

# Package ‘qap’

October 5, 2015

**Title** Heuristics for the Quadratic Assignment Problem (QAP)

**Version** 0.1-0

**Date** 2015-10-05

**Description** Implements heuristics for the Quadratic Assignment Problem (QAP). Currently only a simulated annealing heuristic is available.

**License** GPL-3

**NeedsCompilation** yes

**Author** Michael Hahsler [aut, cre, cph],  
Franz Rendl [ctb, cph]

**Maintainer** Michael Hahsler <mhahsler@lyle.smu.edu>

**Repository** CRAN

**Date/Publication** 2015-10-05 22:14:30

## R topics documented:

qap . . . . .	1
read_qaplib . . . . .	3

<b>Index</b>	<b>5</b>
--------------	----------

---

qap *Solve Quadratic Assignment Problems (QAP)*

---

## Description

This function implements Quadratic Assignment Problems (QAP) heuristics. Currently there is only a simulated annealing heuristic available, but more will be added in the future.

## Usage

```
qap(A, B, method = NULL, ...)  
qap.obj(A, B, o)
```

**Arguments**

A	a symmetric matrix with positive weights/flows between pairs facilities.
B	a symmetric matrix with positive distances between pairs of locations.
method	a character string indicating the used solver. Defaults to "SA", the currently only available method.
...	further arguments are passed on to the solver (see details).
o	a permutation vector for the assignment of facilities to locations.

**Details**

The objective of the QAP is to find the best facility to location assignment. The assignment is represented by a permutation matrix  $X$  and the objective is

$$\min_{X \in \Pi} \text{tr}(AXBX^T)$$

gap.obj calculates the objective function for A and B with the permutation o.

The QAP is known to be NP-hard. This function implements the simple simulated annealing heuristic described by Burkard and Rendl (1984). The code is based on Rendl's FORTRAN implementation of the algorithm available at the QAPLIB Web site.

The solver has the additional arguments `rep = 1L`, `miter = 2 * nrow(A)`, `fiter = 1.1`, `ft = 0.5` and `maxsteps = 50L`

**rep** integer; number of restarts.

**miter** integer; number of iterations at fixed temperature.

**fiter** multiplication factor for miter after miter random transposition trials.

**ft** multiplication factor for t after miter random transposition trials (between 0 and 1).

**maxsteps** integer; maximal number of allowed cooling steps.

**Value**

Returns an integer vector with facility to location assignments. The objective function value is provided as attribute "obj".

**Author(s)**

Michael Hahsler

**References**

R.E. Burkard and F. Rendl. A thermodynamically motivated simulation procedure for combinatorial optimization problems. *European Journal of Operations Research*, 17(2):169-174, 1984.

R.E. Burkard, E. Cela, S.E. Karisch and F. Rendl, QAPLIB - A Quadratic Assignment Problem Library, <http://anjos.mgi.polytml.ca/qaplib/>

**Examples**

```
## load the had12 QAPLIB problem
p <- read_qaplib(system.file("qaplib", "had12.dat", package="qap"))
p

## run 1 repetitions verbose
a <- qap(p$A, p$B, verbose = TRUE)
a

## compare with known optimum (gap, % above optimum)
(attr(a, "obj") - p$opt)/p$opt * 100

## run more repetitions quietly
a <- qap(p$A, p$B, rep = 100)
a

## compare with known optimum (gap, % above optimum)
(attr(a, "obj") - p$opt)/p$opt * 100
```

---

read\_qaplib

*Read QAPLIB Files*

---

**Description**

Reads example file in the format used by QAPLIB.

**Usage**

```
read_qaplib(file)
```

**Arguments**

file            file name.

**Details**

Problems end with the extension `.dat` and solutions with `.sln`. The code tries to read the problem and, if available in the same directory, it also reads the solution and the known optimal value from the solution file.

The package contains a copy of the problem instances and solutions from QAPLIB. The data is stored in the package in directory `qaplib`.

**Value**

Returns a list with the components

D	distance matrix.
W	weight matrix.
solution	a known optimal solution (if available).
opt	known optimal value (if available).

**Author(s)**

Michael Hahsler

**References**

R.E. Burkard, E. Cela, S.E. Karisch and F. Rendl, QAPLIB - A Quadratic Assignment Problem Library, <http://anjos.mgi.polymtl.ca/qaplib/>

**Examples**

```
## load a QAPLIB problem instance
p <- read_qaplib(system.file("qaplib", "had12.dat", package="qap"))
p

## list all QAPLIB instances
dir(system.file("qaplib", package="qap"), pattern = "*.dat")
```

# Index

\*Topic **manip**  
  read\_qaplib, 3  
\*Topic **optim**  
  qap, 1  
  
qap, 1  
  
read\_qaplib, 3