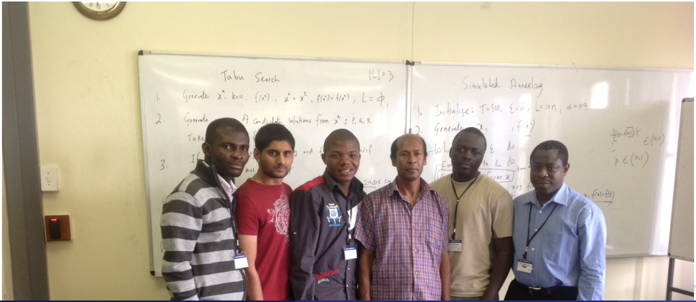
The Quadratic Assignment Problem

Joel L. O. ; Lalla V. ; Mbongo J. ; Ali M. M. ; Tuyishimire E.; Sawyerr B. A.
Supervisor
Suriyakat W. (Not in the picture)



Introduction

- The Quadratic assignment problem (QAP) is one of the fundamental, interesting and challenging combinatorial optimization problems from the category of the facilities location/allocation problems.
- QAP considers the problem of allocating a set of n facilities to a set of n locations, with the cost being a function of the distance d_{kl} between the locations k and l and flow f_{ij} between facilities i and j, plus costs b_{ik} associated with a facility i being placed at a certain location k

Possible Applications of QAP

- Hospital Layout
- Dartboard Design
- Steinberg Wiring Problem
- Typewriter Keyboard Design
- Scheduling
- Production Line etc

Mathematical Model

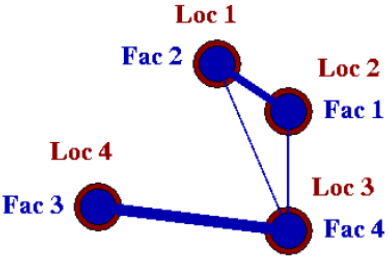


Figure 1:

$$\phi : \{1, 2, \cdots, n\} \to \{1, 2, \cdots, n\}; \quad i \to \phi(i)$$

$$\min_{\phi \in \mathcal{S}_n} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{\phi(i)\phi(j)} + \sum_{i=1}^{n} b_{i\phi(i)}. \tag{1}$$

Mathematical Model

$$\phi \rightarrow X \in \mathcal{X}_n, \; X = (x_{ij})$$

$$x_{ij} = \begin{cases} 1 & \text{if facility } i \text{ is placed at location } j, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The Quadratic Integer Programming formulation

$$(QIP) \qquad \min \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ij} d_{kl} x_{ik} x_{jl} + \sum_{i=1}^{n}\sum_{k=1}^{n} b_{ik} x_{ik}, \qquad s.t. \tag{3}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \text{for } j = 1, \ldots, n, \tag{4}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \text{for } i = 1, \ldots, n, \tag{5}$$

$$x_{ij} \in \{0, 1\} \qquad \text{for } i, j = 1, \ldots, n, \tag{6}$$

The Trace formulation

$$(TF) \qquad \min \qquad tr\left[(FXD^T + B)X^T\right], \qquad s.t. \tag{7}$$

$$X^T e = e, \tag{8}$$

$$Xe = e, \tag{9}$$

$$x_{ij} \in \{0, 1\} \text{ for all } i, j, \tag{10}$$

where e is the column n–vector of ones.

The Kronecker formulation

Given an $n \times n$-matrix X, we define $x = vec(X)$ to be the n^2-vector formed by the columns of X. The QAP can thus be formulated as:

$$(KF) \qquad \min \qquad x^T(F \otimes D)x + b^T x, \qquad s.t. \tag{11}$$

$$X^T e = e, \tag{12}$$

$$Xe = e, \tag{13}$$

$$x_i \in \{0,1\} \text{ for all } i = 1, \ldots, n^2, \tag{14}$$

where $x = vec(X)$ and $b = vec(B)$.

The Standard Quadratic Integer Programming formulation

$$(SQIP) \qquad \min \qquad x^T S x, \qquad s.t. \tag{15}$$

$$Ax = b, \tag{16}$$

$$0 \leq x \leq e, \tag{17}$$

$$x \in \mathbb{Z}^{n^2}. \tag{18}$$

$C_{ijkl} = f_{ij} d_{kl}$ if $i \neq j$ or $k \neq l$ and $C_{iikk} = f_{ii} d_{kk} + b_{ik}$. We define an $n^2 \times n^2$-matrix S in such a way that the element C_{ijkl} of the matrix C is on the row $(i-1)n + k$ and column $(j-1)n + l$ of S and let $x = vec(X)$.

- The formulation of the QAP in standard form as given in equation 15 makes QAP simpler for exact methods to solve
- In this study, Simulated Annealing was used to solve equation 1

Hospital Layout

- In this problem instance, we are concerned with locating 19 facilities in 19 given locations in an Outpatient department of an hospital in Cairo, Egypt [2].
- The yearly flow f_{ij} between each pair of facilities i and $j$is known and so the distance d_{kl} between each pair of locations k and l. Each location can house only one facility and each facility occupies only one location.
- The objective is to locate the facilities so as to minimize the total distance travelled by patients per year.
- The problem is formulated as in equation 15.

Hospital Layout Data

Distance Matrix (d) for the Quadratic Assignment Problem Hospital Layout

0	12	36	28	52	44	110	126	94	63	130	102	65	98	132	132	126	120	126
12	0	24	75	82	75	108	70	124	86	93	106	58	124	161	161	70	64	70
36	24	0	47	71	47	110	73	126	71	95	110	46	127	163	163	73	67	73
28	75	47	0	42	34	148	111	160	52	94	148	49	117	104	109	111	105	111
52	82	71	42	0	42	125	136	102	22	73	125	32	94	130	130	136	130	136
44	75	47	34	42	0	148	111	162	52	96	148	49	117	152	152	111	105	111
110	108	110	148	125	148	0	46	46	136	47	30	108	51	79	79	46	47	41
126	70	73	111	136	111	46	0	69	141	63	46	119	68	121	121	27	24	36
94	124	126	160	102	162	46	69	0	102	34	45	84	23	80	80	69	64	51
63	86	71	52	22	52	136	141	102	0	64	118	29	95	131	131	141	135	141
130	93	95	94	73	96	47	63	34	64	0	47	56	54	94	94	63	46	24
102	106	110	148	125	148	30	46	45	118	47	0	100	51	89	89	46	40	36
65	58	46	49	32	49	108	119	84	29	56	100	0	77	113	113	119	113	119
98	124	127	117	94	117	51	68	23	95	54	51	77	0	79	79	68	62	51
132	161	163	104	130	152	79	121	80	131	94	89	113	79	0	10	113	107	119
132	161	163	109	130	152	79	121	80	131	94	89	113	79	10	0	113	107	119
126	70	73	111	136	111	46	27	69	141	63	46	119	68	113	113	0	6	24
120	64	67	105	130	105	47	24	64	135	46	40	113	62	107	107	6	0	12
126	70	73	111	136	111	41	36	51	141	24	36	119	51	119	119	24	12	0

Flow Matrix (f) for the Quadratic Assignment Problem Hospital Layout

0	76687	0	415	545	819	135	1368	819	5630	0	3432	9082	1503	0	0	13732	1368	1783
76687	0	40951	4118	5767	2055	1917	2746	1097	5712	0	0	0	268	0	1373	268	0	0
0	40951	0	3848	2524	3213	2072	4225	566	0	0	404	9372	0	972	0	13538	1368	0
415	4118	3848	0	256	0	0	0	0	829	128	0	0	0	0	0	0	0	0
545	5767	2524	256	0	0	0	0	47	1655	287	0	42	0	0	0	226	0	0
819	2055	3213	0	0	0	0	0	0	926	161	0	0	0	0	0	0	0	0
135	1917	2072	0	0	0	0	0	196	1538	196	0	0	0	0	0	0	0	0
1368	2746	4225	0	0	0	0	0	0	0	301	0	0	0	0	0	0	0	0
819	1097	566	0	47	0	196	0	0	1954	418	0	0	0	0	0	0	0	0
5630	5712	0	829	1655	926	1538	0	1954	0	0	282	0	0	0	0	0	0	0
0	0	0	128	287	161	196	301	418	0	0	1686	0	0	0	0	226	0	0
3432	0	404	0	0	0	0	0	0	282	1686	0	0	0	0	0	0	0	0
9082	0	9372	0	42	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1503	268	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	972	0	0	0	0	0	0	0	0	0	0	0	0	99999	0	0	0
0	1373	0	0	0	0	0	0	0	0	0	0	0	0	99999	0	0	0	0
13732	268	13538	0	226	0	0	0	0	0	226	0	0	0	0	0	0	0	0
1368	0	1368	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1783	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Best Solution in Literature

Solution Obtained in Literature

The Best Solution obtained in [2] is 11281887 with the Permutation

[14 9 11 10 12 5 3 8 4 13 2 1 7 16 18 17 19 6 15]

And it took 136 secs

Why Heuristics?

- There are three main exact solution methods used to find the global optimal solution for a given QAP: **Dynamic programming**, **Cutting plane techniques**, and **Branch and Bound procedures**.
- Due to the complexity of the QAP, problems of larger dimensions become intractable for these methods .
- Heuristics are needed to handle the ever-increasing complexity of planning issues.
- These heuristics, while not providing the global optimal solution, can produce good answers within reasonable time constraints.

Simulated Annealing Introduction(SA)

- An iterative improvement algorithm
- It follows the idea for obtaining fine crystals in a solid metals with low energy:
  - First melt the solid by increasing the temperature
  - Then, slowly cool it so it crystallizes into a perfect lattice
- Simulated annealing was developed in 1983 to deal with highly nonlinear problems.
- It has been proved from literature that by carefully controlling the rate of cooling of the temperature, SA can find the global optimum

Simulated Annealing

Algorithm 1 Simulated Annealing pseudo-code

$Initialize: T, \epsilon, L, \alpha$
$Generatex, f(x)$
while $T > \epsilon$ **do**
    for $i = 1toL$ **do**
        generate y from x using the methods enumerated above.
        if $f(y) < f(x)$ **then**
            $x = y, f(y) = f(x)$
        else
            if $\exp -(f(y) - f(x))/T > \rho \in 0, 1$ **then**
                $x = y, f(y) = f(x)$
            end if
        end if
    end for
    $T = alpha * T$
end while

Improvement Method 1

Algorithm 2 Swapping Search Method

generate two indices I and J between 1 and N
$I = round(N * random)$
$J = round(N * random)$
while $I = J$ **do**
    $I = round(N * random)$
end while

Swapping Search Method Example

x = (4 2 3 5 1)
if I = 2 and J = 4 then
y = (4 5 3 2 1)

Improvement Method 2

Algorithm 3 Inverse Search Method

1: generate two indices I and J between 1 and N

2: $I = round(N * random)$

3: $J = round(N * random)$

4: **while** $I = J$ **do**

5: $\quad I = round(N * random)$

6: **end while**

7: copy the values within the two indices and reverse it.

Inverse Search Method Example

x = (4 2 3 5 1)
if I = 2 and J = 5 then
y = (4 1 5 3 2)

Improvement Method 3

Algorithm 4 Translation Search Method

generate two indices I and J between 1 and N $(I, J \neq 1andN)$
$I = round(N * random)$
$J = round(N * random)$
Also generate an index K
$K = round(N * random)$
while $I = J$ **do**
    $I = round(N * random)$
end while
copy the values within the two indices and paste it after the index K

Translation Search Method Example

x = (4 2 3 5 1)
if I = 2, J = 4 and K = 5 then
y = (4 1 5 3 2)

Experimental Parameters and Machine Configuration

Experimental Parameters

$T = 500, \epsilon = 10^-3, L = 10 * n, \alpha = 0.9$

Machine Configuration

- HP AMD Turion II UltraDual Core Mobile M620 2.5Ghz
- 4GB RAM (2.75GB Usable)
- 32-bit Operating System
- Windows 7 professional
- MATLAB 2010a

Experimental Results: Table 1: Solutions of the First run by SA

Iteration #	Location at which facilities are assigned																			Cost
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
0	5	10	13	7	14	2	15	6	3	19	1	11	18	9	17	16	4	12	8	24781836
1	5	10	13	7	1	2	15	6	3	19	14	11	18	9	17	16	4	12	8	24613469
10	5	10	13	7	1	2	16	6	9	19	14	11	18	3	17	15	4	12	8	24539470
20	5	10	13	16	1	2	7	6	9	12	14	11	18	3	17	15	4	19	8	24274831
30	5	10	13	16	1	2	7	6	19	12	14	11	18	3	17	9	4	15	8	19903723
50	5	10	13	16	1	3	7	9	19	14	18	11	6	2	17	12	4	15	8	16246610
80	5	10	13	9	1	11	17	3	19	14	18	8	6	2	7	12	4	15	16	14239333
100	5	10	13	9	1	11	2	3	19	14	18	8	6	17	7	12	4	15	16	14202630
120	5	10	13	9	1	14	2	3	19	11	18	8	6	17	7	12	4	16	15	14018514
130	5	10	13	9	1	14	2	3	19	11	18	8	6	17	7	12	4	16	15	14018514
140	5	10	13	9	1	14	2	3	19	11	18	8	6	17	7	12	4	16	15	14018514
150	5	10	13	9	1	14	2	3	19	11	18	8	6	17	7	12	4	16	15	14018514
200	5	10	13	9	3	14	2	1	19	11	18	17	6	8	7	12	4	16	15	13981954
300	5	10	13	2	3	14	9	1	19	11	17	18	4	15	7	12	6	16	8	13807099
393	5	10	13	2	3	14	9	1	19	11	17	18	4	15	7	12	6	16	8	13807099
																			Time Elapsed in Seconds	**2.83**

Graph 1

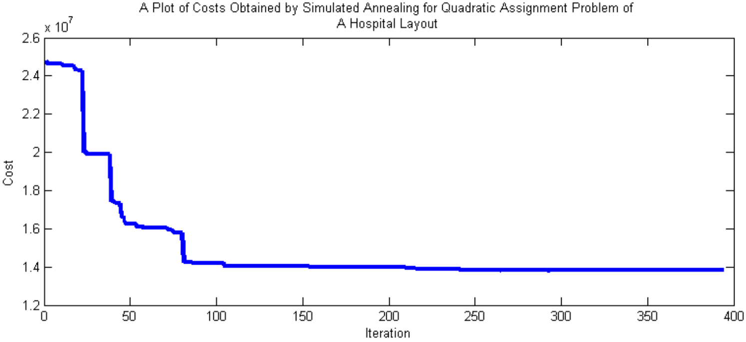


Figure 3: Plot of cost vs iteration for the first run

Table 2: Solutions of the Second run by SA

Iteration #	Location at which facilities are assigned																			Cost
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
0	6	3	16	11	7	17	14	8	5	19	15	1	2	4	18	13	9	10	12	33826199
1	6	3	16	11	7	17	14	8	5	19	15	1	12	4	18	13	9	10	2	33680966
10	2	3	16	11	7	17	14	8	5	19	15	1	13	12	18	4	10	9	6	29927469
20	2	3	11	16	4	17	9	8	5	19	6	1	13	12	18	7	10	14	15	19468578
30	2	3	11	6	4	17	9	18	5	19	16	1	13	12	8	7	10	14	15	18964657
50	2	3	10	6	18	17	9	4	14	19	15	1	13	12	8	7	11	5	16	17951520
100	2	3	10	6	1	9	17	4	11	18	15	19	13	16	8	7	5	14	12	17172203
200	2	3	13	6	5	11	8	10	7	18	15	16	1	9	19	17	4	14	12	14039736
300	2	3	6	13	10	11	8	5	7	19	15	16	1	9	18	17	4	14	12	12245727
400	2	1	6	10	13	11	8	5	12	19	16	15	4	9	17	18	3	14	7	11651627
500	2	1	4	10	13	11	9	5	12	19	16	15	6	8	17	18	3	14	7	10960319
1000	2	1	4	5	10	15	14	6	9	12	11	19	13	7	17	18	3	16	8	10597815
5000	2	1	4	10	5	15	14	6	12	9	11	19	13	7	17	18	3	16	8	10556127
10000	2	1	4	10	5	15	14	6	12	9	11	19	13	7	17	18	3	16	8	10556127
23750	2	1	4	10	5	15	14	6	12	9	11	19	13	7	17	18	3	16	8	10556127
																			Time Elapsed	151.93

[]

Graph 2

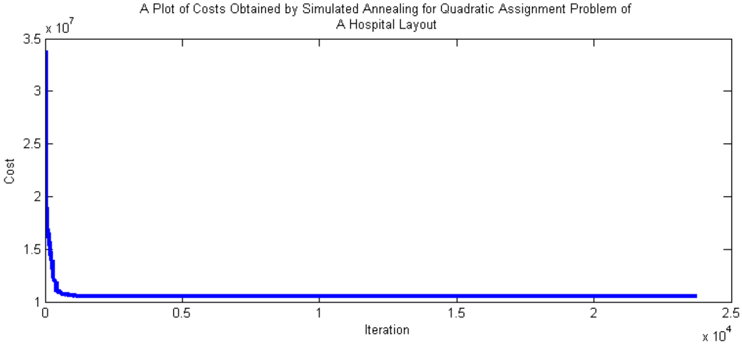


Figure 4: Plot of cost vs iteration for the second run

Table 3: Solutions of the Third run by SA

Iteration #	Location at which facilities are assigned																			Cost
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
0	6	3	16	11	7	17	14	8	5	19	15	1	2	4	18	13	9	10	12	33826199
1	6	3	16	11	7	17	14	8	5	19	15	1	12	4	18	13	9	10	2	33680966
10	2	3	16	11	7	17	14	8	5	19	15	1	13	12	18	4	10	9	6	29927469
20	2	3	11	16	4	17	9	8	5	19	6	1	13	12	18	7	10	14	15	19468578
30	2	3	11	6	4	17	9	18	5	19	16	1	13	12	8	7	10	14	15	18964657
50	2	3	10	6	18	17	9	4	14	19	15	1	13	12	8	7	11	5	16	17951520
100	2	3	10	6	1	9	17	4	11	18	15	19	13	16	8	7	5	14	12	17172203
200	2	3	13	6	5	11	8	10	7	18	15	16	1	9	19	17	4	14	12	14039736
300	2	3	6	13	10	11	8	5	7	19	15	16	1	9	18	17	4	14	12	12245727
400	2	1	6	10	13	11	8	5	12	19	16	15	4	9	17	18	3	14	7	11651627
500	2	1	4	10	13	11	9	5	12	19	16	15	6	8	17	18	3	14	7	10960319
1000	2	1	4	5	10	15	14	6	9	12	11	19	13	7	17	18	3	16	8	10597815
1200	2	1	4	10	5	15	14	6	12	9	11	19	13	7	17	18	3	16	8	10556127
1300	2	1	4	10	5	15	14	6	12	9	11	19	13	7	17	18	3	16	8	10556127
1330	2	1	4	10	5	15	14	6	12	9	11	19	13	7	17	18	3	16	8	10556127
																			Time Elasped in Seconds	**8.61**

Graph 3

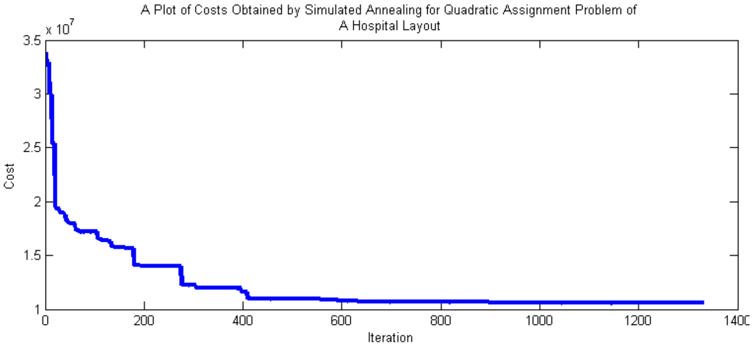


Figure 5: Plot of cost vs iteration for the third run

Conclusion

- We designed and implemented the Simulated Annealing heuristic Algorithm to solve any instance of QAP
- The numerical results obtained shows a superiority of the above heuristic over the heuristic used in [2]
- The result obtained is 6.4% better than the best solution in [2] and 24.4% better than the original layout.

Further Work

1. To implement two other methods that we have designed
2. To implement Tabu Search (TS) algorithm so as to evaluate its results with SA solutions
3. To use other stochastic algorithms, for example Genetic algorithm (GA), Particle Swarm Optimization, etc

References

C.W. COMMANDER.
A survey of the quadratic assignment problem, with applications.
PhD thesis, UNIVERSITY OF FLORIDA, 2003.

Alwalid N. Elshafei.
Hospital layout as a quadratic assignment problem.
Journal of The Operational Research Society, 28:167–179, 1977.

Tjalling C. Koopmans and Martin J. Beckmann.
Assignment problems and the location of economic activities.
Cowles Foundation Discussion Papers 4, Cowles Foundation for Research in Economics, Yale University, 1955.

Alfonsas Miseviius.
A modified simulated annealing algorithm for the quadratic assignment problem, 2003.

Thank You!!!