Package 'qap'

October 1, 2026

Title Heuristics for the Quadratic Assignment Problem (QAP)

Version 0.1-3

Date 2026-09-30

Description Implements a simulated annealing heuristic for the Quadratic Assignment Problem (QAP). Originally formulated as a facility location problem in operations research, the QAP also has applications in data analysis. The problem is NP-hard.

Suggests testthat

URL https://github.com/mhahsler/qap, https://michael.hahsler.net/qap/

BugReports https://github.com/mhahsler/qap/issues

License GPL-3

Encoding UTF-8

Config/roxygen2/version 8.1.0

NeedsCompilation yes

Author Michael Hahsler [aut, cre, cph] (ORCID: <https://orcid.org/0000-0003-2716-1405>), Franz Rendl [ctb, cph]

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

Repository CRAN

Date/Publication 2026-10-01 16:00:02 UTC

Contents

qap 2
read_qaplib 3

Index **5**

qap *Solve a Quadratic Assignment Problem*

Description

Solve a quadratic assignment problem (QAP) with a simulated annealing heuristic. `qap.obj()` calculates the objective value of an assignment.

Usage

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

qap.obj(A, B, o)
```

Arguments

`A`	A numeric matrix of flows between facilities. For `qap()`, it must be at least 2 by 2, symmetric, nonnegative, and finite.
`B`	A numeric matrix of distances between locations. For `qap()`, it must have the same dimensions as `A` and be symmetric, nonnegative, and finite.
`method`	Solver name. Currently only `"SA"` is available.
`...`	Additional arguments passed to the simulated annealing solver. See Details.
`o`	A permutation of `1:nrow(A)` assigning facilities to locations.

Details

The problem is to assign n facilities to n locations to minimize total transportation or interaction costs. Required flows between the facilities are represented by the matrix A and distances between locations is given in matrix B. The QAP seeks an assignment that minimizes the sum of flows times distance. For an assignment represented by a $n \times n$ permutation matrix X used to assign the facilities to the locations in the order given by the permutation, the objective can be written as

$$\min_{X \in \Pi} \operatorname{tr}(AXB^T X^T)$$

where Π is the set of all valid $n \times n$ permutation matrices. Note that for symmetric distances, the distance matrix B does not need to be transposed.

The QAP originated as a facility location problem (Koopmans and Beckmann, 1957) and also has applications in data analysis (Hubert and Schultz, 1976). It is NP-hard. The solver uses the simulated annealing heuristic of Burkard and Rendl (1984), based on Rendl's Fortran implementation from QAPLIB. The authors suggest it can produce heuristic solutions for problems with up to 256 objects, although the implementation does not enforce this limit.

Additional solver arguments are:

`rep` Positive integer number of restarts; default `1L`.

`miter` Positive integer number of iterations at a fixed temperature; default `2 * nrow(A)`.

`fiter` Factor of at least 1 by which `miter` grows after each cooling step; default `1.1`.

`ft` Factor by which the temperature decreases after each cooling step; default `0.5` (strictly between 0 and 1).

`maxsteps` Positive integer maximum number of cooling steps; default `50L`.

`verbose` Print progress; default `FALSE`.

Value

`qap()` returns an integer vector of facility to location assignments with the objective value in its `"obj"` attribute. `qap.obj()` returns the objective value for permutation `o`.

References

Burkard, R. E. and Rendl, F. (1984). A thermodynamically motivated simulation procedure for combinatorial optimization problems. *European Journal of Operational Research*, 17(2), 169-174. doi:10.1016/03772217(84)902315

Koopmans, T. C. and Beckmann, M. (1957). Assignment problems and the location of economic activities. *Econometrica*, 25(1), 53-76. doi:10.2307/1907742

Hubert, L. and Schultz, J. (1976). Quadratic assignment as a general data analysis strategy. *British Journal of Mathematical and Statistical Psychology*, 29(2), 190-241. doi:10.1111/j.20448317.1976.tb00714.x

See Also

`read_qaplib()`

Examples

```
p <- read_qaplib(system.file("qaplib", "had12.dat", package = "qap"))
a <- qap(p$A, p$B, verbose = TRUE)
a
qap.obj(p$A, p$B, a)
(attr(a, "obj") - p$opt) / p$opt * 100
```

`read_qaplib` *Read QAPLIB Files*

Description

Read a problem instance and, when available, its solution from QAPLIB files.

Usage

```
read_qaplib(file)
```

Arguments

`file` Path to a QAPLIB problem file with a `.dat` extension.

Details

If a `.sln` file with the same base name exists in the same directory, the function also reads its solution and objective value. Zero-based solutions are converted to R's one-based indexing. The package includes QAPLIB instances and solutions in its `qaplib` directory.

Value

A list with `A` (the flow matrix), `B` (the distance matrix), `solution` (a known solution, if available), and `opt` (its objective value, if available). The last two components are `NULL` when no solution file exists.

References

Burkard, R. E., Çela, E., Karisch, S. E., and Rendl, F. QAPLIB: A Quadratic Assignment Problem Library.

Examples

```
p <- read_qaplib(system.file("qaplib", "had12.dat", package = "qap"))
p
dir(system.file("qaplib", package = "qap"), pattern = "\\.dat$")
```

Index

qap, 2
qap.obj(), 2

read_qaplib, 3
read_qaplib(), 3