Preben Bucher-Johannessen 11.04.18

INF379 - Assignment 3
Operator search for optimal solution in PDPTW

This assignment consist of the following files:
- **«Assignment3.pdf»:** This file...
- **«Assignment3.m»:** The matlab code file that imports data from the provided input data file *«Call_XXX_Vehicle_YY.txt» (manually changeable filename in code),* chooses a random generator seed and runs my chosen operators 10 000 times and keeps for each iteration feasible solutions based on the Simulated Anaeling acceptance criteria. The files stores each run results and calculates averages as presented below in this document.
- **«results.xls»:** The results from the different runs with different operators and the best solutions.

Cooling Schedule
As a cooling schedule I have chosen a T0 = 1 000 000 and I use linear iteration to cool the Temperature through each iteration. After trying different types I found that this gave me a good balance through the iterations as it is very likely in the beginning for a feasible solution to be accepted however very unlikely when you come nearer to the last 4 000 iterations.

Instances
I have numbered the instances from 1-5 and refer to them as such throughout this document and in the excel sheet. The numbers are as follows: **1**-"Call_007_Vehicle_03", **2**-"Call_018_Vehicle_05", **3**-"Call_035_Vehicle_07", **4**-"Call_080_Vehicle_20", **5**-"Call_130_Vehicle_40"

Operators
Each operator from Assignment3.m are described below. They are named here correspondingly to the matlab file, ie. «operator1» from the matlab file is described under Operator 1 here.

Operator 1
Name: Move-a-random-call
Runtime matlab analysis: 2,636

Description: The first operator I made is also the slowest operator, it does however search very well as it is designed much like the random pickup/delivery generator from assignment 1. First it chooses a random call from the initial solution and then it picks another vehicle to send the call to. To insert the call in the new vehicle's schedule it doesn't just put it in the end of the schedule (Like operator 6) or randomly(like operator 7) but it inserts either the new call or the first element from the original schedule until all elements are used up. This makes it likely to insert a calls pickup/delivery earlier in the schedule of that vehicle.

Operator 2
Name: swap
Runtime matlab analysis: 0,516

Description: Performs a swap between two different calls for one vehicle. (this could in some cases end up changing the pickup of a call with the delivery of the same call and therefore not perform any changes. This is also the case if I haven't yet moved any calls to other vehicles than dummy as dummy is ignored (no point in performing switches for dummy as it affects neither feasibility or objective function). This performs better in later stages due to the initial solution.

Operator 3
Name: 3-exchange
Runtime matlab analysis: 0,516

Description: Same as operator 2 only it switches 3 elements with each other instead of 2. Selected vehicle needs more than 2 calls for this to perform a change. This works better on a later stage for the same reasons as for operator 2.

Operator 4
Name: call-swap
Runtime matlab analysis: 0,803

Description: This operator exchanges pickup&delivery of two random calls with each other. ie. [1,2,1,2,0,0,4,4] -> [4,2,4,2,0,0,1,1] if 1 and 4 are selected.

Operator 5
Name: Move-a-random-call-backwards
Runtime matlab analysis: 2,707

Description: This is similar to the first operator I made and it is also (barely) the slowest operator. Like the first one it searches very well and performs well when combined with Operator 1. First it chooses a random call from the initial solution and then it picks another vehicle to send the call to. It goes through the initial solution backwards to give it a higher probability to insert the new call's pickup/delivery in the end of the new vehicles schedule.

Operator 6
Name: Move-a-random-call-to-back
Runtime matlab analysis: 2,686

Description: This is similar to operator 1 only it inserts the selected call in the end of the new vehicles delivery schedule.

Operator 7
Name: Move-a-random-call-random
Runtime matlab analysis: 2,495

Description: This is similar to operator 1 only it inserts the selected call randomly in the schedule of the new vehicle.

Operator choice

I experimented a lot before finding a good combination of operators. The can be viewed stored in the «Results.xls».

Fast swaps
Early on some operators showed not to be as effective as their «brothers and sisters». The swap operator is in general outperformed by the 3-exchange. Also the call swap (operator 4) seems to be better than the 3-exchange at finding good solutions if I am using limiting my use of operators it also has a longer «reach» as it is not limited to a vehicle. The best performance was found using 3 and 4 together for flexibility.

Preben Bucher-Johannessen 11.04.18

Moving calls

Operators 1, 5, 6, 7 are all similar operators that move calls from one vehicle to another. Operator 6 was expected to be the worst, and it also doesn't save any time making it an operator that I have avoided.

Operator 1 & 5 are working very well together as they each try to put a call in the beginning of and end of a delivery schedule. Even with only 3 operators (together with operator 4) they perform very well. However they seem to be the most efficient for smaller data_sets as they are unable to move calls to the middle of a delivery schedule.

Operator 7 is a good operator for moving calls to random positions in a vehicles schedule and has therefore an increased chance of finding the optimum solutions in larger data input.

Choice

As my operators have different advantages I ended up with using 3 different sets of operators to find the optimal solutions as follows:

Choice1: operators 1,3,4,5,7

This is an good choice for larger datasets as it includes all the best Call-moving operators and the best swap operators.

Choice2: operators 1,3,4,7

This is the «merge» version of random (op 7) and specific placement (op 1) which could lead to finding some solutions others might not.

Choice3: operators 1,3,4,5

This is a pure placement operator which should perform well for the smaller datasets but will struggle with finding the optimal solutions in bigger datasets.

Results

I will list the results below according to the choices of operators described above:

Choice 1:

Oper	**1,3,4,5,7**															
Insta	***Initial Obj.***	***Run #1***	***Run #2***	***Run #3***	***Run #4***	***Run #5***	***Run #6***	***Run #7***	***Run #8***	***Run #9***	***Run #10***	***Average Obj***	***Averag***	***Best Obj***	***Best I***	***Average***
1	3760286	1264269	1256140	1256140	1256140	1256140	1256139	1256139	1256139	1256139	1256139	1256952.4	66.573	1256139	66.59	0.31459
2	8761492	2989464	2989464	2989464	2989464	2989464	2989464	2989464	2989464	2989464	2810270	2971544.6	66.084	2810270	67.92	0.59209
3	18322178	8249626	8249626	8249626	8249626	8249626	7791410	7791410	7791410	7791410	7791410	8020518	56.225	7791410	57.48	1.01298
4	42211425	19432081	19432081	19432081	19432081	19432081	19432081	19026449	19026449	19026449	19026449	19269828.2	54.349	19026449	54.93	2.61473
5	75446687	33783117	33028675	32328342	30507570	30507570	30507570	30507570	30507570	30507570	30507570	31269312.4	58.554	**30507570**	59.56	5.20988

Choice 2:

Oper	**1,3,4,7**															
Insta	***Initial Obj.***	***Run #1***	***Run #2***	***Run #3***	***Run #4***	***Run #5***	***Run #6***	***Run #7***	***Run #8***	***Run #9***	***Run #10***	***Average Obj***	***Averag***	***Best Obj***	***Best I***	***Average***
1	3760286	1390325	1256139	1256139	1256139	1256139	1256139	1256139	1256139	1256139	1256139	1269557.6	66.238	1256139	66.59	0.31506
2	8761492	3550140	3537891	3185700	3185700	3185700	3185700	3152684	3152684	3152684	3152684	3244156.7	62.973	3152684	64.02	0.59691
3	18322178	8580567	8580567	8580567	8343329	8343329	8343329	8003460	8003460	8003460	7860925	8264299.3	54.895	7860925	57.1	0.97035
4	42211425	21842641	21842641	21842641	21842641	20477869	20477869	18540321	18540321	18540321	18540321	20248758.6	52.03	**18540321**	56.08	2.5989
5	75446687	35120732	32863668	32523427	32523427	32523427	32523427	32523427	32523427	32523427	32523427	32817181.6	56.503	32523427	56.89	4.86461

Preben Bucher-Johannessen 11.04.18

Choice 3:

Opera	1,3,4,5															
Insta	**Initial Obj.**	**Run #1**	**Run #2**	**Run #3**	**Run #4**	**Run #5**	**Run #6**	**Run #7**	**Run #8**	**Run #9**	**Run #10**	**Average Obj**	**Averag**	**Best Obj**	**Best I**	**Average**
1	3760286	1256139	1256139	1256139	1256139	1256139	1256139	1256139	1256139	1256139	1256139	1256139	66.595	**1256139**	66.59	0.31479
2	8761492	3444976	3108681	3108681	3108681	3108681	2644319	2644319	2644319	2644319	2644319	2910129.5	66.785	**2644319**	69.82	0.60237
3	18322178	7540955	7540955	6887236	6887236	6887236	6887236	6887236	6887236	6887236	6887236	7017979.8	61.697	**6887236**	62.41	1.03599
4	42211425	20969567	20969567	19321825	19321825	19321825	19321825	19321825	18781491	18781491	18781491	19489273.2	53.829	18781491	55.51	2.67119
5	75446687	33429622	33429622	32750407	32750407	32750407	30890171	30890171	30890171	30890171	30890171	31956132	57.644	30890171	59.06	5.0043

Choice 1:
As expected the first choice is on average performing very well for bigger data input. It also found a best solution better than all other combinations on the biggest dataset.

Choice 2:
The merge choice is performance wise falling behind choice 1 and choice 2 but it seems to be good and finding a better optimal solution for mid-sized datasets. This combination found an optimum no other combination could find and this seems to be controlled by the random input from operator 7.

Choice 3:
The best combination for smaller datasets. It finds the best solutions for instance 1-3 in the most runs and also outperforms all other combinations on an average as well as optimum.

Best solutions

The **highlighted** best solutions in the pictures above from choice 1-3 can be found in the worksheet «Best Solutions» in «results.xls» and are summarized here:

Instance 1:
Objective = 1 256 139
Solution: [4,4,6,1,6,1,0,5,5,2,2,0,3,7,7,3,0]

Instance 2:
Objective = 2 644 319
Solution: [4 14 4 14 3 3 0 15 5 15 1 17 17
5 1 0 11 16 11 16 10 10 9 9 0 12
12 8 8 0 18 18 7 7 2 2 0 13 6
13 6]

Instance 3:
Objective: 6 887 236
Solution: [28 21 28 21 35 30 35 30 20 20 0 23 23
17 17 8 2 2 8 0 7 5 5 7 10 10
33 33 4 4 0 34 13 34 13 22 22 27 27
15 15 0 19 25 25 19 11 1 11 1 0 6
6 16 29 16 29 32 32 26 26 0 18 18 14
14 31 31 0 9 9 3 3 24 12 12 24]

Instance 4:
Objective: 18 540 321
Solution: [61 61 18 38 18 38 0 79 79 60 70 70 60
43 43 0 34 34 24 24 65 65 0 48 48 12
12 21 41 21 1 41 1 0 62 62 22 40 22
40 73 73 0 72 72 39 39 78 78 0 74 7
7 74 2 2 0 26 26 19 6 6 19 0 55
55 23 23 35 8 8 35 0 33 33 29 30 30
29 75 75 0 67 67 27 27 64 64 0 53 53
13 31 13 31 0 49 49 15 15 0 80 37 80
17 17 37 0 50 50 14 32 32 14 0 3 5
5 3 0 68 68 36 71 71 36 0 42 42 76
76 0 16 16 44 44 25 25 0 4 4 54 54
10 10 0 52 52 56 20 20 56 9 9 57 57
46 59 69 63 59 47 77 66 66 58 28 28 69
51 11 63 77 11 45 47 51 46 58 45]

Instance 5:
Objective: 30 685 403
Solution: [52 52 112 112 11 11 0 57 57 116 116 20 20
79 79 0 111 111 55 55 40 40 0 71 71
127 127 0 39 39 22 22 42 42 0 103 103 28
2 28 2 106 106 0 61 61 13 13 0 126 35
35 126 38 38 0 80 80 19 19 0 78 67 67
8 78 8 0 74 98 74 98 0 82 82 41 41
0 96 96 105 105 23 23 0 107 107 5 5 0
37 122 122 37 130 130 0 50 50 113 113 0 34
34 24 24 0 58 58 114 114 128 29 128 29 0
118 118 62 62 0 84 7 84 92 7 6 6 92
0 26 26 14 4 14 4 0 64 64 44 44 0
31 31 95 95 70 70 0 120 120 54 54 27 27
36 36 86 86 0 47 47 45 45 0 72 72 9
9 0 90 90 101 101 0 108 66 108 66 0 32
32 119 12 119 12 0 59 59 100 100 0 3 3
0 77 77 63 63 110 110 46 46 0 93 93 85
85 0 121 18 121 18 75 75 0 30 30 88 88
0 16 16 53 53 0 49 49 56 56 0 73
102 102 73 1 124 124 1 0 48 21 21 48 81
129 129 81 0 91 91 60 60 125 125 0 65 69
65 69 94 94 76 123 123 51 76 87 33 33 15
104 104 43 43 17 17 68 68 109 109 83 83
115 115 51 15 10 10 89 89 97 97 87 99 99
117 117 25 25]

Conclusion:
For larger sets it is better to use operators that can give more random neighboring solutions to reach certain optimums. For smaller datasets it is better to use operators that have better search methods.

Possible improvements:

Preben Bucher-Johannessen 11.04.18

Using a random generator to jump from one area to another might be better to reach certain local optimum values. I am however uncertain if this in combination with choice 3 which seems to be the best choice of the 3 for local search will be able to definitely reach all local optimums due to the incapability of reaching certain positions for large vehicle schedules.

Operator 1: In aftermath I would improve operator 1 (or 5) to rather pick a random startingpoint in a vehicles schedule and then set in the pickup/delivery of the selected call with 50% probability. This would probably improve the performance of operator 1 to also then work better for bigger inputs. This would in effect be a combination of operator 7 and operator 1, generating more feasible solutions than operator 7 and reaching more of the schedule (in larger cases) than operator 1/5. It would be a better way to search locally.