Combinatorial and Global Optimization, pp. 297–303
P.M. Pardalos, A. Migdalas and R. Burkard, Editors

Tight QAP bounds via linear programming

K.G. Ramakrishnan (kgr@research.bell-labs.com)
Mathematical Sciences Research Center, Bell Laboratories

M.G.C. Resende (mgcr@research.att.com)
Information Sciences Research Center, AT&T Labs Research

B. Ramachandran (bramacha@ecn.purdue.edu)
Department of Chemical Engineering, Purdue University

J.F. Pekny (pekny@ecn.purdue.edu)
Department of Chemical Engineering, Purdue University

Abstract

Lower bounds for the quadratic assignment problem (QAP) tend to deteriorate rapidly with the size of the QAP. Recently, Resende, Ramakrishnan, and Drezner (1995) computed a linear programming based lower bound for the QAP using an interior point algorithm for linear programming to solve the linear programming relaxation of a classical integer programming formulation of the QAP. That linear program can be viewed as a two-body interaction formulation. Those bounds were found to be the tightest for a large number of instances from QAPLIB, a library of QAP test problems. In this paper, we apply the same interior point approach to compute lower bounds derived from the three-body interaction formulation of Ramachandran and Pekny (1996). All instances from QAPLIB, having dimension up to $n = 12$, were solved. The new approach produces tight lower bounds (lower bounds equal to the optimal solution) for all instances tested. Attempts to solve the linear programming relaxations with CPLEX (primal simplex, dual simplex, and barrier interior point method) were successful only for the smallest instances ($n \leq 6$ for the barrier method, $n \leq 7$ for the primal simplex method, and $n \leq 8$ for the dual simplex method).

Keywords: Quadratic assignment, linear programming bounds.

1 LP-based lower bounds for the QAP

In this section we briefly review integer programming formulations of the QAP that are useful for producing lower bounds. Let the binary variables x_{ij} represent the assignment of facility i to location j and denote by c_{ijkl} the cost of assigning facility i to location j and facility k to location l. The QAP can be formulated as the following integer quadratic program:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} c_{ijkl}\ x_{ij}\ x_{kl} \tag{1}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots, n, \tag{2}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, \ldots, n, \tag{3}$$

$$x_{ij} \in \{0,1\}, \quad i,j = 1, \ldots, n. \tag{4}$$

Linear programming based bounds for the QAP [1, 9, 7] have relied on the following mixed integer formulation obtained by the linearization of the quadratic objective with the introduction of continuous variables $y_{ijkl} = x_{ij}\ x_{kl}$. The resulting linear integer program is

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k\neq i}\sum_{l>j} c_{ijkl}\ y_{ijkl} \tag{5}$$

subject to (2, 3, 4) and

$$\sum_{i\neq k} y_{ijkl} = x_{kl}, \quad \forall (j,k,l), l > j \tag{6}$$

$$\sum_{i\neq k} y_{klij} = x_{kl}, \quad \forall (j,k,l), l < j \tag{7}$$

$$\sum_{j<l} y_{ijkl} + \sum_{j>l} y_{klij} = x_{kl}, \quad \forall (i,k,l), k \neq i \tag{8}$$

$$y_{ijkl} \geq 0, \quad \forall (i,j,k,l), i \neq k, j \neq l. \tag{9}$$

Though lower bounds obtained from the linear programming relaxation of this formulation are, in general, better than previously known lower bounds [9], there is still a significant gap between the optimal solution and the lower bound for problems as small as dimension $n = 8$. For example, problem nug08 from QAPLIB [2] has an optimal solution of 214 and an LP-based lower bound of 204. This gap deteriorates with the increase in the size of the problem, necessitating the solution of a large number of linear programs in branch and bound algorithms [8]. For example, nug30 has a best known solution of 6124 and an LP-based lower bound of 4805.

Ramachandran and Pekny [7] have recently proposed a higher-order formulation of the QAP based on the application of lifting procedures to (5–9). Defining three-body interaction coefficients as $c_{ijklpq} = c_{ijkl} + c_{klpq} + c_{ijpq}$, the QAP can be formulated as:

$$\min \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k\neq i}\sum_{l>j}\sum_{p\neq i,k}\sum_{q>l} c_{ijklpq}\, z_{ijklpq} + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k\neq i}\sum_{l\geqslant j} c_{ijkl}\, y_{ijkl} \tag{10}$$

subject to (2–4), (6–9) and

$$\sum_{i\neq k,p} z_{ijklpq} = y_{klpq}, \quad \forall(j,k,l,p,q), p\neq k, l>j, q>l, \tag{11}$$

$$\sum_{i\neq k,p} z_{klijpq} = y_{klpq}, \quad \forall(j,k,l,p,q), p\neq k, j>l, q>j, \tag{12}$$

$$\sum_{i\neq k,p} z_{klpqij} = y_{klpq}, \quad \forall(j,k,l,p,q), p\neq k, q>l, j>q, \tag{13}$$

$$\sum_{j<l<q} z_{ijklpq} + \sum_{l<j<q} z_{klijpq} + \sum_{l<q<j} z_{klpqij} = y_{klpq}, \quad \forall(i,k,l,p,q), \quad p\neq k\neq i, q>l, \tag{14}$$

$$z_{ijklpq} \geq 0. \tag{15}$$

It can be shown that the optimal objective function of (10–15) is $(n-1)$ times that of QAP. Prior to our study, this formulation had been tested only for small instances of QAP of size at most $n = 8$ [7], showing that the LP relaxations were 100% tight in those cases. Larger instances of quadratic assignment problems could not be solved due to the limitations of CPLEX, the LP solver used. Decomposition methods based on this formulation have also yielded better lower bounds than the LP based lower bounds using the formulation (5–9) for a number of problems [7].

In this paper, our main objective is to use the interior point code ADP to obtain superior lower bounds using (10–15).

2 Experimental results

In this section, we describe computational results. Because of the size of the linear programs, we have limited this study to all QAPLIB instances having dimension $n \leq 12$. ADP requires about 1.2 Gbytes of memory to run the largest instances in the test set, which have 299,256 variables and 177,432 constraints.

The experiments were done on a 250MHz Silicon Graphics Challenge. The ADP code is written in C and Fortran. It was compiled with the `cc` and `f77` compilers

Table 1: QAPLIB instances of dimension $n \leq 12$

			LP-based lower bound		linear programming relaxation		
name	n	BKS	RRD95	3-body	rows	cols	NZ(A)
nug05	5	50	50	50	1410	825	5850
nug06	6	86	86	86	3972	2886	20232
nug07	7	148	148	148	9422	8281	57134
nug08	8	214	204	214	19728	20448	139008
nug12	12	578	523	578	177432	299256	1954944
esc08a	8	2	0	2	19728	20448	139008
esc08b	8	8	2	8	19728	20448	139008
esc08c	8	32	22	32	19728	20448	139008
esc08d	8	6	2	6	19728	20448	139008
esc08e	8	7	0	7	19728	20448	139008
esc08f	8	18	18	18	19728	20448	139008
rou10	10	174220	170384	174220	66620	90550	601400
rou12	12	235528	224278	235528	177432	299256	1954944
scr10	10	26992	26874	26992	66620	90550	601400
scr12	12	31410	29827	31410	177432	299256	1954944
lipa10a	10	473	473	473	66620	90550	601400
lipa10b	10	2008	2008	2008	66620	90550	601400

using compiler flags `CFLAGS = -O -DVAX -cckr -p` and `FFLAGS = -O2 -p -trapuv`. Running times were measured by making the system call `times` and converting to seconds, using the `HZ` defined in `sys/param.h`.

ADP requires many parameters to be set. We used the parameter setting described in [9].

Table 2 summarizes these instances, listing for each instance, its name, dimension (n), best known solution (BKS), the lower bound computed by Resende, Ramakrishnan, and Drezner [9] by solving (5–9) (RRD95 bound), the lower bound resulting from the 3-body formulation (3-body), and the dimension of the 3-body linear programming formulation (rows, columns, and number of nonzeros in the coefficient matrix). Note that of the 17 instances, the lower bounds computed in [9] were tight for only 6 instances, whereas all 3-body lower bounds were tight.

Table 2 summarizes the ADP runs. For each instance, the table lists its name, the number of interior point iterations (ipitr), number of conjugate gradient iterations (cgitr), maximum number of conjugate gradient iterations in a single interior point iteration (max-cgitr), average number of conjugate gradient iterations per interior point iteration (avg-cgitr), and number of preconditioners computed (#-precond),

Table 2: ADP interior point solution statistics

name	ipitr	cgitr	max-cgitr	avg-cgitr	#-precnd	time (secs)
nug05	48	311	37	6	36	3.2s
nug06	55	557	32	9	48	12.2s
nug07	59	721	54	12	51	43.3s
nug08	63	1036	83	16	56	139.1s
nug12	91	4655	201	50	86	6504.2s
esc08a	57	1130	75	19	53	146.4s
esc08b	65	4682	425	70	60	435.8s
esc08c	61	953	79	15	51	130.6s
esc08d	61	1110	251	17	51	138.6s
esc08e	66	2472	101	36	62	275.5s
esc08f	59	879	53	14	53	121.5s
rou10	69	1476	101	21	59	800.8s
rou12	80	2736	173	33	72	4222.0s
scr10	71	1788	101	24	60	950.0s
scr12	83	3240	201	38	78	5038.8s
lipa10a	66	943	66	14	58	603.1s
lipa10b	66	900	62	13	54	580.1s

and the total CPU time in seconds.

We make the following observations regarding the experimental results:

- The lower bounds computed are tight for all instances tested.
- No other lower bounding technique for the QAP has produced tight bounds for all instances from this set of problems.
- CPU times ranged from a little over 3 seconds on the smallest instance to a little under 2 hours for the longest $n = 12$ run. In the concluding remarks we discuss the relevance of this to branch and bound methods.

3 Concluding remarks

In this paper, we used an interior point algorithm [5] that uses a preconditioned conjugate gradient algorithm to compute lower bounds for the QAP by solving a linear programming relaxation of the 3-body interaction formulation of Ramachandran and

Pekny [7]. On all QAPLIB [2] instances of dimension $n \leq 12$, the computed lower bounds were tight, i.e. they equaled the optimal objective function value.

A good lower bound by itself is of little use. However, in a branch and bound algorithm, a good lower bound can make a significant difference. Ramakrishnan, Resende, and Pardalos [8] showed that the weaker LP-based lower bound (QAPLP) studied in [9] can reduce substantially the number of nodes of the branch and bound tree that need to be scanned. Though the solution time for computing those bounds is significantly greater than the time needed to compute the classical Gilmore-Lawler bound [3, 6], the large number of scanned nodes for a Gilmore-Lawler based branch and bound algorithm makes the LP-based branch and bound method more attractive, specially for large quadratic assignment problems. For example, using the branch and bound code described in [8], QAPLIB instance `chr18a` was solved after scanning 18 level-1 nodes of the search tree and 17 level-2 nodes in about 1600s, while on the same machine the identical branch and bound code using the Gilmore-Lawler lower bound in place of the LP-based lower bound had not solved the problem after having scanned over 1636 million nodes in over 12 days of CPU time.

To this date, there exist QAPLIB instances of dimension $n = 16$ that remain unsolved. Though solving a 3-body interaction lower bound for $n = 16$ is beyond the capabilities of today's LP solvers, one can use this bound deeper in the search tree, where the subproblems solved have smaller dimension. A practical approach is to combine the QAPLP lower bound to compute bounds for shallow search tree nodes, with the 3-body interaction lower bound to compute bounds for deeper nodes.

Since the 3-body interaction LP contains the entire set of constraints of the LP used for the QAPLP bound, the 3-body bound will always be at least as good as the QAPLP lower bound. Lower bounds that are better than QAPLP but not as good as the 3-body bound can be computed by considering a subset of the constraints (11–14). The number of constraints used should be a function of the depth of the node being scanned in the search tree.

Linear programming formulations of the QAP have been shown to produce tight bounds. Further understanding of structural properties of the QAP polytope will hopefully provide yet tighter bounds. For two recent investigations in this direction, see Rijal [10] and Jünger and Kaibel [4].

References

[1] W.P. Adams and T.A. Johnson. Improved linear programming-based lower bounds for the quadratic assignment problem. In P.M. Pardalos and H. Wolkowicz, editors, *Quadratic assignment and related problems*, volume 16 of *DIMACS*

Series on Discrete Mathematics and Theoretical Computer Science, pages 43–75. American Mathematical Society, 1994.

[2] R.E. Burkard, S. Karisch, and F. Rendl. QAPLIB – a quadratic assignment problem library. *European Journal of Operational Research*, 55:115–119, 1991. Updated version – Feb. 1994.

[3] P.C. Gilmore. Optimal and suboptimal algorithms for the quadratic assignment problem. *J. SIAM*, 10:305–313, 1962.

[4] M. Jünger and V. Kaibel. A basic study of the QAP-polytope. Technical Report 96.215, Institut für Informatik, Universität zu Köln, Köln, Germany, 1996.

[5] N.K. Karmarkar and K.G. Ramakrishnan. Computational results of an interior point algorithm for large scale linear programming. *Mathematical Programming*, 52:555–586, 1991.

[6] E.L. Lawler. The quadratic assignment problem. *Management Science*, 9:586–599, 1963.

[7] B. Ramachandran and J.F Pekny. Higher order lifting techniques in the solution of the quadratic assignment problem. In *State of the Art in Global Optimization: Computational Methods and Applications*, pages 75–92. Kluwer Academic Publishers, 1996.

[8] K.G. Ramakrishnan, M.G.C. Resende, and P.M. Pardalos. A branch and bound algorithm for the quadratic assignment problem using a lower bound based on linear programming. In *State of the Art in Global Optimization: Computational Methods and Applications*, pages 57–73. Kluwer Academic Publishers, 1996.

[9] M.G.C. Resende, K.G. Ramakrishnan, and Z. Drezner. Computing lower bounds for the quadratic assignment problem with an interior point algorithm for linear programming. *Operations Research*, 43:781–791, 1995.

[10] M. Rijal. *Scheduling, design and assigment problems with quadratic costs.* PhD thesis, New York University, 1995.