



//minimisation problem

Initialization:

MaxIter = 10;

EstQuality = 8 //assume the best possible distance is 8

iter = 1;

Initial solution construction phase:

//Generate an initial solution at random, Sol

//Note that a solution is represented as a sequence of cities (path)

Sol = C-B-A-D-E-C

$f(\text{Sol}) = 6+1+4+1+5 = 17$

Water level, LEVEL = $f(\text{Sol}) = 17$

Solbest = Sol = C-B-A-D-E-C

$f(\text{Solbest}) = f(\text{Sol}) = 17$

$\text{UP} = (\text{LEVEL} - \text{EstQuality}) / \text{MaxIter} = (17-8)/10 = 0.9$

Improvement phase:

//in the do while loop

do while (iter <=MaxIter)

~~//generate a new solution using a swap neighbourhood structure i.e., swap.~~

Assume we swap between cities A and C in the Sol = C-B-A-D-E-C.

First iteration

Sol* = A-B-C-D-E-A;

$f(\text{Sol}^*) = 1+6+4+1+3=15$

if $f(\text{Sol}^*) < f(\text{Solbest})$ // $15 < 17 \rightarrow \text{true}$

Solbest = Sol* = A-B-C-D-E-A

$f(\text{Solbest}) = f(\text{Sol}^*) = 15$

if $f(\text{Sol}^*) < \text{LEVEL}$ // $15 < 17 \rightarrow \text{true}$

Sol = Sol* = A-B-C-D-E-A

$f(\text{Sol}) = f(\text{Sol}^*) = 15$

$\text{LEVEL} = \text{LEVEL} - \text{UP} = 17 - 0.9 = 16.1$

Second iteration:

$Sol = A-B-C-D-E-A$

$Sol^* = A-B-D-C-E-A$ //swap between C and D

$f(Sol^*) = 1+2+5+5+3 = 16$

if $f(Sol^*) < f(Sol_{best})$ //16 < 15) -> false

....

if $f(Sol^*) < LEVEL$ // 16 < 16.1 -> true

$Sol = Sol^* = A-B-D-C-E-A$

$f(Sol) = f(Sol^*) = 16$

$LEVEL = LEVEL - UP = 16.1 - 0.9 = 15.2$

Third iteration:

Sol = A-B-D-C-E-A

Sol* = A-C-D-B-E-A

//swap between B and C

$f(\text{Sol}^*) = 7+4+2+5+3 = 21$

if $f(\text{Sol}^*) < f(\text{Solbest})$

// $21 < 15 \rightarrow \text{false}$

....

if $f(\text{Sol}^*) < \text{LEVEL}$

// $21 < 15.2 \rightarrow \text{false}$

...

else

Sol = Sol

//new solution is not accepted

//solution remains the same: A-B-D-C-E-A

LEVEL = LEVEL – UP

// $15.2 - 0.9 = 14.3$

