CONVEX HULL RELAXATION (CHR)
FOR CONVEX AND NONCONVEX MINLP PROBLEMS
WITH LINEAR CONSTRAINTS

by

Aykut Ahlatçıoğlu [1]

and

Monique Guignard[2,3]

Abstract.

The behavior of enumeration-based programs for solving MINLPs depends at least in part on the quality of the bounds on the optimal value and of the feasible solutions found. We consider MINLP problems with linear constraints. The convex hull relaxation (CHR) is a special case of the primal relaxation (Guignard 1994, 2007) that is very simple to implement. In the convex case, it provides a bound stronger than the continuous relaxation bound, and in all cases it provides good, and often optimal, feasible solutions. We present computational results for QAP, GQAP, MSAP and CDAP instances.

Keywords

mixed integer nonlinear programming; primal relaxation; feasible solution; convex hull; bound

[1] Department of Economics, Princeton University. aahlatci at princeton.edu
[2] Department of OPIM, the Wharton School, University of Pennsylvania.
[3] Corresponding author: guignard_monique at yahoo.fr and guignard at wharton.upenn.edu.

1. Introduction

The Convex Hull Relaxation (CHR) is a special case of the primal relaxation for nonlinear mixed-integer programming problems introduced in Guignard (1994). The primal relaxation replaces part of the constraints by their integer convex hull. To be more specific, the primal relaxation is defined as follows:

Definition 1.1. Given the nonlinear integer programming problem

(NIP) $\quad \mathrm{Max}_x \; \{ f(x) \mid Ax \le b,\ Cx \le d,\ x \in S\},$

where S contains the integrality conditions over x, the primal relaxation with respect to the constraints Cx ≤ d is defined as the problem

(PR) $\quad \mathrm{Max}_x \; \{ f(x) \mid Cx \le d,\ x \in \mathrm{Co}\{y \in S \mid Ay \le b\}\}.$

In the linear case, this is equivalent to Lagrangean relaxation (Held and Karp 1970, 1971, 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 $\mathrm{Max}\,\{f(x) \mid Ax \le b,\ x \in S\}$ as $\mathrm{Max}\,\{ f(x) \mid x \in \mathrm{Co}\{y \in S \mid Ay \le b\}\}$. This relaxation is of course the tightest primal relaxation one can think of, since it does not separate any set of constraints. The computational advantage of CHR is that it 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 CHR is the generation of feasible integer points at no extra cost. These 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 chooses to) 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, CHR has been used as a heuristic for generating good feasible solutions for QAP, GQAP (Pessoa et al., 2010), MSAP and CDAP.

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, actually a small portion of the extreme points of the integer convex hull. One cannot therefore use these points to construct the actual convex hull. Yet, even though one does not generate all integer feasible vertices of the convex hull, 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.

2. The CHR algorithm

Consider now the more general nonlinear integer program (NLIP)

$$\text{(NLIP)} \qquad \min_x \ \{f(x) \mid x \in S\}$$

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

<u>Definition 2.1.</u> The Convex Hull Relaxation of the nonlinear integer programming problem (NLIP) min {f(x) | x∈S}

is the continuous nonlinear programming problem

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

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). See Figure 1. it is easy to see however that (CHR) is a relaxation of (NLIP).

--Insert Figure 1 about here –

This relaxation is a <u>primal</u> relaxation, in the x-space. It is actually a primal relaxation that does not "relax" any constraint. The apparent difficulty in solving (CHR) is related to 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. (1987) with Restricted Simplicial Decomposition, provides an efficient way to solve (CHR) to optimality, by solving a sequence of, a priori simpler, linear integer direction-finding problems and of nonlinear continuous problems over a simplex, which is a progressively enlarged inner approximation of the original polytope.

3. Applying simplicial decomposition to the CHR problem

3.1. Assumptions

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

- (i) the feasible region must be compact and convex,
- (ii) the objective function must be pseudoconvex, and
- (iii) the constraints must be linear.

Condition (1) is automatically satisfied for pure 0-1 nonlinear problems with linear constraints. In this paper we only consider MINLPs with linear constraints, but the objective function may be nonconvex, in which case we may only get a local optimum (see Figure 2).

-- Insert Figure 2 about here --

A study is currently underway to possibly enlarge CHR to problems with convex nonlinear constraints.

In what follows, we will assume that problem (NLIP) is of the form

(NLIP) min x {f(x) | s∈S}

with S = {x∈Rn | Ax ≤ b, x ∈ Y}, and that co(S) is compact and convex, so that conditions (1) and (iii) are satisfied. We will consider separately the convex and nonconvex cases for f(x).

We will now describe how SD (simplicial decomposition) can be applied to CHR.

3.2. The Subproblem

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

(CHS) min y $\{\nabla f(x^k)T \times (y - x^k) \mid y \in Co \{Ax \le b, x \in Y\}\}$.

We refer to x^k as the linearization point.

--Insert Figure 3 about here –

(CHS) is an implicit linear problem, indeed we assumed that co(S) is compact and convex, and its feasible region is a polytope. (CHS) will thus have at least an optimal solution that is an extreme point of the convex hull (see Figure 3). It can therefore be reformulated as an explicit integer program (see Figure 4), which we will call the Integer Programming Subproblem (IPS):

(IPS) $\min_y \{\nabla f(x^k) \times (y - x^k) \mid Ay \leq b, y \in Y\}$.

--Insert Figure 4 about here –

(IPS) will have at least a basic optimal solution, y^k, that is a vertex of the convex hull. If it is a new point, it will further expand the inner approximation, and one will solve a new improved master problem. If not, x^k is already optimal for the convex hull relaxation (CHR) problem. Indeed if the solution to (IPS), or (CHS), lies within the current inner approximation to co(S), then $\nabla f(x^k) \times (y - x^k) \geq 0$ must hold for all $y \in co(S)$, and the current linearization point satisfies the sufficiency optimality condition, thus is optimal for (CHR).

At each iteration k, we obtain an integer feasible point y^k to the original (NLIP) problem. Problem (IPS) is usually much easier to solve than problem (NLIP) because its objective function is linear. Convergence to the optimal solution will be discussed in section 4. If x^k is not optimal, we proceed to the master problem.

3.3. The Master Problem

The following nonlinear continuous programming problem with one simple constraint is called the *Master Problem* (MP):

$$\text{(MP)} \quad \text{Min}_\beta \ \{f(X\beta) \mid \textstyle\sum_i \beta_i = 1, \beta_i \geq 0, i = 1, 2, ..., r\}.$$

It is a problem in the weight vector β. X is the growing n × r matrix comprised of a subset of extreme points of the convex hull found while solving the subproblems, along with the current linearization point x^k or a past one. There are r such points in X. Then at the master problem stage, (MP) is solved, which is a minimization problem over an $r-1$ dimensional simplex that approximates from the inside the convex hull of S. If the optimal solution of (CHR) is within this simplex, then the

algorithm will terminate. If not, the optimal solution β∗ of (MP) will be used to compute the next linearization point, xk+1, which can be
computed using the formula:

$$x^{k+1} = \sum_i \beta^*_i \times X_i .$$

Then one returns to the subproblem, finds another extreme point and increases the dimension of the simplex for (MP).

Figures 5 and 6 show the first and last iterations in the application of SD to the problem of minimizing the distance from the polytope to the center of the circle. The polytope is the convex hull of its vertices. Vertices y(1), y(2) and y(3) are the subproblems solutions at iterations 1, 2 and 3 respectively. Point z(0) is the original linearization point, z(1)= y(1) is the closest point to the center on the line segment between z(0) and y(1). z(2) is the closest point to the center in the triangle with vertices z(0), y(1) and y(2). Finally z(3) = y(3) is the closest point to the center in the polytope with vertices z(0), y(1), y(2) and y(3). The next subproblem yields y(3) again, and the algorithm has converged to an optimal solution z(3). Notice that not all vertices of the polytope are needed for the algorithm to converge. In our convex experiments, given the dimensions of the problems solved, the number of vertices we generated was obviously very small compared to the (unknown) total number of vertices.

--insert Figures 5 and 6 about here --

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.

3.4. Convergence to the Optimal Solution of CHR

As already mentioned, 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).

4. Calculating lower and upper bounds for convex 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

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

4.1 Convex GQAPs.

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.

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

$$
\begin{aligned}
\text{(GQAP)} \quad & \text{Min} \quad f(x) = x^T Q\, x + h^T.\, x \\
& \text{s.t.} \quad \textstyle\sum_i x_{ij} = 1 \qquad \forall i \qquad \text{(MC)} \\
& \qquad\quad \textstyle\sum_i a_{ij}\, x_{ij} \le b_j \qquad \forall j \qquad \text{(KP)} \\
& \qquad\quad x_{ij} \in \{0, 1\}, \qquad \forall i,j \qquad \text{(0-1)}
\end{aligned}
$$

where $f(x) = x^T Q\, x + h^T.\, x$ is a quadratic convex 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} .

We used two types of data. Both were adapted from the literature, with problems having up to 600 variables, generating instances with either small or relatively large residual integrality gaps. We measured the improvement over the continuous bound as one replaces the NLP bound by the CHR bound. The optimal integrality gaps are not known, as we were not able to solve any but the smallest two problems to optimality using a specialized BB code. We use the best integer solution found during the SD run to compute the integrality gaps. The BB code on the other hand was able to solve to optimality most of the original instances from the literature, so it seems that the problems we generated here are considerably more difficult.

For this first series of instances, we computed the matrix of the objective function by premultiplying the original matrix Q by its transpose, where the entries of Q are products of a flow by a distance as given in the original GAP instances. These problems tend to have moderate integrality gaps. The improvement was in the range of 43 to 99 %. The largest runtime on a fast workstation was 12 seconds. The results can be found in Table 1.

--Insert Table 1 about here --

The second data set uses the same data from the literature, and generates the objective function matrix Q as the product of a matrix by its transpose, but this matrix is now randomly generated with coefficients between -a and +b, like -20 and +15. These tend to have large integrality gaps and to be considerably more difficult to solve. The results can be found in Table 2. We show in Ahlatçıoğlu and Guignard (2010) that for such convex problems, using Plateau's Quadratic Convex Reformulation (QCR) (2006) before using CHR yields substantially reduced integrality gaps. The effect of QCR is to increase the continuous bound, and possibly the CHR bound, and the numerical experiment show that this is indeed the case. In addition to producing an increased lower bound, CHR produces good feasible solutions quickly, which might also prove useful when designing BB codes.

5. Using CHR as a heuristic for nonconvex problems

We tested the ability of CHR to find optimal or almost optimal feasible solutions for nonconvex problems. We found out that for nonconvex problems, the algorithm terminated very quickly, as a consequence generating only a few feasible solutions. It also was clear that the result depended very much on the choice of the initial linearization point. Given that a single SD run took only seconds, even for large problems, we investigated the possibility of starting from a variety of linearization points. Our results show that in fact, no single rule is always best, and every one of our starting strategies proved best for a few examples. The initial linearization point is the optimal solution of the linear problem obtained by replacing the quadratic objective function in the original problem by the linear function $q^T. x$, where q is one of the options below, and either <u>min</u>imizing or <u>max</u>imizing $q^T. x$.

The choices we used for q were:

y.l: the optimal solution of the continuous relaxation of the problem.

1: all entries equal to 1

0: all entries equal to 0

Rand: randomly generated

The QAP results (see Tables 3, 4 and 5) with standard instances from the QAPLIB library show the details of the runs for each of these 2x4 choices. In red one can see which choice yielded the best result. Final gaps are tabulated in the last column. Tables 6 and 7 report the number of iterations of CHR.

The MSAP problem is a multi-story space assignment problem (see Figure 7)that can be viewed a special case of GQAP . A more thorough description can be found in Hahn et al. (2010). Table 8 gives results for problems with up to 286 binary variables.

--insert Figure 7 about here --

The CDAP, or cross-dock assignment problem, assigns incoming doors to incoming trucks and outgoing doors to outgoing trucks in a cross-dock in order to minimize the cost of goods traffic within the cross-dock (see Figure 8). This problem can be modeled as a GQAP or a GQ3AP. Figure 9 shows run time comparisons between these two models and CHR. CHR times are remarkably constant for different dimensions, contrary to the other two models. Table 9 shows results with CHR, demonstrating once again that it can produce quality solutions in a very short time.

Finally we ran most GQAP instances available in the literature. We compared results with those of Cordeau et al's memetic algorithm. We obtain solutions of the same quality in seconds instead of minutes.

6. Conclusion

The convex hull relaxation method appears as a powerful tool for solving approximately nonlinear integer programming problems with linear constraints. In the convex case, it provides a lower bound that often strongly dominates the continuous relaxation bound. In the convex as well as the nonconvex case, it provides feasible solutions that are almost always close to optimal or even optimal, in times that are surprisingly small. The number of iterations does not grow substantially with problem size. The nonlinear programs are growing slowly, starting with two variables and increasing by one at each iteration. In the nonconvex case, SD terminates very quickly and this explains why runtimes are so small. In the convex case, the number of iterations is larger, but has never become unmanageable. The linear integer problems are of constant size, and usually much easier to solve than the original nonlinear integer problem. Occasionally though one may encounter a particular linearized objective that make this instance much harder than the others. The LIP times depend heavily on the linear solvers used in the CHR implementation. We used minos or conopt to solve the NLPs and cplex to solve the LIPs, under GAMS. The experiments were done over a couple of years, and we always used the most recent implementation of GAMS and of the solvers, starting with cplex 11 and ending with cplex 12. Most runs were done on a Lenovo Thinkpad X61 Centrino duo running Windows XP.

Future research will concentrate on extending the approach to problems with nonlinear constraints, most likely combining CHR with outer approximation (OA) ideas. CHR is based on an inner approximation of the integer convex hull, while OA is based on an outer approximation of the continuous feasible region. CHR provides bounds and feasible solutions, thus a bracket on the optimum, while OA when it converges produces an integer optimal solution. OA makes more sense for mixed-integer problems, while CHR can be applied indifferently to pure or mixed integer problems. One could try to obtain bounds by using CHR over outer approximations of the continuous constraint set. Work in that direction is in progress.

7. References

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., M. Bussieck, M. Esen, M. Guignard, J. Jagla and A. Meeraus, (2010), "Combining QCR and CHR for Convex Quadratic MINLP Problems with Linear Constraints," submitted for publication.

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

Albornoz, V., (1998), "Diseno de Modelos y Algoritmos de Optimizacion Robusta y su Aplicacion a la Planificacion Agregada de la Produccion." Doctoral Dissertation, Universidad Catolica de Chile, Santiago, Chile.

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., (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.

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

Hahn P., J. MacGregor Smith and Yi-Rong Zhu, (2010), "The Multi-Story Space Assignment Problem," Ann Oper Res (2010) 179: 77–103.

Held M., R. M. Karp, (1970), "The Traveling-Salesman Problem and Minimum Spanning Trees," Operations Research 18, No. 6, 1138-1162

Held M., R. M. Karp, (1971), The Traveling-Salesman Problem and Minimum Spanning Trees: Part II," Mathematical Programming 1, 6–25.

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, A., P.M. Hahn, M. 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.

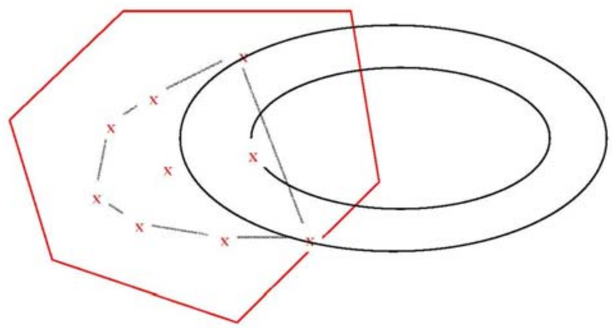
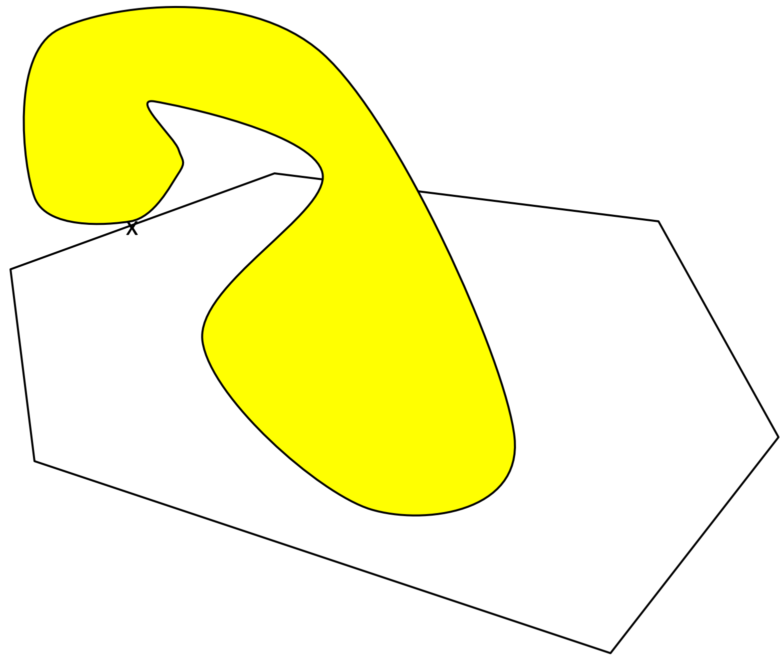
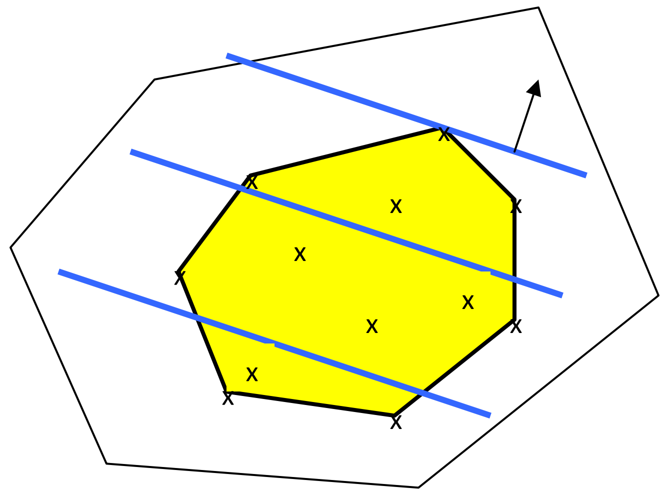


Figure 1
In integer nonlinear programming, the optimal solution may be inside the convex hull of integer feasible solutions



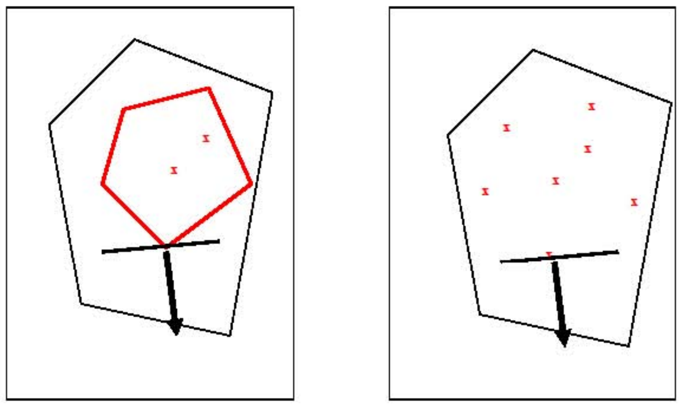
In the nonconvex case, a stationary point such as x may not be a global minimizer, and the value of the objective function at that point may not be a minimum.

Figure 2



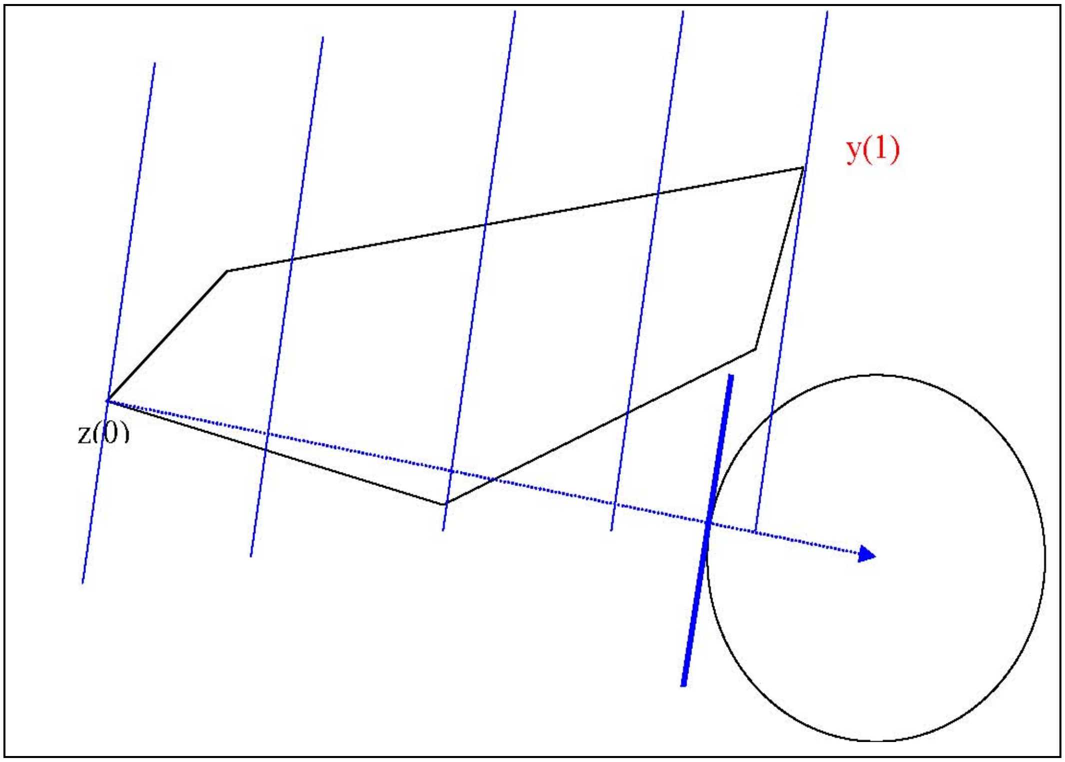
In linear integer programming, if the problem is bounded, the optimal solution is always a boundary point of the convex hull of integer feasible solutions, and one optimal solution is a corner (integer) point.

Figure 3



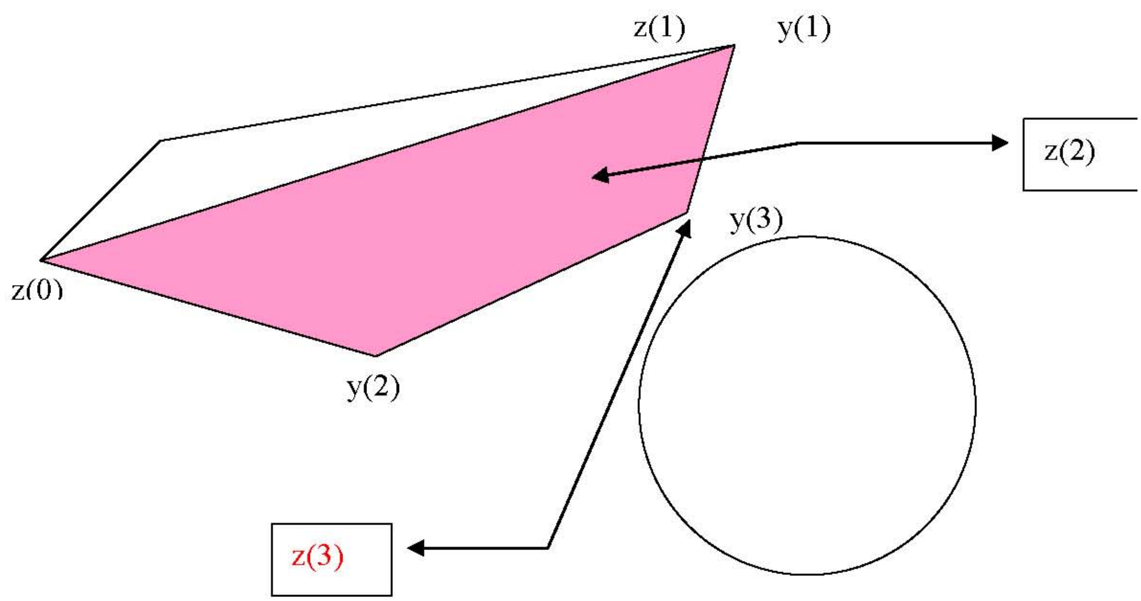
the min over the convex hull of the integer points is equal to the min over the integer points

Figure 4



Simplicial Decomposition: iteration 1
At z(0), linearize the obj. fn. and minimize over the polyhedron
$\Rightarrow$ extreme point y(1)

Figure 5



SD minimize the nonlinear obj. fn. over the convex hull of z(0), y(1), y(2) and y(3) → feasible point z(3)

Figure 6

Facility Evacuation Design (MSAP)

- Locate departments in a multi-story building to simultaneously minimize:
  - Time wasted moving between departments
  - Delay getting to an adequate escape route
- Take into account:
  - Population and space requirements
  - Escape route capacities and delays
  - Department location needs

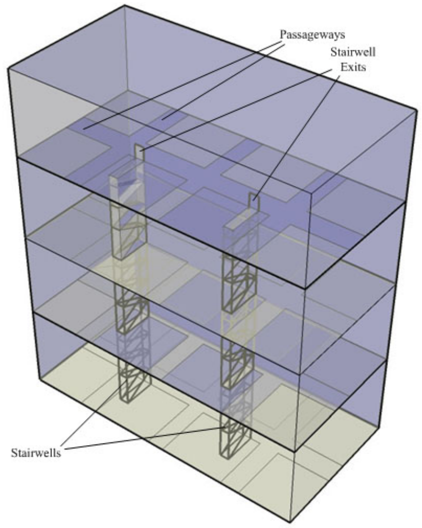


Figure 7

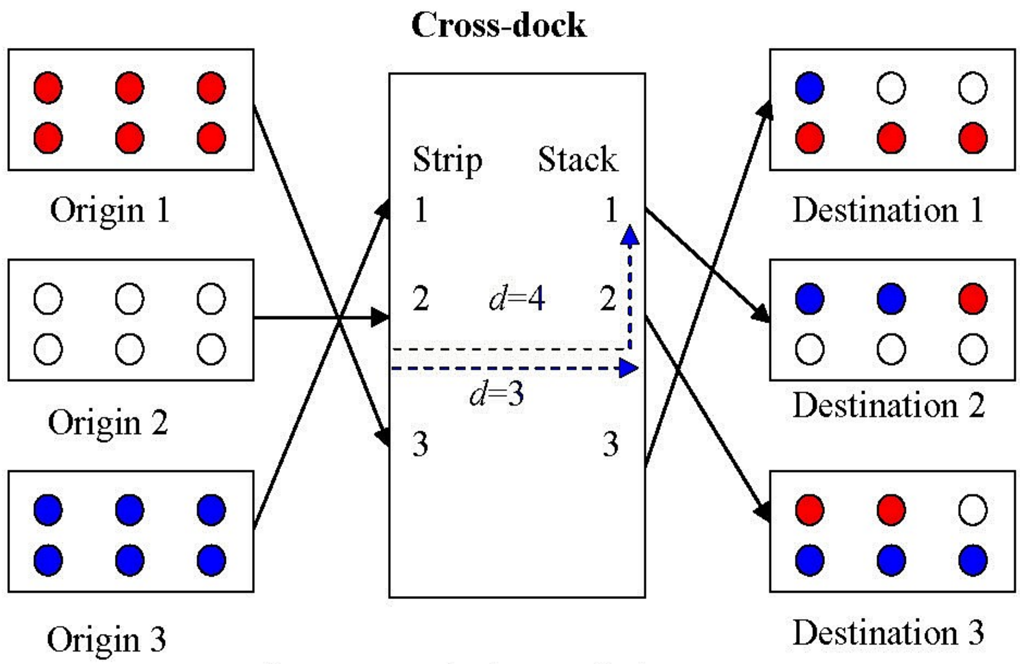


Figure 8 – Typical Cross-dock Layout

Figure 9

Times in seconds for solving CDAP using CHR or GQAP / GQ3AP solvers

	12x6	14x7	15x8	18x7	25x8
GQ3AP	341	2143	17990	--	--
GQAP	214	314	989	13800	28800
CHR	16	21	7	20	18

Table 1

CHR results for convex GQAPs with small integrality gaps

Best IFV: best integer feasible value, CB: continuous bound. On a Thinkpad x61 dual core.

Instance	Continuous bound	CHR bound	Best IFV	CB % gap	CHR % gap	% gap improvement	CPU sec./ # iter.
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 2

CHR applied to convex GQAP instances with large duality gaps

instance	# vars	V(cont)	T(cont)	Gap(cont)	V(CHR)	V(BFS)	T(CHR)	Gap(CHR)	imptrvt
c16x7_ldg_15_2 0	112	3068411		29%	3234964	4328390	42	25%	**13%**
c-30-06-95_ldg_20_15	180	14523694	51	8%	14560611	15843402	2	8%	**3%**
c-30-06-95_ldg_15_20	180	9912936		12%	9926556	11254676	4	12%	**1%**
c20x10_ldg_15_20	200	4183813		21%	4191066	5295100	2	21%	**1%**
c-30-07-75_ldg_20_15	210	14483856		10%	14528676	16017068	1	9%	**3%**
c-30-07-75_ldg_15_20	210	9459033	50	13%	9474383	10916198	15	13%	**1%**
c-30-08-55_ldg_20_15	240	13860307		8%	13881814	15075317	5	8%	**2%**
c-30-08-55_ldg_15_20	240	9174920		12%	9188671	10480453	4	12%	**1%**
c-40-07-75_ldg_15_20	280	16377748	51	9%	16423144	18041204	5	9%	**3%**
c-30-10-65_ldg_20_15	300	13399656		10%	13453725	14846665	1	9%	**4%**
c-30-10-65_ldg_15_20	300	8952179	56	14%	8954776	10353966	11	14%	**0%**
c-20-15-35_ldg_20_15	300	5932552		18%	5940992	7237544	6	18%	**1%**
c-20-15-55_ldg_20_15	300	5941673		20%	5978807	7399266	3	19%	**3%**
c-20-15-75_ldg_20_15	300	5969412		22%	6205973	7698854	3	19%	**14%**
c-20-15-35_ldg_15_20	300	3934019		25%	3934243	5219490	7	25%	**0%**
c-20-15-55_ldg_15_20	300	3943191		28%	3996137	5499388	11	27%	**3%**
c-20-15-75_ldg_15_20	300	3969636		29%	4167590	5618908	12	26%	**12%**
c-40-09-95_ldg_10_20	360	77027537	60	1%	77257318	78105845	8	1%	**21%**
c-40-09-95_ldg_15_20	360	15721361	65	11%	15804077	17587356	7	10%	**4%**
c-40-10-65_ldg_10_20	400	75991473		1%	76060242	76761733	2	1%	**9%**
c-40-10-65_ldg_15_20	400	15191883		8%	15199578	16515897	14	8%	**1%**
c-50-10-65_ldg_15_20	500	22985184		6%	22988259	24545547	6	6%	**0%**
c27x21_ldg_20_15	567	9890615	134	13%	9910219	11416733	13	13%	**1%**
c27x21_ldg_15_20	567	6493746		21%	6990670	8205946	0	15%	**29%**
c-30-20-35_ldg_20_15	600	11980414		11%	11999672	13446607	5	11%	**1%**
c-30-20-35_ldg_15_20	600	7932361	51	14%	8333435	9264122	0	10%	**30%**

Table 3

Integrality gaps for various initial strategies for QAPs

	Opt. Value	y.l max %	y.l min %	1 min %	1 max %	0 max %	0 min %	rand min %	rand max %	min.gap
bur 26a	5426670	0.3	0.8	0.7	0.9	0.9	0.9	0.5	0.9	**0.3**
bur 26b	3817852	0.2	0.5	0.5	0.5	0.5	0.5	0.7	0.3	**0.2**
bur 26c	5426795	0.1	0.5	0.0	0.0	0.3	0.3	1.5	0.1	**0.0**
bur 26d	3821225	0.1	0.5	0.0	0.0	0.4	0.4	1.6	1.3	**0.0**
bur 26e	5386879	0.0	11.7	0.3	13.3	1.0	1.0	1.1	15.4	**0.0**
bur 26f	3782044	0.0	9.4	0.5	0.4	0.4	14.0	1.6	0.4	**0.0**
bur 26g	10117172	0.3	5.6	0.7	13.7	0.1	11.4	1.2	13.9	**0.1**
bur 26h	7098658	0.3	0.8	0.1	14.3	1.8	1.8	2.1	1.9	**0.1**
kra 32	88700	4.3	5.4	5.0	7.3	2.9	5.3	5.5	3.8	**2.9**
lipa 30a	13178	2.2	2.5	2.6	2.7	2.5	2.5	2.6	2.9	**2.2**
lipa 30b	151426	0.0	29.9	15.6	20.2	19.2	19.2	0.0	30.1	**0.0**
nug 12	578	4.2	3.1	5.9	4.5	2.1	2.1	8.0	2.4	**2.1**
nug 14	1014	2.2	0.2	6.1	1.8	11.8	11.8	3.9	1.6	**0.2**
nug 15	1150	3.3	1.4	1.9	3.8	1.9	0.9	0.0	3.3	**0.0**
nug 16a	1610	1.7	2.2	0.7	0.1	0.7	0.7	1.9	0.7	**0.1**
nug 16b	1240	1.0	1.9	0.0	1.0	5.5	5.5	1.9	0.0	**0.0**
nug 17	1732	0.1	4.4	0.0	2.4	1.0	1.0	3.0	2.1	**0.0**
nug 18	1930	3.7	5.8	2.4	3.5	4.9	4.9	1.1	4.0	**1.1**
nug 20	2570	2.1	1.3	1.6	1.6	0.0	0.0	0.5	5.8	**0.0**
nug 21	2438	1.6	1.3	2.2	2.1	0.4	0.4	2.4	1.2	**0.4**
nug 22	3596	0.7	3.2	0.1	7.9	2.1	2.1	2.7	1.8	**0.1**
nug 24	3488	1.1	1.1	2.5	0.2	0.9	0.9	3.2	0.6	**0.2**
nug 25	3744	1.0	0.4	0.3	0.3	1.1	1.1	0.3	0.3	**0.3**
nug 27	5234	1.4	0.8	5.7	9.3	3.4	3.4	0.5	3.7	**0.5**

Table 4

Integrality gaps for various initial strategies for QAPs

	Opt. Value	**y.l max %**	**y.l min %**	**1 min %**	**1 max %**	**0 max %**	**0 min %**	**rand min %**	**rand max %**	**min. gap**
nug 28	5166	4.0	2.9	2.8	1.9	9.5	9.5	2.6	1.2	**1.2**
nug30	6124	1.1	8.3	0.8	2.4	1.6	1.6	1.6	1.0	**0.8**
ste 36a	9526	6.2	18.2	6.9	12.1	12.5	12.5	14.7	9.9	**6.2**
ste 36b	15852	3.1	11.9	15.2	37.7	14.4	14.4	13.2	14.5	**3.1**
ste 36c	8239110	4.1	7.0	7.7	10.2	4.7	4.7	8.1	11.1	**4.1**
tai12a	224416	0.0	7.3	7.4	6.0	12.6	12.6	8.9	7.8	**0.0**
tai 12b	39464925	12.2	43.3	26.9	17.8	32.5	32.5	18.2	4.8	**4.8**
tai 15a	388214	2.8	4.3	4.4	3.8	1.1	1.1	2.9	2.5	**1.1**
tai 15b	51765268	0.3	1.0	0.6	1.5	0.6	0.6	1.0	1.2	**0.3**
tai 17a	491812	3.1	2.4	4.6	0.5	6.0	6.0	4.4	7.7	**0.5**
tai20a	703482	3.2	5.2	8.2	4.7	6.3	6.3	2.1	5.3	**2.1**
tai20b	122455310	12.7	12.3	11.5	0.6	4.3	4.3	1.4	10.7	**0.6**
tai 25a	1167256	2.5	3.6	6.5	1.8	4.0	4.0	5.1	4.5	**1.8**
tai 25b	344355646	0.5	21.1	9.6	18.5	5.2	5.2	7.4	4.8	**0.5**

Table 5

Integrality gaps for various initial strategies for QAPs

	Opt. Value	**y.l max %**	**y.l min %**	**1 min %**	**1 max %**	**0 max %**	**0 min %**	**rand min %**	**rand max %**	**min.gap**
bur 26a	5426670	0.3	0.8	0.7	0.9	0.9	0.9	0.5	0.9	**0.3**
bur 26b	3817852	0.2	0.5	0.5	0.5	0.5	0.5	0.7	0.3	**0.2**
bur 26c	5426795	0.1	0.5	0.0	0.0	0.3	0.3	1.5	0.1	**0.0**
bur 26d	3821225	0.1	0.5	0.0	0.0	0.4	0.4	1.6	1.3	**0.0**
bur 26e	5386879	0.0	11.7	0.3	13.3	1.0	1.0	1.1	15.4	**0.0**
bur 26f	3782044	0.0	9.4	0.5	0.4	0.4	14.0	1.6	0.4	**0.0**
bur 26g	10117172	0.3	5.6	0.7	13.7	0.1	11.4	1.2	13.9	**0.1**
bur 26h	7098658	0.3	0.8	0.1	14.3	1.8	1.8	2.1	1.9	**0.1**
kra 32	88700	4.3	5.4	5.0	7.3	2.9	5.3	5.5	3.8	**2.9**
lipa 30a	13178	2.2	2.5	2.6	2.7	2.5	2.5	2.6	2.9	**2.2**
lipa 30b	151426	0.0	29.9	15.6	20.2	19.2	19.2	0.0	30.1	**0.0**
nug 12	578	4.2	3.1	5.9	4.5	2.1	2.1	8.0	2.4	**2.1**
nug 14	1014	2.2	0.2	6.1	1.8	11.8	11.8	3.9	1.6	**0.2**
nug 15	1150	3.3	1.4	1.9	3.8	1.9	0.9	0.0	3.3	**0.0**
nug 16a	1610	1.7	2.2	0.7	0.1	0.7	0.7	1.9	0.7	**0.1**
nug 16b	1240	1.0	1.9	0.0	1.0	5.5	5.5	1.9	0.0	**0.0**
nug 17	1732	0.1	4.4	0.0	2.4	1.0	1.0	3.0	2.1	**0.0**
nug 18	1930	3.7	5.8	2.4	3.5	4.9	4.9	1.1	4.0	**1.1**
nug 20	2570	2.1	1.3	1.6	1.6	0.0	0.0	0.5	5.8	**0.0**
nug 21	2438	1.6	1.3	2.2	2.1	0.4	0.4	2.4	1.2	**0.4**
nug 22	3596	0.7	3.2	0.1	7.9	2.1	2.1	2.7	1.8	**0.1**
nug 24	3488	1.1	1.1	2.5	0.2	0.9	0.9	3.2	0.6	**0.2**
nug 25	3744	1.0	0.4	0.3	0.3	1.1	1.1	0.3	0.3	**0.3**
nug 27	5234	1.4	0.8	5.7	9.3	3.4	3.4	0.5	3.7	**0.5**

Table 6

Number of SD iterations for QAP instances solved heuristically by CHR

	y.l		e		0		Random		
Instance ID	**Max**	**Min**	**Min**	**Max**	**Max**	**Min**	**Min**	**Max**	**Average**
nug 25	13	32	28	23	24	24	26	27	**24.625**
nug 27	9	27	6	4	30	30	23	41	**21.25**
nug 28	7	27	27	23	4	4	32	31	**19.375**
nug30	7	6	51	27	34	34	23	20	**25.25**
ste 36a	12	22	38	26	22	22	33	27	**25.25**
ste 36b	7	16	15	23	17	17	25	19	**17.375**
ste 36c	13	52	51	46	57	57	44	50	**46.25**
tai12a	4	21	19	17	21	21	13	20	**17**
tai 12b	7	17	17	19	14	14	14	13	**14.375**
tai 15a	9	25	18	26	48	48	30	23	**28.375**
tai 15b	7	19	19	5	20	20	28	19	**17.125**
tai 17a	8	21	32	30	26	26	24	20	**23.375**
tai20a	23	29	7	35	24	24	20	26	**23.5**
tai20b	11	31	42	26	25	25	26	29	**26.875**
tai 25a	13	50	10	32	22	22	28	28	**25.625**
tai 25b	17	32	54	49	34	34	31	40	**36.375**
Average	9.8	21.7	24.6	22.9	22.8	21.6	23.2	22.1	

Table 7

Number of SD iterations for QAP instances solved heuristically by CHR

	y.l		e		0		Random		
Instance ID	**Max**	**Min**	**Min**	**Max**	**Max**	**Min**	**Min**	**Max**	**Average**
bur 26a	3	8	7	5	5	5	16	5	**6.75**
bur 26b	5	11	22	19	19	19	29	12	**17**
bur 26c	4	17	18	24	13	13	4	13	**13.25**
bur 26d	3	10	16	20	13	13	8	8	**11.375**
bur 26e	2	1	10	1	6	6	8	1	**4.375**
bur 26f	3	1	8	14	16	1	8	27	**9.75**
bur 26g	3	1	5	1	18	1	9	1	**4.875**
bur 26h	4	11	11	1	6	6	7	5	**6.375**
kra 32	17	38	27	50	52	46	58	59	**43.375**
lipa 30a	34	61	58	52	34	34	28	26	**40.875**
lipa 30b	2	1	53	9	9	9	24	1	**13.5**
nug 12	11	13	6	12	18	13	11	18	**12.75**
nug 14	13	26	26	31	8	8	26	31	**21.125**
nug 15	9	18	23	17	27	25	27	20	**20.75**
nug 16a	13	21	26	25	27	27	32	31	**25.25**
nug 16b	10	34	25	38	31	31	22	22	**26.625**
nug 17	12	11	28	27	20	20	28	24	**21.25**
nug 18	16	13	24	23	10	10	19	20	**16.875**
nug 20	10	28	31	31	22	22	22	7	**21.625**
nug 21	12	27	16	21	20	20	14	24	**19.25**
nug 22	9	22	35	5	21	21	32	24	**21.125**
nug 24	11	27	27	33	48	48	31	26	**31.375**

Table 8

MSAP results

OV:optimal value in integers.

Problem name	OV	CHR	gap %	dimensions as GQAP		
10x7x2-loose_1_Rescaled	2582	2582	0.00%	9	x	20
10x7x2-loose_1_Rescaled_rev	2582	2582	0.00%	9	x	20
10x7x2-tight_1_Rescaled	2582	2582	0.00%	9	x	20
13x7x2-large-tightNE_10_Rescaled	27395	30515	11.39%	9	x	28
13x7x2-large-tight_1_Rescaled	2582	2894	12.08%	9	x	28
13x8x2-med-loose_10_Rescaled	6895	6955	0.87%	10	x	26
13x8x2-med-tight_10_Rescaled	6895	7285	5.66%	10	x	26
13x8x2-small-loose_100_Rescaled	38335	42010	9.59%	10	x	26
13x8x3-large-tight_1_Rescaled	3086	3086	0.00%	11	x	26
13x8x3-med-loose_10_Rescaled	6305	6955	10.31%	11	x	26
13x8x3-med-tight_10_Rescaled	6305	6305	0.00%	11	x	26
13x8x3-small-loose_100_Rescaled	35105	35815	2.02%	11	x	26
13x8x3-small-tight_100_Rescaled	35105	35105	0.00%	11	x	26
Teste_1 _Rescaled	53	53	0.00%	4	x	6

Table 9

CDAP results

OV:optimal value in integers.

Problem name	OV	CHR	gap %	dimensions as GQAP		
10_5ave	661232	661331	0.02	10	x	20
10_610_6	640741	640741	0.00	10	x	20
10_710_7	639120	639120	0.00	10	x	20
10_810_8	639120	648031	1.39	10	x	20
10_910_9	639120	645420	0.99	10	x	20
5_55_5	654614	654614	0.00	10	x	20
5_65_6	652686	652686	0.00	10	x	20
5_85_8	636546	636546	0.00	10	x	20
5_95_9	636546	636546	0.00	10	x	20
6_56_5	662480	662480	0.00	10	x	20
6_66_6	649861	649861	0.00	10	x	20
6_76_7	649861	649861	0.00	10	x	20
6_86_8	649861	649861	0.00	10	x	20
6_96_9	634903	634903	0.00	10	x	20
7_57_5	676802	676802	0.00	10	x	20
7_67_6	661125	661125	0.00	10	x	20
7_77_7	652871	653246	0.06	10	x	20
7_87_8	644794	650662	0.91	10	x	20
7_97_9	631062	636882	0.92	10	x	20
8_58_5	655627	655943	0.05	10	x	20
8_68_6	641766	641766	0.00	10	x	20
8_78_7	634576	641766	1.13	10	x	20
8_88_8	633608	633608	0.00	10	x	20
8_98_9	630218	630218	0.00	10	x	20
9_59_5	652000	653542	0.24	10	x	20
9_69_6	647691	651449	0.58	10	x	20
9_79_7	645443	649921	0.69	10	x	20
9_89_8	642514	642514	0.00	10	x	20
9_99_9	636926	638189	0.20	10	x	20