SCHOOL OF BUSINESS

DEPARTMENT OF MANAGEMENT SCIENCE & TECHNOLOGY

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

ACADEMIC YEAR OF 2021-2022

LARGE SCALE OPTIMIZATION

AUTHORS:

Konstantinos Ninas (f2822108)

Orestis Loukopoulos (f2822104)

John Vaniotis (f2822101)

Stamatis Sideris (f2822113)

SUPERVISING PROFESSOR: Emmanouil Zachariadis

Problem Description

Consider a central repository (Node with id: 0) and a set of n = 300 customers (Nodes with id: 1,..,n=300). All nodes are in a square of side size-100. The transition time from node to node is equal to the Euclidean distance between two nodes. Each customer has a required service time and a profit. A fleet of 5 vehicles is located in the central warehouse. The vehicles start from the warehouse, serve the customers and return to the central warehouse. Each vehicle conducts one route. Each customer must be covered (not necessarily) by a single vehicle visit. In this case, the customer bestows their profit to the route. The total time of each route cannot exceed 150 time units. The scope of the problem is to design a 5-route schedule that maximizes the total profit.

Model Structure

The solution of the problem presented, was reached through object-oriented programming. Initially, several classes in Python were constructed. The base of the solution was the class 'Model', which contained all the customers, the distance between the customers and a profit per cost ratio matrix for all customers (it is calculated by dividing the profit with the service time and the distance between two nodes). The profit for each customer is a random number between 5 and 20, while the service time for each customer is a random number between 5 and 10 time units. Each node (customer) has a random longitude and latitude (numbers between 0-100), an ID number, a service time, a profit and an index that shows whether the customer has been serviced. The warehouse is located in the coordinates (50,50). Finally, a class has been created for the routes of the fleet of the vehicles. Each vehicle has a sequence of nodes (customers), starting from the warehouse, a capacity, a cost and a profit. Finally, to ensure consistent results, a random seed was used.

Greedy Solution

The first step for the construction of the greedy algorithm, was the creation of a 'Solver' class. The 'Solver' class contains information for all the nodes, the customers, the distance matrix, profit ratio, the time capacity constraint, an empty solution and a class that initializes a nearest neighbor solution algorithm. The algorithm inspects all the customers (nodes) that can fit (considering that the vehicle must return to the warehouse) in the open routes, selects the one with the greatest value in the profit matrix and stores them and their best route. Upon inspecting all the customers, the warehouse is inserted at the last position of each route and the cost is updated with the time spend to travel from the last customer of each route to the warehouse.

```
Route  1 : Profit is 172 Cost is 144.17116647513 42 Customers: [0, 299, 172, 69, 118, 27, 161, 182, 119, 276, 288, 0]
Route  2 : Profit is 157 Cost is 148.5135516163118 Customers: [0, 253, 294, 197, 267, 297, 211, 99, 128, 244, 213, 0]
Route  3 : Profit is 152 Cost is 145.63552613112614 Customers: [0, 89, 212, 239, 33, 290, 261, 144, 145, 120, 0]
Route  4 : Profit is 113 Cost is 148.67848834494959 Customers: [0, 279, 154, 67, 17, 292, 230, 134, 0]
Route  5 : Profit is 147 Cost is 148.82794190409263 Customers: [0, 249, 112, 257, 164, 41, 179, 75, 152, 105, 202, 0]
Solution Cost is: 735.8266744716145 Solution Profit is: 741
```

1. Greedy Algorithm Solution

Local Searches

Local searches are procedures that iteratively transition from one solution to another via specific modifications to the current solution structure. Four such procedures were constructed, each one performing different modifications to the current solution. For each procedure a clone solution of the initial greedy solution was stored to compare its results with those of the best neighbor solution. If the candidate solution proves to be better than the clone solution, it replaces the clone solution. The local search examines the neighborhood of the new clone solution until it reaches a local minimum (or maximum). Finally, for each local search, the total number of iterations, the type of local search performed, the new cost, the new profit, and the new routes are printed.

The first local search performed is 'Relocation'. The aim of this local search is to minimize the solution's total cost by relocating covered customers from their current position in the route to another (this is not limited to the route of the vehicle that currently serves the customer but all the possible routes that can feasibly serve him). It is observed that the relocation local search iterated 15 times, the profit remained constant (as expected) and the total cost dropped approximately by 29 (from 735 to 706) time units.

```
Solution Cost is: 735.8266744716145 Solution Profit is: 741
1 730.3358653807076
2 727.5663293953921
3 724.9293268588597
4 723.3744283735618
5 722.0266402264213
6 718.89475908412
7 717.7952061673077
8 716.7581933150393
9 715.2783983215467
10 711.1756023453561
11 710.3130438107962
12 709.1659187831764
13 708.9840966320646
14 706.6487199158199
15 706.4846370116155
Relocation   -   Total number of moves: 15 move cost is:  706.4846370116155 Profit is: 741
Route  1 : Profit is 172  Cost is 148.72850096760803 Customers: [0, 299, 27, 69, 118, 161, 182, 119, 276, 288, 239, 212, 0]
Route  2 : Profit is 157  Cost is 116.73810527062392 Customers: [0, 294, 197, 267, 297, 211, 99, 244, 253, 0]
Route  3 : Profit is 152  Cost is 145.87835216209442 Customers: [0, 89, 33, 290, 261, 144, 145, 120, 179, 41, 0]
Route  4 : Profit is 113  Cost is 148.21556229949456 Customers: [0, 279, 172, 67, 154, 17, 292, 230, 134, 0]
Route  5 : Profit is 147  Cost is 146.92411631179436 Customers: [0, 249, 112, 75, 152, 105, 213, 128, 257, 164, 202, 0]
Solution Cost is: 706.4846370116155 Solution Profit is: 741
```

2. Relocation Local Search

The second local search performed is 'Swap'. The aim of this local search is to minimize the solution's total cost by swapping the position of any pair of covered customers (in the same or different routes), while not violating the capacity constraint for each vehicle examined. It is noticed that only 3 iterations were performed with this local search, the profit remains constant (as expected) and the solution's total cost dropped approximately by 9 (from 735 to 726) time units.

```
Solution Cost is: 735.8266744716145 Solution Profit is: 741
1 730.3358653807076
2 727.8112979692385
3 726.2563994839406
Swap   -   Total number of moves: 3 move cost is:  726.2563994839406 Profit is: 741
Route  1 : Profit is 172  Cost is 144.1711664751342 Customers: [0, 299, 172, 69, 118, 27, 161, 182, 119, 276, 288, 0]
Route  2 : Profit is 157  Cost is 145.9889842048427 Customers: [0, 253, 294, 197, 267, 297, 211, 99, 128, 213, 244, 0]
Route  3 : Profit is 152  Cost is 145.63552613112614 Customers: [0, 89, 212, 239, 33, 290, 261, 144, 145, 120, 0]
Route  4 : Profit is 113  Cost is 143.1876792540426 Customers: [0, 279, 67, 154, 17, 292, 230, 134, 0]
Route  5 : Profit is 147  Cost is 147.27304341879483 Customers: [0, 249, 257, 112, 164, 41, 179, 75, 152, 105, 202, 0]
Solution Cost is: 726.2563994839406 Solution Profit is: 741
```

3. Swap Move Local Search

The third local search performed is the ‘Insertion’. The aim of this local search is to maximize the profit per cost ratio of the solution by examining all the possible insertion positions in the current solution for all uncovered customers. The customer selected is the one with the best profit to cost ratio that does not violate the capacity constraint of the route they will be inserted to. It is noticed that this local search method was unable to locate any better neighbor solution. As a result, the solution’s cost and profit remain constant.

```
Solution Cost is: 735.8266744716145 Solution Profit is: 741
Insertion   -   Total number of moves: 0 move cost is:  735.8266744716145 Profit is: 741
Route  1 : Profit is 172  Cost is 144.1711664751342 Customers: [0, 299, 172, 69, 118, 27, 161, 182, 119, 276, 288, 0]
Route  2 : Profit is 157  Cost is 148.5135516163118 Customers: [0, 253, 294, 197, 267, 297, 211, 99, 128, 244, 213, 0]
Route  3 : Profit is 152  Cost is 145.63552613112614 Customers: [0, 89, 212, 239, 33, 290, 261, 144, 145, 120, 0]
Route  4 : Profit is 113  Cost is 148.67848834494959 Customers: [0, 279, 154, 67, 17, 292, 230, 134, 0]
Route  5 : Profit is 147  Cost is 148.82794190409263 Customers: [0, 249, 112, 257, 164, 41, 179, 75, 152, 105, 202, 0]
Solution Cost is: 735.8266744716145 Solution Profit is: 741
```

4. Insertion Local Search

The final local search performed is the ‘Profitable Swap’. The aim of this local search is to maximize the profit per cost ratio of the solution by replacing any covered customer in the current solution with any uncovered customer. For each possible swap the time constraint for each vehicle is taken into consideration. It is observed that this local search also failed to locate any better neighbor solution. Hence, the solution’s cost and profit also remain unchanged.

```
Solution Cost is: 735.8266744716145 Solution Profit is: 741
Profitable Swap   -   Total number of moves: 0 move cost is:  735.8266744716145 Profit is: 741
Route  1 : Profit is 172  Cost is 144.1711664751342 Customers: [0, 299, 172, 69, 118, 27, 161, 182, 119, 276, 288, 0]
Route  2 : Profit is 157  Cost is 148.5135516163118 Customers: [0, 253, 294, 197, 267, 297, 211, 99, 128, 244, 213, 0]
Route  3 : Profit is 152  Cost is 145.63552613112614 Customers: [0, 89, 212, 239, 33, 290, 261, 144, 145, 120, 0]
Route  4 : Profit is 113  Cost is 148.67848834494959 Customers: [0, 279, 154, 67, 17, 292, 230, 134, 0]
Route  5 : Profit is 147  Cost is 148.82794190409263 Customers: [0, 249, 112, 257, 164, 41, 179, 75, 152, 105, 202, 0]
Solution Cost is: 735.8266744716145 Solution Profit is: 741
```

5. Profitable Swap Local Search

Variable Neighborhood Descent - VND

Variable Neighborhood Descent is an algorithmic framework for solving combinatorial optimization problems. The main idea is the systematic change of the move types (e.g., Relocation, Swap, Insertion, Profitable Swap) during the search in the solution space. VND in particular searches the solution space in a deterministic way. All possible permutations of the above move types were tested and the optimal permutation is the following:

1. Swap
2. Relocation
3. Insertion
4. Profitable Swap

This specific permutation of local searches leads to 23 applications of the mentioned move types that result to an increase of the solution’s total time by almost 4 time units and to an increase of the solution’s profit by 41 profit units. For exhibition purposes, an additional example of a VND application is provided that utilizes all the local searches that have been created (table 7).

```
Solution Cost is: 735.8266744716145 Solution Profit is: 741
Swap Move  -   Iteration number: 1 Solution profit: 741 Solution Cost: 730.3358653807076
Swap Move  -   Iteration number: 2 Solution profit: 741 Solution Cost: 727.8112979692385
Swap Move  -   Iteration number: 3 Solution profit: 741 Solution Cost: 726.2563994839406
Relocation Move  -   Iteration number: 4 Solution profit: 741 Solution Cost: 723.3744283735618
Relocation Move  -   Iteration number: 5 Solution profit: 741 Solution Cost: 722.0266402264213
Relocation Move  -   Iteration number: 6 Solution profit: 741 Solution Cost: 718.89475908412
Relocation Move  -   Iteration number: 7 Solution profit: 741 Solution Cost: 717.7952061673077
Relocation Move  -   Iteration number: 8 Solution profit: 741 Solution Cost: 716.7581933150393
Relocation Move  -   Iteration number: 9 Solution profit: 741 Solution Cost: 715.2783983215467
Relocation Move  -   Iteration number: 10 Solution profit: 741 Solution Cost: 711.1756023453561
Swap Move  -   Iteration number: 11 Solution profit: 741 Solution Cost: 709.9895464041726
Swap Move  -   Iteration number: 12 Solution profit: 741 Solution Cost: 709.1414559450701
Relocation Move  -   Iteration number: 13 Solution profit: 741 Solution Cost: 705.1064447842833
Relocation Move  -   Iteration number: 14 Solution profit: 741 Solution Cost: 704.9246226331716
Relocation Move  -   Iteration number: 15 Solution profit: 741 Solution Cost: 703.75138566233
Relocation Move  -   Iteration number: 16 Solution profit: 741 Solution Cost: 703.5873027581256
Relocation Move  -   Iteration number: 17 Solution profit: 741 Solution Cost: 698.0859343484306
Insertion Move  -   Iteration number: 18 Solution profit: 747 Solution Cost: 708.5023422134293
Swap Move  -   Iteration number: 19 Solution profit: 747 Solution Cost: 705.096601706549
Insertion Move  -   Iteration number: 20 Solution profit: 764 Solution Cost: 720.2230546565519
Relocation Move  -   Iteration number: 21 Solution profit: 764 Solution Cost: 715.516229938734
Insertion Move  -   Iteration number: 22 Solution profit: 777 Solution Cost: 728.8791023583127
Insertion Move  -   Iteration number: 23 Solution profit: 782 Solution Cost: 739.9905132654976
Route  1 : Profit is 172  Cost is 148.72850096760803 Customers: [0, 299, 27, 69, 118, 161, 182, 119, 276, 288, 239, 212, 0]
Route  2 : Profit is 198  Cost is 147.82248527019794 Customers: [0, 294, 197, 267, 297, 214, 281, 211, 102, 99, 124, 253, 0]
Route  3 : Profit is 152  Cost is 148.58978632987447 Customers: [0, 89, 33, 290, 261, 144, 145, 120, 179, 202, 0]
Route  4 : Profit is 113  Cost is 148.21556229949456 Customers: [0, 279, 172, 67, 154, 17, 292, 230, 134, 0]
Route  5 : Profit is 147  Cost is 146.6341783983225 Customers: [0, 249, 112, 257, 244, 128, 213, 105, 152, 75, 41, 164, 0]
Solution Cost is: 739.9905132654976 Solution Profit is: 782
```

6. VND

```
Solution Cost is: 735.8266744716145 Solution Profit is: 741
Profitable Swap Move  -   Iteration number: 1 Solution profit: 741 Solution Cost: 733.5511844284764
Swap Move  -   Iteration number: 2 Solution profit: 741 Solution Cost: 728.2054172526355
Swap Move  -   Iteration number: 3 Solution profit: 741 Solution Cost: 725.6808498411664
Swap Move  -   Iteration number: 4 Solution profit: 741 Solution Cost: 724.1259513558686
Insertion Move  -   Iteration number: 5 Solution profit: 752 Solution Cost: 732.2129225037336
Relocation Move  -   Iteration number: 6 Solution profit: 752 Solution Cost: 729.3309513933548
Insertion Move  -   Iteration number: 7 Solution profit: 759 Solution Cost: 734.4842568405563
Relocation Move  -   Iteration number: 8 Solution profit: 759 Solution Cost: 733.1364686934157
Insertion Move  -   Iteration number: 9 Solution profit: 765 Solution Cost: 739.3568551617452
Relocation Move  -   Iteration number: 10 Solution profit: 765 Solution Cost: 738.2573022449329
Relocation Move  -   Iteration number: 11 Solution profit: 765 Solution Cost: 737.720828286271
Relocation Move  -   Iteration number: 12 Solution profit: 765 Solution Cost: 737.0183851634282
Relocation Move  -   Iteration number: 13 Solution profit: 765 Solution Cost: 736.5393536942721
Relocation Move  -   Iteration number: 14 Solution profit: 765 Solution Cost: 733.3163109943276
Swap Move  -   Iteration number: 15 Solution profit: 765 Solution Cost: 731.7295332460542
Swap Move  -   Iteration number: 16 Solution profit: 765 Solution Cost: 730.8814427869517
Insertion Move  -   Iteration number: 17 Solution profit: 773 Solution Cost: 739.8670700800809
Relocation Move  -   Iteration number: 18 Solution profit: 773 Solution Cost: 737.9156216077135
Profitable Swap Move  -   Iteration number: 19 Solution profit: 776 Solution Cost: 739.2478106880975
Route  1 : Profit is 178  Cost is 149.04376479632307 Customers: [0, 299, 27, 140, 172, 69, 118, 161, 182, 119, 276, 288, 0]
Route  2 : Profit is 164  Cost is 148.26031854166544 Customers: [0, 294, 197, 267, 297, 211, 99, 104, 128, 213, 244, 253, 0]
Route  3 : Profit is 152  Cost is 145.63552613112614 Customers: [0, 89, 212, 239, 33, 290, 261, 144, 145, 120, 0]
Route  4 : Profit is 124  Cost is 147.90528519233084 Customers: [0, 67, 154, 17, 292, 230, 134, 233, 20, 0]
Route  5 : Profit is 158  Cost is 148.4029160266518 Customers: [0, 249, 112, 257, 105, 152, 75, 179, 41, 164, 202, 46, 0]
Solution Cost is: 739.2478106880975 Solution Profit is: 776
```

7. Additional VND Example