mixed-integer programming problems (see also Guignard 2003 and Guignard 2007b). This primal relaxation replaced part of the constraints by their integer convex hull. To be more specific, given the nonlinear integer problem Max $\{ f(x) \mid Ax \le b, Cx \le d, x \in S\}$, where S contains the integrality conditions over x, the primal relaxation with respect to the constraints $Cx \le d$ is the problem Max $\{ f(x) \mid Cx \le d, x \in Co\{y \in S \mid Ay \le b\}\}$. In the linear case, this is equivalent to the Lagrangean relaxation of the constraints $Cx \le d$ (see Geoffrion 1974). In the nonlinear case, in general it is not. Contesse and Guignard (1995) showed that this relaxation can be solved by an augmented Lagrangean method, and this was successfully implemented by S. Ahn in his Ph.D. dissertation (1997). Albornoz (1998) and later, independently, Ahlatçıoğlu (2007), thought of using this relaxation without separating any constraint, i.e., of defining the convex hull relaxation (CHR) of the problem Max $\{f(x) \mid Ax \le b, x \in S\}$ as Max $\{ f(x) \mid x \in Co\{y \in S \mid Ay \le b\}\}$. The advantage is that this relaxation can be solved very simply by using Frank and Wolfe's algorithm (1956) or, better, Von Hohenbalken's simplicial decomposition (1977), following an idea of Michelon and Maculan (1992) for solving directly the Lagrangean dual in the linear case, without resorting to Lagrangean multipliers.

A side benefit of this procedure is the generation of feasible integer points, which quite often provide a tight upper bound on the optimal value of the problem. Additionally, a feature added in the recent releases of CPLEX, the solution pool, gives access to (all if one wants) integer feasible points generated during any branch-and-bound run, and in the case of CHR, during the solution of the linearized 0-1 problems, one at each iteration of simplicial decomposition. This provides a larger pool of integer feasible solutions, and thus a higher probability of finding good solutions for the original quadratic problem, as one can sort these solutions according to a secondary criterion, in this case the quadratic objective function. In the nonconvex case, this has been used as a heuristic for generating good feasible solutions (Pessoa et al., 2010).

If one wants to improve on the CHR bound, one has to realize how the bound is computed. The CHR relaxation computes its bound over the convex hull of all integer feasible points, but probably exhibits only a small portion of these. One cannot therefore use these points to construct the actual convex hull of all integer feasible solutions. Yet, the bound is equal to the optimum of the original objective function (notice that in principle it does not have to be quadratic as far as CHR is concerned), and thus there is no point in trying to generate valid cuts. The only improvement can come from using objective function properties, potentially combined with properties of the constraints. This is where the second technique, which we will generically refer to as QCR (for Quadratic Convex Reformulation), comes in.

This technique's initial aim was the convexification of a nonconvex quadratic function in 0-1 variables, using properties of the 0-1 variables, and if appropriate, linear equality constraints of a 0-1 quadratic model. It was pioneered by Hammer and Rubin (1970). They were convexifying a nonconvex quadratic function $f(x) = x^TQx+c^Tx$ of the 0-1 variables x_j, j=1,...,n, by adding to it the null term $\sum_j u(x_j^2 - x_j)$, u real, so that the minimum over u of the modified f(x) for $x \in [0, 1]^n$ would be as large, i.e., as tight a bound as possible. They showed that taking u equal to the smallest eigenvalue, λ_{min}, i.e., replacing Q_{jj} for all j by $Q_{jj} + \lambda_{min}$, would achieve this. Billionnet and Elloumi (2008) then showed that one could improve on this scheme by adding a term $\sum_j u_j(x_j^2 - x_j)$, where the u_j's are the optimal dual variables of a certain semi-definite program. Finally, M.C. Plateau (2006) showed that, for a quadratic 0-1 programming problem with objective function f(x) and with linear constraints, some of them equalities of the form $\sum_j a_{ij}x_j = b_i$, the best lower bound provided by the convexification of f(x) coming from adding to it terms of the form $\sum_j u_j(x_j^2 - x_j)$ and $\sum_k \left(\sum_i \alpha_{ki}x_i\right) \left(\sum_j a_{kj}x_j - b_k\right)$, is obtained by using for u and α the dual variables of an enlarged SDP program. In her thesis, Plateau remarks that this convexification scheme can also be applied to functions that are already convex in order to tighten lower bounds. In the rest of the paper, we will refer to this scheme as "de-convexification", even though it keeps the functions convex, but in a sense, less so.

What we are proposing in this paper is to first de-convexify the quadratic objective function, using one of the techniques described above, and then to apply CHR to the resulting, already improved, model. We use the GAMS implementation of Plateau's extended QCR method available in the GAMS model library, following a joint project between GAMS, M.C. Plateau, and a research group at the University of Pennsylvania. This provides us with the best currently known lifting of the lower bound before using CHR. Using the weaker method of Hammer and Rubin or that of Elloumi and Billionnet before applying CHR already substantially improves the continuous lower bound as well as the CHR bound.

We first briefly describe in section 2 the CHR method, following Ahlatçıoğlu and Guignard (2007), and the QCR method in section 3, following Plateau (2006) and Guignard (2007-1). In section 4 we then present numerical results on the effect of combining both techniques, for convex quadratic assignment problems (GQAP).

2. The CHR algorithm

Consider the following nonlinear integer program (NLIP)

(NLIP) min $\{f(x) \mid x \in S\}$

where f(x) is a nonlinear convex function of $x \in R^n$, $S = \{x \in Y : Ax \leq b\}$, A is an $m \times n$ constraint matrix, b is a resource vector in R^m, Y is a subset of R^n specifying integrality restrictions on x.

Definition 2.1. ***We define the Convex Hull Relaxation of (NLIP) to be***

(CHR) min $\{f(x) \mid x \in Co(S)\}$.

Problem (CHR) is in general not equivalent to (NLIP) when f(x) is nonlinear, because an optimal solution of (CHR) may not be integer, and therefore not feasible for (NLIP). However, it is easy to see that (CHR) is indeed a relaxation to (NLIP).

This relaxation is a primal relaxation, in the x-space. It is actually a primal relaxation that does not "relax" any constraint. The difficulty in solving (CHR) comes from the implicit formulation of the convex hull. However the idea of decomposing the problem into a sub-problem and a master problem, first introduced by Frank & Wolfe (1956), and furthered by Von Hohenbalken with Simplicial Decomposition (1973), and Hearn et al. with Restricted Simplicial Decomposition (1987), provides an efficient way to solve (CHR) to optimality, by solving a sequence of linear integer problems and of essentially unconstrained nonlinear problems.

2.1. Applying simplicial decomposition to the CHR problem

2.1.1. Assumptions

In order for simplicial decomposition to guarantee a global optimal solution of (CHR), several conditions must be satisfied:

(i) the feasible region must be compact and convex,

(ii) the objective function must be convex, and

(iii) the constraints must be linear.

2.1.2. The Subproblem

The first part of the decomposition problem is the sub-problem, which can be viewed as a feasible descent direction finding problem. At the kth iteration of simplicial decomposition, given a feasible point xk of (CHR), one must find a feasible descent direction in the polyhedron Co {Ax = b, x ∈ X} by solving the linear programming problem

$$\text{(CHS)} \qquad \min_y \; \{\nabla f(xk)T \times (y - xk) \mid y \in \text{Co}\,\{Ax \leq b, x \in Y\}\}.$$

xk is called the linearization point.

The objective function of problem (CHS) being linear, its optimum over Co {Ax ≤ b, x ∈ Y} is reached at an extreme point of {Ax ≤ b, x ∈ Y}, i.e., at an integer point of {Ax ≤ b}, and problem (CHS) is equivalent to the Integer Programming Subproblem (IPS):

$$\text{(IPS)} \qquad \min_y \; \{\nabla f(xk) \times (y - xk) \mid Ay \leq b,\, y \in Y\}$$

in which one has removed the mention of the convex hull.

Problem (IPS) having a linear objective function is usually much easier to solve than problem (NLIP). The solution to (IPS) will be a new extreme point of the convex hull of the set of feasible integer solutions to (NLIP), unless x^k is already optimal for the convex hull relaxation (CHR) problem. Thus, at each nonterminal iteration we obtain an integer feasible point to the original (NLIP) problem. Convergence to the optimal solution is discussed in section 2.2. If x^k is not optimal, we proceed to the master problem.

2.1.3. The Master Problem

The master problem consists in optimizing the original nonlinear objective function over the convex hull of the points generated by the subproblems. One can express it in terms of nonnegative weights β_i, i=1,…,r where r is the current iteration number, adding up to 1, of the form

$$(MP) \quad \text{Min } f(X\beta) \text{ s.t. } \sum_{Ii}\beta_i = 1, \beta_i \geq 0, i = 1, 2, ..., r.$$

X is the n × r matrix comprised of a subset of extreme points yk of the convex hull, generated by the subproblems, along with one of the linearization points xk. There are r such points (or column vectors) in X. Then at the master problem stage, (MP) is solved, which is a minimization problem over an r−1 dimensional simplex.

If the optimal solution of (CHR) is within this simplex, then the algorithm terminates. If not, the optimal solution β* of (MP) will be used to compute the next iterate, xk+1, which can be found using the formula:

$$xk+1 = \sum i \ \beta^* i \times Xi .$$

Then we go back to the subproblem, find another extreme point and increase the dimension of the simplex for (MP).

For some pathological cases, putting no restriction on r could potentially pose computational problems. Restricted simplicial decomposition, introduced by Hearn et al. (1987) puts a restriction on the number of extreme points that can be kept.

2.2. Convergence to the Optimal Solution of CHR

Because the objective function is convex, the necessary and sufficient optimality condition for x^k to be the global minimum is

$$\nabla f(x^k)^T (y^* - x^k) \geq 0$$

Lemma 2 of Hearn et al. (1987) proves that if x^k is not optimal, then $f(x^{k+1}) < f(x^k)$, so that the sequence is monotonically decreasing. Finally Lemma 3 of Hearn et al. (1987) shows that any convergent subsequence of x^k will converge to the global minimum. The algorithm used in this study follows the restricted simplicial decomposition (Hearn et al. 1987).

2.3. Calculating lower and upper bounds for convex GQAP problems

As stated in Definition 2.1, (CHR) is a relaxation to the (NLIP). Simplicial Decomposition finds an optimal solution, say, x*, to (CHR), and this provides a lower bound on v(NLIP):

$$LB_{CHR} = f(x^*)$$

On the other hand, at each iteration k of the subproblem an extreme point, y^{*k}, of the convex hull is found, which is an integer feasible point of (NLIP). Each point y^{*k} yields an Upper Bound (UB) on the optimal value of (NLIP), and the best upper bound on v(NLIP) can be computed as

$$UB_{CHR} = \min f(y^{*1}), f(y^{*2}),\ldots, f(y^{*k})\}.$$

To demonstrate the ability of the CHR approach to compute bounds often significantly better than the continuous relaxation bounds, we implemented CHR to find a lower bound on the optimal value of convex GQAPs.

We used two types of data. First we adapted data for GQAPs from the literature, with problems of size 30x15, 50x10, etc., and measured the improvement over the continuous bound by computing how much the gap is reduced when one replaces the NLP bound by the CHR bound. We computed the matrix of the objective function by premultiplying the original objective function matrix, whose entries are products of a flow by a distance as given in the original GQAP instances, by its transpose. These problems tend to have moderate integrality gaps. The improvement was in the range 43 to 99 %. The largest runtime on a fast workstation was 12 seconds.

The second data set uses again data from the linear GQAP literature, and generates the objective function matrix as the product of a matrix by its transpose, but this matrix is now randomly generated with coefficients between -a and +b, for various combinations of a and b. These tend to have large duality gaps and to be considerably more difficult to solve. We will describe in the next paragraph how one can partially reduce this gap.

3. A priori Improvement of Lower Bounds using QCR by de-convexification

For some of the convex data sets generated for GQAP, the gaps between the continuous and/or the CHR bound on the one hand, and the optimal value on the other, are very large. Since CHR computes a bound based on the convex hull of all 0-1 feasible solutions, there is nothing that can be done to improve that part of the model, like adding cuts or tightening inequalities. The only way to improve the lower bound is to use the objective function.

The reason we mention the continuous relaxation bound is that we know that the CHR bound must be at least as good or better for convex problems. If we can manage to increase the continuous bound, we might be able to increase the CHR bound as well.

Consider the convex function $f(x) = u\, x\, (x-1)$, $x \in \{0, 1\}$, u a positive scalar. The problem is to minimize f(x) subject to $x \in \{0, 1\}$. The function is zero for x = 0 or 1, but it is negative in between. If one computes the continuous bound on f(x) for $x \in [0, 1]$, one gets $u(1/2)(-1/2) = -u/4$, and if u is large, so is the integrality gap. Notice however that if we replace f(x) by $g(x) = e.x.(x - 1)$ with $e > 0$, e very close to 0, it would produce an equivalent problem $\min \{g(x) \mid x \in \{0, 1\}\}$. Indeed g(x) coincides with f(x) over the feasible set {0, 1}, yet it yields a much better continuous lower bound, equal to -e/4, and it is also a convex function as long as e is positive, no matter how small.

If one has a convex objective function of n variables, the same behavior may occur, i.e., the continuous bound, and thus the integrality gap, may be very large because the value of the objective function drops substantially when the variables are allowed to be between 0 and 1.

Convexification in its simplest form (Hammer and Rubin, 1970) adds terms of the form $u(x_{ij}^2 - x_{ij})$, with u real, to the quadratic objective function. To convexify a nonconvex quadratic function, one tends to add positive terms to the diagonal of the matrix, to make it positive semidefinite. Here as we start from already convex objective functions, we will subtract positive terms from the diagonal, as long as the objective function remains convex. This will not change the objective function value for x_{ij} 0 or 1. We will call this backward process diagonal de-convexification, even though it leaves the problem convex. Subtracting the smallest eigenvalue of the matrix from every diagonal element improves the continuous bound as well as the CHR bound for convex GQAPs. More sophisticated de-convexification methods use semidefinite programming (see for instance Billionnet, Elloumi and Plateau (2008), as well as the GAMS website link http://www.gams.com/modlib/libhtml/gqapsdp.htm showing an application to GQAP using CSDP) but are clearly more expensive to setup. We will now describe them.

3.1. Convex problems.

Most MINLP algorithms, no matter what technology they are using, are eventually relying on Branch-and-Bound to guarantee an optimal solution. This guarantee is based on the fact that no integer feasible solution can produce a value better than the final bound on the optimum, and a bound on the optimum is normally based on a relaxation of the MINLP, which produces a valid bound only if the function to be minimized is convex. It is thus important to make sure that the objective function of an MINLP with linear constraints is convex, and convexify it if it is not. If the objective function is convex, one may want to modify it to obtain a larger continuous lower bound, but so that it still coincides with the original one for all 0-1 feasible solutions. In any case, we will be searching for a substitute convex objective function that produces the best possible continuous lower bound.

3.2. Convexification methods.

We will now review in detail convexification methods for *quadratic* problems with linear constraints. Let us consider problem

(MIQCP) Min $f(x) = x^TQx+c^Tx$ s.t $Ax = b$, $Cx \le d$, $x_j \in \{0,1\}$, $j=1,...,n$.

3.2.1. Hammer and Rubin's eigenvalue method

Hammer and Rubin (1970) proposed a uniform diagonal modification of the matrix Q, that is, to add the same amount to each diagonal element of the matrix, specifically, a null term $\sum_j u(x_j^2 - x_j)$, u real. They show that the best value for u, i.e., that which produces the tightest lower bound possible on $v(\overline{MIQCP})$, is obtained by taking u equal to the smallest eigenvalue, λ_{min}, of Q. In other words, they replace Q_{jj} for all j by $Q_{jj} + \lambda_{min}$. The method requires the computation of the smallest eigenvalue of matrix Q. In our initial experiments, we used the fact that the GAMS program EIGVAL01, available in the GAMS Test Library, computes efficiently all eigenvalues of a square symmetric matrix. We embedded EIGVAL01 in a GAMS program that computed the smallest eigenvalue, in order to improve (convexify or (de)-convexify) a given quadratic function. Numerically speaking, though, the modification of the matrix creates a problem on the boundary between convex and nonconvex problems, and we found it necessary to very slightly perturb the diagonal to make sure one stays on the convex side.

3.2.2. Billionnet, Elloumi and Plateau

Plateau (2006) and Billionnet, Elloumi and MC Plateau (2007) generalized the method further by allowing modification of the entire matrix Q and solving an SDP problem that gets the best resulting continuous bound. Their method can be used for nonconvex problems, for convexifying the objective function to obtain an equivalent convex quadratic problem with the tightest possible continuous bound (see Figure 1), as well as for convex problems, for "de-convexifying" the objective function to obtain an equivalent convex quadratic problem with the tightest possible continuous bound (see Figure 2).

Given a 0-1 programming problem (Q0-1) with a quadratic objective function q(x) and linear equality and possibly inequality constraints, of the form

(Q01) Min $q(x) = x^T Qx + c^T x$

s.t. $Ax = b$

$A' x \leq b'$

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

with A of full rank, one tries to replace the original convex quadratic objective function by a convex quadratic function $q_{\alpha,u,v}(x)$, where u is a vector, α a matrix and v a scalar, such that the optimal solution and optimal value remain unchanged, and such that the new continuous relaxation bound is as large as possible. [6]

The equivalent model has an objective function of the form

$$q_{\alpha,u,v}(x) = x^T Qx + c^T x + \Sigma_j \; u_j(x_j^2 - x_j) + \Sigma_{i,j} \alpha_{ij} x_j (A_i x - b_i) + v \; \| Ax-b \|^2$$

$= \quad x^T Qx + c^T x + \Sigma_j u_j(x_j^2 - x_j) + \Sigma_{i,j} \alpha_{ij} x_j (A_i x - b_i) + v (x^T A^T A x - 2 b^T Ax + b^T b)$

$= \quad x^T Q_{\alpha,u,v}\, x + c_{\alpha,u,v}{}^T x + b^T b,$

where $Q_{\alpha,u,v}$ is an nxn matrix like Q, and $c_{\alpha,u,v}$ is an n-vector like c. Clearly, $q_{\alpha,u,v}(x)$ is equal to q(x) for every feasible solution of (Q01). The equivalent, convex, model, is

$(Q01_{\alpha,u,v})$ Min $q_{\alpha,u,v}(x)$

$$\equiv x^T Qx + c^T x + \Sigma_j u_j(x_j^2 - x_j) + \Sigma_{ij} \alpha_{ij} x_j (A_i x - b_i) + v \| Ax - b \|^2$$

$$\text{s.t.} \quad Ax = b, \; A' x \le b', \; x \in \{0,1\}^n.$$

It can be rewritten equivalently, after defining the matrix X by $X_{ij}=x_i x_j$, all i, j:

$(Q01_{\alpha,u,v})$ Min $\{q_{\alpha,u,v}(x) \mid Ax = b,\; A' x \le b',\; X_{ij}=x_i x_j,\; x\in\{0,1\}^n\}$

$= \text{Min} \quad \{q_{\alpha,u,v}(x) \mid Ax = b,\; \sum_j A_{ij} X_{ij} = b_i x_j,\; \Sigma_{i,j,k} A_{ki} A_{kj} X_{ij} - 2b^T Ax + b^T b = 0,\; A' x \le b',$

$\quad X_{ij}=x_i x_j, x\in\{0,1\}^n,\; X\in S^n\},$

where S_n denotes the set of symmetric nxn matrices.

The semidefinite relaxation of this last formulation of $(Q01_{\alpha,u,v})$ consists in replacing the set of constraints $X_{ij} = x_i x_j$, i.e., $X - x x^t = 0$, with the linear matrix inequality $X - x x^t \succeq 0$. By Schur's Lemma, $X - x x^t \succeq 0$ is equivalent to (and thus can be replaced by)

$$\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0 .$$

So finally, $(Q01_{\alpha,u,v})$ is relaxed by

- keeping only X_{ii} (= $x_i x_i$) = x_i for all i, since x_i is a 0-1 variable

---

[6] Billionnet and Elloumi (2008) considered unconstrained models, and thus implicitly had α=v=0.

• and replacing $X - x\,x^t = 0$ by $\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0$,

and this yields the following problem C(Q01):

$$\text{Min} \quad \{q_{\alpha,u,v}(x) \mid Ax = b, \quad \textstyle\sum_j A_{ij} X_{ij} = b_i\, x_j,$$

$$\Sigma_{i,j,k} A_{ki}\, A_{kj}\, X_{ij} - 2b^T Ax + b^T b = 0,\ A'\,x \le b',\ \ X_{ii} = x_i,\ \ \begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0,\ x \in \{0,1\}^n\}.$$

Let $X = \{x \in \{0,1\}^n \mid Ax = b,\ A'\,x \le b'\}$ and let $C(X) = \{x \in [0,1]^n \mid Ax = b,\ A'\,x \le b'\}$. The relaxation of problem (Q01), or equivalently, ($Q01_{\alpha,u,v}$), is $R_{\alpha,u,v}$ (Q01): Min $_{x \in C(X)}$ $\{q_{\alpha,u,v}(x)\}$, whose optimal value depends on α, u and v. The ultimate goal is to determine values for $v \ge 0$, α and u that maximize the optimal value of $R_{\alpha,u,v}$ (Q01) with $q_{\alpha,u,v}$ convex, i.e., with $Q_{\alpha,u,v}$ positive semi-definite. To put it differently, one wants to solve the problem

C(Q01) $\qquad \text{Max}_{\,v \ge 0,\ \alpha, u,\ Q_{a,u,v} \succeq 0}\ \text{Min}_{\,x \in C(X)}\ \{q_{\alpha,u,v}(x)\}.$

The following theorem (Plateau 2006)[7] identifies the optimal values of α, u and v.

<u>THEOREM 1</u> ***The optimal value of***

C(Q01) $\qquad \textbf{Max}_{\,v \ge 0,\ \alpha, u}\quad \textbf{Min}_{\,x \in C(X)}\ \{q_{\alpha,u,v}(x)\}$

is equal to the optimal value of the following semidefinite programming problem

(SDQ01) $\qquad \textbf{Min} \quad cx + \Sigma_i \Sigma_{j \ne i}\, Q_{ij}\, X_{ij}$

$\qquad \textbf{s.t.} \quad \Sigma_i\, A_{ki}\, X_{ij} - b_k x_j = 0, \quad \forall k,j \qquad (1)$

$\qquad X_{ii} = x_i \qquad \forall i \qquad (2)$

[7] Plateau (2006) did not mention v. Guignard (2007-1) introduced v and suggested that it would possibly be more interesting computationally than α. Billionnet, Elloumi and Plateau (2009) compare the (α,u,0) and (0,u,v) options and state that (α,u,0) and (0,u,v) yield the same bound, since the associated semidefinite programs have the same optimal value. We chose to present the approach for (α,u,v) for sake of completeness. For our experiments, (0,u,v) proved superior to (α,u,0) as SDP problems could be solved to optimality for larger problems. We do not use (α,u,v) in our experiments.

$$\Sigma_{i,j,k} A_{ki}\, A_{kj}\, X_{ij} - 2\, b^T A\, x + b^T b = 0 \quad (3)$$

$$Ax = b \quad (4)$$

$$A'x \le b' \quad (5)$$

$$\begin{bmatrix} 1 & x^T \\ x & X \end{bmatrix} \succeq 0 \quad (4)$$

$$x \in R^n,\ X \in S_n$$

where S_n denotes the set of symmetric nxn matrices.

<u>THEOREM 2</u> The optimal values α^*, u^* and v^* of α, u and v are given by the optimal dual variables associated with constraints (1), (2) and (3) respectively:

$\Sigma_i A_{ki} X_{ij} - b_k x_j = 0,$	$\forall k,j$	(1)	α_{kj}
$X_{ii} = x_i$	$\forall i$	(2)	u_i
$\Sigma_{i,j,k} A_{ki} A_{kj} X_{ij} - 2\, b^T A\, x + b^T b = 0$		(3)	v

In addition, $v(SDQ01) = v(\overline{Q01_{\alpha,u,v}})$ where $(\overline{Q01_{\alpha,u,v}})$ is the continuous relaxation of problem ($Q01_{\alpha,u,v}$).

Obtaining a QCR bound therefore requires the solution of an SDP problem that produces the best α^*, u^* and v^*, and whose optimal value is the desired bound. If one wants to solve problem (Q01) to optimality, and it is not convex, one cannot directly use BB, but one can instead solve the convex problem ($Q01_{\alpha,u,v}$) by BB, and its continuous relaxation $(\overline{Q01_{\alpha,u,v}})$ yields the same value as (SDQ01). If (Q01) is convex, one can try to solve it directly, but ($Q01_{\alpha,u,v}$) is in general a better choice, with a tighter lower bound at the top of the BB tree. In the convex case, we say, by slightly abusing the language, that we have “de-convexified” (Q01), in fact we have made its objective function less steep, while keeping it convex. Plateau (2006) already mentions this possibility (p. 173).

4. Application: GQAP (generalized quadratic assignment problem)

To illustrate the improvement in bounds over the continuous relaxation provided by CHR alone, and by QCR followed by CHR, we chose to apply these methods to the generalized quadratic assignment problem (GQAP). An earlier joint project between the authors, with the collaboration of M.-C. Plateau, had resulted in a GAMS prototype of the QCR method for the GQAP (see in the GAMS Model Library, model #339).

The GQAP model assigns tasks i to agents j, while minimizing quadratic costs. It can be written

(GQAP) Min $f(x) = x^T C x + h^T. x$

s.t. $\sum_i x_{ij} = 1 \quad \forall i \quad$ (MC)

$\sum_i a_{ij} x_{ij} \le b_j \quad \forall j \quad$ (KP)

$x_{ij} \in \{0, 1\}, \quad \forall i,j \quad$ (0-1)

where $f(x) = x^T C x + h^T. x$ is a quadratic objective function, the (MC) constraints are multiple choice constraints for assigning each task i to exactly one agent j, the (KP) constraints are knapsack-type capacity constraints for each agent j, and the (0-1) constraints are the binary restrictions on the assignment variables x_{ij} .

4.1. Large instances with small integrality gaps

Table 1 shows results for relatively large GQAP instances with smaller integrality gaps. The last two digits of an instance name are for identification, while the first four specify problem size, for instance 30x20 or 40x10. Taking standard datasets from the literature, one just modifies the (a priori nonconvex) objective function by premultiplying its objective function matrix C by its transpose.

-- Insert Table 1 about here --

The Best IFV is the best integer feasible value found by CHR, the CB and CHR % gaps represent the residual integrality gaps for the continuous bound (CB) and CHR bound respectively. Notice that the continuous gap is computed relative to the best IFV found by CHR. The last column lists the elapsed time in seconds on a Thinkpad X61 dual core laptop, and the number of simplicial decomposition iterations for CHR.

As Table 1 shows, the CHR bound almost bridges the gap in most cases. For these problems, we did not try de-convexification, because the CHR residual gaps were already very small.

4.2. Instances with large integrality gaps.

We will now use generalized quadratic assignment instances with a large integrality gap. We present results for a number of instances. These instances are large and difficult, even for a specialized BB algorithm. We could verify that for the first two problems, we obtained optimal solutions. For very few medium size problems, BB eventually generated better feasible solutions than we did, however only a handful of problems could be solved to optimality. Others problems attempted had to be aborted because of lack of convergence.

The objective function matrix C is obtained by randomly generating a 600x600 matrix with entries between –a and +b, for instance -15 and +20, multiplying this matrix by its transpose and taking the necessary number of rows and columns to obtain a matrix that is positive semidefinite. As in 4.1, the rest of the data comes from standard datasets from the literature.

-- Insert Table 2 about here –

Table 2 consists of two parts, the left part concerns CHR without prior de-convexification (DCV), the right one with de-convexification. Elapsed times are reported in seconds, but only for CHR. The runs were made using GAMS 23.5.2 for Windows 32bit, on an Intel(R) Core(TM) i7 CPU 860 with 8 GB of RAM. When CHR times were larger than 100 seconds, we also reported the time taken by CPLEX 12.1 to solve the continuous relaxation of the problem (rMIP). One problem took an enormous amount of time for CHR plus de-convexification, it so happens that some linear MIP subproblems were especially difficult for CPLEX. In a normal run, of course, one would interrupt the run, forgetting about obtaining an improved CHR bound, but keeping the integer feasible solutions already obtained.

The bold numbers show which of CHR, or CHR plus de-convexification, yields the best IFV. Given that CHR alone costs very little, it is probably best to run both cases and keep the best IFV of both runs. The gaps are computed separately, using their own best IFV, but the last columns use the best overall IFV to compute the gap improvement.

For these problems, QCR yielded the larger part of the improvement, but it does not provide feasible solutions. One advantage of CHR is that it provides both a bound and good feasible values.

5. Conclusion

The Convex Hull Relaxation (CHR), possibly combined with de-convexification from QCR as a preprocessing step, provides tight lower and upper bounds by (1) transforming a nonlinear integer optimization problem in one over the convex hull of all integer feasible solutions, and (2) replacing this problem by a sequence of integer linear programs and simple nonlinear continuous programs over a simplex. The potential strength of the proposed algorithm is that the difficulty of the problems solved at each iteration usually stays relatively unchanged from iteration to iteration. It will be most suitable for those nonlinear integer problem types that would be much easier to solve with a linear objective function. One should expect that CHR will have a robust performance for large-scale problems if one has access to solvers able to handle large integer linear programs and simple nonlinear programs efficiently. Current experiments seem to confirm this behavior. Preceding the CHR bound computation by de-convexification of the quadratic objective function to a point where the function is still convex but has achieved its highest possible continuous bound is a very natural complement to CHR, as it concentrates on the objective function and possibly the linear equality constraints of the problem. There is indeed no obvious further improvement possible on the CHR bound coming solely from the constraint set, because this constraint set has already been maximally reduced to produce a formulation equivalent to the minimization over the convex hull of integer feasible solutions. The positive results on CHR coupled with the QCR de-convexification are very encouraging, and the initial experiments for the GQAP using Plateau's QCR method show a large improvement of the CHR bounds for problems with large gaps.

Future research will consider other models than the GQAP, including general convex MINLP models from the literature. We will also investigate whether solving CHR, QCR and again CHR in a single algorithm would improve the overall behavior, using the integer points generated during the initial CHR phase as a hot start for the second CHR call. Indeed CHR takes much longer after de-convexification.

One last comment is that QCR provides a convex function with the best continuous bound, but not necessarily the best CHR bound. We have tested replacing (u,v) by α(u,v), α a scalar between 0 and 1, and found that in most cases, α less than 1 provided a better CHR bound. But at this point it is not clear how to determine the best α beforehand.

Acknowledgement.

We would like to express our gratitude to Marie-Christine Plateau for all the help she provided us as we were implementing her QCR method for the GQAP in GAMS.

References

Ahn, S., (1997), Ph.D. dissertation, University of Pennsylvania.

Ahlatçıoğlu, A., (2007) summer paper, OPIM Dept., Univ. of Pennsylvania. Now subsumed by Ahlatçıoğlu and Guignard (2007).

Ahlatçıoğlu, A. and M. Guignard, (2007), "The convex hull relaxation for nonlinear integer programs with linear constraints," OPIM Department Report (revised, 2008).

Ahlatçıoğlu A. and M. Guignard, (2010), "The convex hull relaxation (CHR) for convex and nonconvex MINLP problems with linear constraints," OPIM Dept. Report, Univ. of Pennsylvania, submitted for publication.

Ahn, S., L. Contesse and M. Guignard, (1995) "An Augmented Lagrangean Relaxation for Nonlinear Integer Programming Solved by the Method of Multipliers, Part II: Application to Nonlinear Facility Location," Working Paper, latest revision 2007.

Albornoz, V., (1998), "Diseño de Modelos y Algoritmos de Optimización Robusta y su Aplicación a la Planificación Agregada de la Producción." Doctoral Dissertation, Universidad Catolica de Chile, Santiago, Chile.

Billionnet, A. and S. Elloumi, (2001), "Best reduction of the quadratic semi-assignment problem," *Discrete Applied Mathematics*, Vol. 109, 197-213.

Billionnet, A. and S. Elloumi, (2007), “Using a Mixed Integer Quadratic Programming Solver for the Unconstrained Quadratic 0-1 Problem,” Math. Program., Ser.A109, 55–68.

Billionnet A., S. Elloumi and M.-C. Plateau, (2008), “Quadratic 0-1 Programming: Tightening Linear or Quadratic Convex Reformulation by Use of Relaxations,” *RAIRO-Oper. Res.* 42, 103-121.

Billionnet A., S. Elloumi, M.-C. Plateau, (2009), “Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method,” *Discrete Applied Mathematics* 157, 1185–1197.

Contesse L. and M. Guignard, (1995), “An Augmented Lagrangean Relaxation for Nonlinear Integer Programming Solved by the Method of Multipliers, Part I, Theory and Algorithms” Working Paper, OPIM Department, University of Pennsylvania, latest revision 2009.

Frank, M. and P. Wolfe, (1956) , “An algorithm for quadratic programming,” *Naval Research Quarterly*, 3(1,2), 95-109.

Geoffrion, A.M., (1974), "Lagrangean Relaxation for Integer Programming," *Mathematical Programming Study*, **2**, 82-114.

Guignard, M., (1994), “Primal Relaxation in Integer Programming,” VII CLAIO Meeting, Santiago, Chile, 1994, also Operations and Information Management Working Paper 94-02-01, University of Pennsylvania.

Guignard, M. , (2003), “Lagrangean Relaxation,” TOP, 11(2), 151-228.

Guignard, M., (2007), "Extension to the Plateau Convexification Method for Nonconvex Quadratic 0-1 Programs," OPIM Department Research Paper 07-09-21, University of Pennsylvania.

Guignard, M., (2007), "A New, Solvable, Primal Relaxation For Nonlinear Integer Programming Problems with Linear Constraints," *Optimization Online*, http://www.optimization-online.org/DB_HTML/2007/10/1816.html, also Operations and Information Management Working Paper, University of Pennsylvania.

Guignard, M. and A. Ahlatçıoğlu, (2010), "The Convex Hull Relaxation for Nonlinear Integer Programs with Convex Objective and Linear Constraints," European Workshop in Mixed-Integer Nonlinear Programming, Marseille, France, May 2010.

Guignard, M., A. Ahlatçıoğlu, M. Bussieck, M. Esen and A. Meeraus, (2010), "De-convexification tightens CHR bounds on convex 0-1 quadratic programming problems," EURO Lisbon, July 2010.

Hammer, P.L. and A.A. Rubin, (1970), "Some remarks on quadratic programming with 0-1 variables," *RAIRO* Vol. 3, 67-79.

Hearn, D.W., S. Lawphongpanich and J.A. Ventura, (1987), "Restricted Simplicial Decomposition: Computation and Extensions," *Mathematical Programming Study* 31, 99-118.

Michelon, P. and N. Maculan, (1992), "Solving the Lagrangean dual problem in Integer Programming," Report #822, Departement d'Informatique et de Recherche Operationnelle, University of Montreal.

Pessoa, Artur Alves, Peter M. Hahn, Monique Guignard and Yi-Rong Zhu, (2010), "Algorithms for the generalized quadratic assignment problem combining

Lagrangean decomposition and the Reformulation-Linearization Technique," *European Journal of Operational Research* 206, 54–63.

Plateau, M.C., (2006), "Reformulations quadratiques convexes pour la programmation quadratique en variables 0-1," Doctoral Dissertation, Laboratoire Cedric, CNAM, France.

Von Hohenbalken, B., (1977), "Simplicial decomposition in nonlinear programming algorithms," *Mathematical Programming*, 13, 49-68.

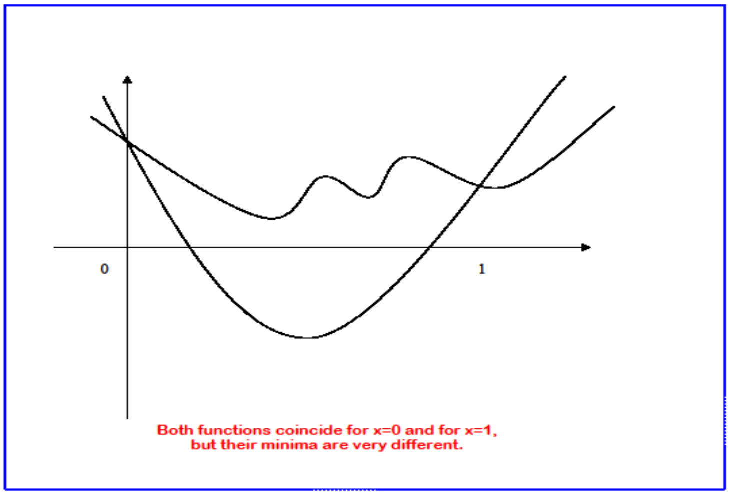
0
1
Both functions coincide for x=0 and for x=1,
but their minima are very different.

Figure 1

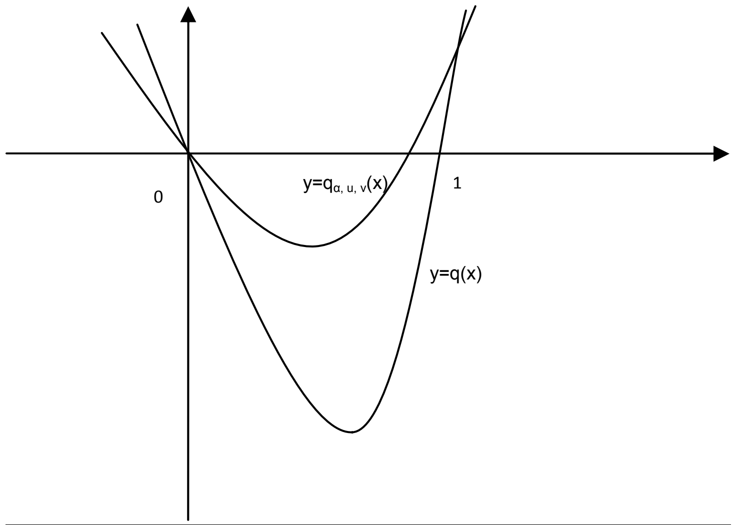

Figure 2

Instance	Continuous bound	CHR bound	Best IFV	CB % gap	CHR % gap	% gap improvement	CPU sec./ # iter. Thinkpad X61 dual core
30-20-35	28,996,191	29,210,393	29,267,194	0.93	0.19	79.04%	4.45/24
30-20-55	18,721,636	18,925,414	18,932,878	1.12	0.04	96.47%	1.08/7
30-20-75	24,882,925	25,273,194	25,299,161	1.65	0.10	93.76%	5.01/12
30-20-95	21,162,833	23,787,852	23,809,571	11.12	0.09	99.18%	6.54/4
35-15-35	32,750,712	32,769,891	32,795,216	0.13	0.07	43.09%	2.10/17
35-15-55	27,443,328	27,605,620	27,621,169	0.64	0.06	91.26%	3.59/15
35-15-75	30,638,516	30,920,476	30,928,923	0.94	0.03	97.09%	2.31/6
35-15-95	34,722,239	35,825,436	35,886,295	3.24	0.17	94.77%	24/6/6
50-10-75	56,103,056	56,606,460	56,615,187	0.90	0.002	98.30%	23.96/6
50-10-95	71,684,990	72,082,091	72,099,812	0.58	0.02	95.73%	6.02/5
16-07-10	2,681,377	2,807,912	2,823,041	5.02	0.54	89.32%	0.81/6
20-10-10	7,856,187	7,890,835	7,895,189	0.49	0.06	88.84%	0.58/6
27-21-10	15,502,600	15,626,812	15,650,868	0.95	0.15	83.78%	3.01/16

Table 1

Dataset (type)	# of 0-1 vars.	Cont. Bd	rMIP time sec	CHR bd w/o DCV	Best IFV	Elapsed CHR time sec	Int. gap	DCV Bd	CHR Bd after DCV	Best IFV	Elapsed CHR time sec	Int. gap	best IFV overall	% imp
c16x7_ldg_15_20	112	3068411		3234964	**4328390**	42	25.26%	3761048	3849327	**4328390**	32	11.07%	4328390	56%
c20x10_ldg_15_20	200	4183813		4191066	5295100	2	20.85%	4825174	4829943	**5208615**	36	7.27%	5208615	63%
c27x21_ldg_20_15	567	9890615	134	9910219	11416733	13	13.20%	10393957	10406468	**11366793**	110	8.45%	11366793	34%
c27x21_ldg_15_20	567	6493746		6990670	8205946	0	14.81%	6985514	6996779	**8191065**	92	14.58%	8191065	1%
c-20-15-35_ldg_15_20	300	3934019		3934243	5219490	7	24.62%	4489434	4490577	**5202924**	33	13.69%	5202924	44%
c-20-15-35_ldg_20_15	300	5932552		5940992	7237544	6	17.91%	6463595	6468719	**7149261**	37	9.52%	7149261	44%
c-20-15-55_ldg_15_20	300	3943191		3996137	5499388	11	27.33%	4499050	4520329	**5455645**	40	17.14%	5455645	36%
c-20-15-55_ldg_20_15	300	5941673		5978807	7399266	3	19.20%	6484369	6504517	**7365978**	26	11.70%	7365978	38%
c-20-15-75_ldg_15_20	300	3969636		4167590	**5618908**	12	25.83%	4529767	4611270	5717248	27	19.34%	5618908	31%
c-20-15-75_ldg_20_15	300	5969412		6205973	**7698854**	3	19.39%	6525202	6616681	7784209	38	15.00%	7698854	28%
c-30-06-95_ldg_15_20	180	9912936		9926556	11254676	4	11.80%	10750566	10760427	**11174073**	18	3.70%	11174073	67%
c-30-06-95_ldg_20_15	180	14523694	51	14560611	15843402	2	8.10%	15359508	15377402	**15790026**	144	2.61%	15790026	66%
c-30-07-75_ldg_15_20	210	9459033	50	9474383	10916198	15	13.21%	10295662	10309964	**10729717**	116	3.91%	10729717	67%
c-30-07-75_ldg_20_15	210	14483856		14528676	16017068	1	9.29%	15310519	15322282	**15794032**	29	2.99%	15794032	63%
c-30-08-55_ldg_15_20	240	9174920		9188671	**10480453**	4	12.33%	9941638	9947587	10526129	52	5.50%	10480453	59%
c-30-08-55_ldg_20_15	240	13860307		13881814	**15075317**	5	7.92%	14621282	14628894	**15075317**	18	2.96%	15075317	63%
c-30-10-65_ldg_15_20	300	8952179	56	8954776	**10353966**	11	13.51%	9681441	9686018	10408567	107	6.94%	10353966	52%
c-30-10-65_ldg_20_15	300	13399656		13453725	**14846665**	1	9.38%	14140789	14154828	**14846665**	85	4.66%	14846665	50%
c-30-20-35_ldg_15_20	600	7932361	51	8333435	**9264122**	0	10.05%	8429977	8432704	9269782	114	9.03%	9264122	11%
c-30-20-35_ldg_20_15	600	11980414		11999672	13446607	5	10.76%	12494333	12505441	**13388875**	84	6.60%	13388875	36%
c-40-07-75_ldg_15_20	280	16377748	51	16423144	18041204	5	8.97%	17240620	17258648	**17879798**	110	3.47%	17879798	57%
c-40-09-95_ldg_10_20	360	77027537	60	77257318	**78105845**	8	1.09%	77526986	77628607	78677774	237	1.33%	78105845	44%

c-40-09-95_ldg_15_20	360	15721361	65	15804077	**17587356**	7	10.14%	16514130	16538547	17687599	9966 (89 it., 9822 MIP, 144 NLP)	6.50%	17587356	41%
c-40-10-65_ldg_10_20	400	75991473		76060242	**76761733**	2	0.91%	76449137	76511560	76776037	27	0.34%	76761733	64%
c-40-10-65_ldg_15_20	400	15191883		15199578	**16515897**	14	7.97%	15925282	15940622	16586420	78	3.89%	16515897	56%
c-50-10-65_ldg_15_20	500	22985184		22988259	**24545547**	6	6.34%	23786116	23792874	24551041	68	3.09%	24545547	52%

Table 2