1969-09

A quadratic assignment problem without column constraints

Greenberg, Harold

Wiley Online

Greenberg, Harold. "A quadratic assignment problem without column constraints." Naval Research Logistics Quarterly 16.3 (1969): 417-421.
http://hdl.handle.net/10945/60198

A QUADRATIC ASSIGNMENT PROBLEM WITHOUT COLUMN CONSTRAINTS

Harold Greenberg

Naval Postgraduate School
Monterey, California

ABSTRACT

We convert a quadratic assignment problem [1] with a nonconvex objective function into an integer linear program. We then solve the equivalent integer program by a simple enumeration that produces global minima.

We consider the problem (a variation of the quadratic assignment problem):

$$\text{(1)}\qquad \begin{aligned} &\text{minimize} \sum_{j=1}^{n}\sum_{i=1}^{m}\sum_{k=1}^{m} a_{ik}x_{ij}x_{kj} \\ &\text{when} \sum_{j=1}^{n} x_{ij}=1 \qquad (i=1,\ldots,m) \\ &x_{ij}=0 \text{ or } 1, \quad (i=1,\ldots,m;\, j=1,\ldots,n) \end{aligned}$$

where $m > n$ and $A=(a_{ik})$ is a given symmetric matrix with nonnegative elements and zero main diagonal terms. Note that (1) is the standard quadratic assignment problem with the column constraints omitted.

For a discussion of problem (1) see [1], where an algorithm is presented that gives a local minimum. Global minima are not always produced by the method of [1] because the objective function in (1) is not convex. In this paper, we convert (1) to an integer program. We then present an algorithm that produces the global minima by a simple enumeration.

THE EQUIVALENT INTEGER PROGRAM

We make the change of variables

$$\text{(2)}\qquad y_{ik}=\sum_{j=1}^{n} x_{ij}x_{kj}, \qquad (i=1,\ldots,m-1;\, k=i+1,\ldots,m).$$

We are interested in converting (1) to a problem containing the y_{ik} alone. Once the y_{ik} are found from the equivalent problem, we must show how to determine the x_{ij}.

Consider the m by n matrix $X=(x_{ij})$ having elements zero or one. We make the following observations:

1. The linear constraints in (1) allow only a single one to appear in each row of X.

2. We obtain $y_{st}=1$ only when $x_{sj}=1$ and $x_{tj}=1$ for $s<t$ and a single j value. Thus the objective function in (1) can have a nonzero value only when X has two or more unit elements in the same column.

3. If d elements of a column of X have unity value then at least $\binom{d}{2}$ elements y_{ik} have unity value. (By $\binom{d}{2}$ we mean two combinations out of d and $\binom{d}{2}=\frac{d(d-1)}{2}\cdot$)

4. If $y_{st}=1$ and $y_{rt}=1$ for $s<r<t$ then $y_{sr}=1$. This is true because $y_{st}=1$ only if $x_{sj}=1$ and $x_{tj}=1$, and $y_{rt}=1$ only if $x_{rj}=1$ and $x_{tj}=1$. Thus, since $x_{sj}=1$ and $x_{rj}=1$ we must have $y_{sr}=1$. Similarly, if $y_{st}=1$ and $y_{tr}=1$ for $s<t<r$, then $y_{sr}=1$.

5. The columns of X may be interchanged in any way and will still produce the same values for y_{ik} and the objective function in (1).

From these observations, we can think of problem (1) as one of loading zeroes and ones into matrix X and calculating the effect on the y_{ik} and in turn the objective function in (1). The major constraint on the y_{ik} is determined by the assignments in X. This constraint is obtained by finding the loading of X that minimizes $\sum_{i=1}^{m-1}\sum_{k=i+1}^{m} y_{ik}$, which is obtained by attempting to equalize the number of ones in each column. We write $m=nd+m_0$, where $d=\left[\frac{m}{n}\right]$ and $m_0=(\text{mod } n)$; $[x]$ means the greatest integer less than or equal to x and mod n is the usual remaindering operation.

A solution that minimizes $\sum_{i=1}^{m-1}\sum_{k=i+1}^{m} y_{ik}$ will have $(n-m_0)$ columns with d ones and m_0 columns with $d+1$ ones. Thus

$$\min \sum_{i=1}^{m-1}\sum_{k=i+1}^{m} y_{ik}=n\binom{d}{2}+dm_0.$$

In general, we have the constraint

$$\sum_{i=1}^{m-1}\sum_{k=i+1}^{m} y_{ik}\geqslant n\binom{d}{2}+dm_0. \tag{3}$$

The integer program equivalent to (1) is

$$\text{minimize} \sum_{i=1}^{m-1}\sum_{k=i+1}^{m} a_{ik}y_{ik}, \tag{4}$$

subject to $y_{ik}=0$ or 1, the results of the fourth observation, and the constraint (3). Once the y_{ik} are found the x_{ij} are obtained by loading the matrix X depending on which y_{ik} values are unity. The observations above tell us the values of x_{ij} to set to one; in particular, the fifth observation indicates that there is considerable freedom in doing so. The major constraint is (3) and although the quantification of the fourth observation requires many constraints, they may be included in a simple enumeration to solve the equivalent integer program. Note that the integer program uses the terms above the main diagonal of A and has a solution value that is one-half that for (1). The integer program is solved in the following algorithm:

We define ϕ as the null set and $|T|$ as the number of elements in set T. We also have $d=\left[\frac{m}{n}\right]$, $m_0=m$ (mod n), and $r=n\binom{d}{2}+dm_0$.

1. List the values of the problem as

	1	2	3	. . .	m
1		a_{12}	a_{13}	...	a_{1m}
2			a_{23}	...	a_{2m}
.				...	.
.				...	.
.				...	.
$m-1$					$a_{m-1,m}$;

set $\bar{s}=0$ for this list. Go to 2.

2. Given lists of the form

$\bar{s}$	1	2	3	. . .	m
$\bar{T}_1$	$\bar{a}_{11}$	$\bar{a}_{12}$	$\bar{a}_{13}$	. . .	$\bar{a}_{1m}$
$\bar{T}_2$	$\bar{a}_{21}$	$\bar{a}_{22}$	$\bar{a}_{23}$	. . .	$\bar{a}_{2m}$
.	.	.	.	. . .	.
.	.	.	.	. . .	.
.	.	.	.	. . .	.
$\bar{T}_{m-1}$	$\bar{a}_{m-1,1}$	$\bar{a}_{m-1,2}$	$\bar{a}_{m-1,3}$	. . .	$\bar{a}_{m-1,m}$

find $\bar{a}_{ef}=\min \bar{a}_{ij}$ for all unmarked elements on all lists. Mark the element. Set $s=\bar{s}$, $T_e=\bar{T}_e$ and go to 3.

3. If $s+|T_e|\geqslant r$ go to 4. Otherwise form an additional list with values obtained from the newly marked list as follows:

(a) Replace the $\bar{T}_e$ heading with $\bar{T}_e=T_e\cup f$.

(b) Delete the f column and the T_e column if it exists. Delete the f row if it exists.

(c) Set $\bar{s}=s+|T_e|$.

(d) Elements in the $\bar{T}_e$ row are $\bar{a}_{eh}=\bar{a}_{ef}+\sum_i a_{ih}$ for $i\epsilon\bar{T}_e$ and where $a_{ih}=a_{hi}$ for $i>h$. The indices h are the column headings. The remaining elements are $\bar{a}_{ij}=\bar{a}_{ef}+\bar{a}_{ij}$. Go to 2.

4. The problem is solved with objective function value $\bar{a}_{ef}$. Using the newly marked list, we find the x_{ij} by loading the X matrix as follows:

(a) Replace $\bar{T}_e$ by $\bar{T}_e=T_e\cup f$.

(b) Set $k=1, j=1$ and go to 4(c).

(c) Set $x_{ij}=1$ for $i\epsilon\bar{T}_k$; set $k=k+1$ and go to 4(d).

(d) If $\bar{T}_k=\emptyset$, go to 4(e). Otherwise set $j=j+1$ and go to 4(c).

(e) If an assignment has been made in row m, end; otherwise set $x_{mn}=1$ and end.

This completes the algorithm. The method implicitly contains an enumeration of all possible values of the y_{ik} that satisfy the fourth observation in order of increasing values of the objective function (4). We stop when constraint (3) is satisfied. The corresponding x_{ij} are then produced and represent the solution to the global minimum of problem (1). This is seen as follows: suppose the values of the y_{ik} and x_{ij} produced by the algorithm are y^0_{ik} and x^0_{ij} and there exist $\bar{x}_{ij}$ that satisfy the constraints in (1) with

$$\sum_{j=1}^{n}\sum_{i=1}^{m}\sum_{k=1}^{m} a_{ik}\bar{x}_{ij}\bar{x}_{kj}<\sum_{j=1}^{n}\sum_{i=1}^{m}\sum_{k=1}^{m} a_{ik}x^0_{ij}x^0_{kj}.$$

If we calculate

$$\bar{y}_{ik}=\sum_{j=1}^{n} \bar{x}_{ij}\bar{x}_{kj}$$

as in (2), we have

$$\sum_{i=1}^{m-1} \sum_{k=i+1}^{m} a_{ik}\bar{y}_{ik} < \sum_{i=1}^{m-1} \sum_{k=i+1}^{m} a_{ik}y_{ik}^{0},$$

where the $\bar{y}_{ik}$ satisfy the fourth observation. If the $\bar{y}_{ik}$ also satisfy (3), we would have a contradiction since the y_{ik}^{0} produce the minimum value in (4); however, the right side of (3) is the smallest possible value for

$$\sum_{i=1}^{m-1} \sum_{k=i+1}^{m} y_{ik}$$

consonant with (2). Thus (3) is satisfied by the $\bar{y}_{ik}$ and the x_{ij}^{0} must be the global minimum in (1).

Care must be taken in computational work using the algorithm since storage does become a problem; however, the lists developed reduce in size as the algorithm progresses. Lists, beyond the first, with all marked elements may be discarded. Further, redundant information may develop in several lists and may be discarded; however, the number of lists grows rapidly and storage may be a problem even for moderate-sized problems.

We use as example the problem in [1]. Step 1 of the algorithm defines the problem:

(5)

	1	2	3	4	5
1		0*	2*	4	3
2			6	2*	3
3				5	3
4					3

where $m=5$, $n=3$, $d=1$, $m_0=2$, and $r=2$. The minimum value listed is zero, which is then marked; we form the list

(6)

1	3	4	5
1, 2	8	6	6
3		5	3*
4			3

The minimum unmarked value is 2 in (5); which is then marked; we form the list

(7)

1	2	4	5
1, 3	8	11	8
2			5
4		4	5

The minimum unmarked value is 2 in (5); which is then marked; we form the list

(8)

1	1	3	5
1		4	5
2, 4	6	13	8
3			5

The solution is apparent in (6) for the element 3. The objective function value is 3 (or 6 in the original problem) where $\bar{T}_1=(1, 2)$, $\bar{T}_2=(3, 5)$, $\bar{T}_3=4$. Thus we have

(9)
$$X=\begin{vmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{vmatrix}.$$

Note an alternate solution in (6) with $\bar{T}_1=(1, 2)$, $\bar{T}_2=3$, $\bar{T}_3=(4, 5)$.

REFERENCE

[1] Carlson, R. C., and G. L. Nemhauser, "Scheduling to Minimize Interaction Cost," Operations Research, **14**, 52–58 (1966).

* * *