Simulated Annealing

The Simulated Annealing algorithm (see the SA Algorithm section) generated is based on the stopping criteria introduced in [1]. Perturbation is performed by 1-Swap operator (see Perturb function under SA Algorithm section), while discrete representation is selected to keep the solutions. The parameters T, L, ip, r, and fp is set respectively to 10000000, 200, 0.7, 0.9, 0.01 respecting the findings derived from [1].

Pseudocode of Simulated Annealing algorithm is given as follows:

```
X ← InitialSolution
Z ← f(X)
Incumbent ← Z
Initialize input parameters: T, L, ip, r, fp
NumberOfIterations ← 0
terct ← 0
while terct < 5 :
    j ← 0
    do L times:
        NumberOfIterations ← NumberOfIterations + 1
        X′ ← Perturb (X)      // remove an existing facility and add a new facility
        Z′ ← f (X′)
        Δ ← Z′ − Z
        if Δ ≤ 0:
            Incumbent ← Z′
            X ← X′
            BestSolution ← X′
            BestIteration ← NumberOfIterations
            j ← j + 1    //keep track number of accepted solution
        else Δ < 0:
            select a random variable m ~U(0,1)
            if e^(−Δ/T) > m:
                Z ← Z′
                X ← X′
                j ← j + 1
            end if
        end if
    end do
    if j/L ≤ fp:
        terct = terct + 1
    else:
        terct = 0
    end if
    if j/L > ip:
        T = T/2
    else:
        T = r · T
    end if
```

39. $\boldsymbol{end\ while}$

40. $\boldsymbol{Return}\ Incumbent, BestSolution, BestIteration$

Variable Neighborhood Search

Within Variable Neighborhood Search Algorithm (see VNS function) discrete solution representation is preferred, while 1-swap, 2-swap, and 3-swap shaking moves are used as it is asked. For local search, 1-swap move operator is selected in order to discover the neighbors of the shake-d solution. The stopping criteria is met when no better solution is found within 20 iterations.

Pseudocode of Variable Neighborhood Search algorithm is given as follows:

1. $K \leftarrow \{\, 1 - swap,\ 2 - swap,\ 3 - swap \,\}$ *//using 1-swap, 2-swap and 3-swap neigborhood structures set K is generated for shaking step*

2. $X \leftarrow InitialSolution$

3. $Z \leftarrow f(X)$

4. $Incumbent \leftarrow Z$

5. $NumberOfIterations \leftarrow 0$

6. $\boldsymbol{Repeat}$:

7. $NumberOfIterations \leftarrow NumberOfIterations + 1$

8. $k \leftarrow 0$

9. $\boldsymbol{while\ k} <= \ length(K)$:

10. $X' \leftarrow random\ solution\ of\ X\ using\ neighborhood\ structure\ k$ *//shaking step*

11. $Generate\ all\ neighbors\ of\ X'\ based\ on\ 1 - swap\ move$ *//local search: randomly select one facility from X' and add new facility*

12. $Z'' \leftarrow\ minimum\ objective\ value\ within\ neighborhood\ N(X')$

13. $X'' \leftarrow\ solution\ corresponds\ to\ the\ minimum\ objective\ value$

14. $\boldsymbol{if}\ Z'' < Incumbent$:

15. $Incumbent \leftarrow Z''$

16. $X \leftarrow X''$

17. $continue$

18. $\boldsymbol{else}$:

19. $k \leftarrow\ k + 1$

20. $\boldsymbol{end\ if}$

21. $\boldsymbol{end\ while}$

22. $\boldsymbol{if}\ (Incumbent - Z'') < 0\ within\ last\ 20\ neighbors$:

23. $\boldsymbol{stop}$

24. $\boldsymbol{Return}$ *X, Incumbent, NumberOfIterations*

[1] Liu, C., Kao, R., Wang, A., Solving Location-allocation Problems with Rectilinear Distances by Simulated Annealing, *J. Opl. Res. Soc*, vol. 45, no. 11, 1994, pp. 1304-1315.