Remark on Algorithm 769: Fortran Subroutines for Approximate Solution of Sparse Quadratic Assignment Problems Using GRASP

TIM HOPKINS
University of Kent

We present a number of corrections and improvements to Algorithm 769 [Pardalos et al. 1997].

Categories and Subject Descriptors: G.1.6 [**Numerical Analysis**]: Optimization, Integer programming; G.2.1 [**Discrete Mathematics**]: Combinatorics, Combinatorial algorithms; G.4 [**Mathematics of Computing**]: Mathematical Software—*Reliability and robustness*

General Terms: Algorithms, Performance, Reliability

This remark details a number of source changes that are required to the GRASP package (Algorithm 769 [Pardalos et al. 1997])

(1) to ensure that the software adheres to the Fortran 77 and Fortran 95 standards [ANSI 1979; ISO/IEC 1997],

(2) to improve the portability of the code, and

(3) to remove a number of extraneous dummy arguments from subroutine definitions.

In the routine `insrtq` the dummy arguments `v` and `iv` are assigned to. This contravenes the standard in that the actual arguments used in calls from `strcst` are an expression (`-f(index)`, just before the statement labelled 10) and a do-loop control variable (`i`, in the `do 40` loop). The statements that need changing in `insrtq` are

```
     v = q(sq)                              tempv = q(sq)
    iv = iq(sq)                            tempiv = iq(sq)

    if (q(tsz) .gt. v) then                if (q(tsz) .gt. tempv) then

     q(sq) = v                              q(sq) = tempv
    iq(sq) = iv                            iq(sq) = tempiv
```

Author's address: Computing Laboratory, University of Kent, Canterbury, Kent, CT2 7NF, UK.

The new local variables tempv and tempiv must also be declared integer.

In the routine stage2 the array arguments a and b should both be dimensioned n and not n2. In fact the variable n2 may be removed completely from the argument lists of the routines gqaps, srtcst, stage1, stage2, local and evalij and replaced by n in the routines insrtq and removq. This variable is only used to dimension array arguments; all dimension statements of the form array(n2) need to be replaced by array(n*n).

The argument optb is unused in gqaps and should be removed as should j from the argument lists of stage1 and stage2. In the case of stage1 it is still needed as a local variable.

The routine requires the generation of random integers in a given range. In the original code this is implemented using the seed value to Schrage's portable generator [Schrage 1979]. This has two drawbacks; first it restricts the generator being used and second it involves a constant $(2^{31} - 1)$ being embedded in the code. Both these problems may be removed by defining a small function ranint of the form

```
      integer function ranint(maxval)
      integer maxval
      real rand
c
c  return a random integer in the range
c  1 ... maxval
c
      ranint = int(maxval*rand()) + 1
      end
```

and replacing

(1) the three executable statements following label 10 in stage1 by

```
          nselct = ranint(int(alpha*beta*(n*n - n)))
```

(2) the two statements following the block-IF after label 60 in stage2 by

```
          nselct = ranint(scp)
```

(3) the three executable statements following label 100 in stage2 by

```
          nselct = ranint(int(alpha*sizec))
```

Declaration statements for rand and xrand should be replaced by

```
      integer ranint
```

in both stage1 and stage2.

In addition the variable high may be removed completely from stage2 by replacing the only remaining reference

```
60    high = scp
```

by

```
60    continue
```

Finally, the constant, `infty`, may be replaced by a call either to `i1mach(9)` (using the Port library routine [Fox et al. 1978]) or, if a Fortran 90 compiler is available, to `huge(0)`.

REFERENCES

ANSI. 1979. *Programming Language Fortran X3.9-1978.* American National Standards Institute, New York.

Fox, P. A., Hall, A. D., and Schryer, N. L. 1978. Framework for a portable library. *ACM Trans. Math. Softw. 4,* 2 (June), 177–188.

ISO/IEC. 1997. *Information Technology – Programming Languages – Fortran - Part 1: Base Language (ISO/IEC 1539-1:1997).* ISO/IEC Copyright Office, Geneva.

Pardalos, P. M., Pitsoulis, L. S., and Resende, M. G. C. 1997. Algorithm 769: Fortran subroutines for approximate solution of sparse quadratic assignment problems using GRASP. *ACM Transactions on Mathematical Software 23,* 2 (June), 196–208.

Schrage, L. 1979. A more portable fortran random number generator. *ACM Transactions on Mathematical Software 5,* 2, 132–138.