

Call7 vehicle 3

	Average objective	Best objective	Average running time
Simmulated annealing adaptive framework	1476936.50	1476444.00	2.52

[3, 3, 0, 7, 1, 7, 1, 0, 5, 5, 6, 6, 0, 4, 2, 4, 2]

Call18 vehicle 5

	Average objective	Best objective	Average running time
Simmulated annealing adaptive framework	2729190.70	2479627.00	9.24

[4, 14, 4, 14, 3, 3, 0, 15, 15, 6, 17, 6, 17, 10, 10, 9, 9, 0, 11, 16, 16, 11, 0, 12, 12, 1, 1, 0, 18, 5, 5, 18, 8, 7, 7, 8, 2, 2, 0, 13, 13]

Call35 vehicle 7

	Average objective	Best objective	Average running time
Simmulated annealing adaptive framework	5507420.00	5130347.00	30.88

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

Call80 vehicle 20

	Average objective	Best objective	Average running time
Simmulated annealing adaptive framework	14035978.30	14035978.30	119.01

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

Call130 vehicle 40

	Average objective	Best objective	Average running time
Simmulated annealing adaptive framework	20001593.90	18986636.00	390.55

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

My algorithm uses 4 weighted operators (2-opt, 3-opt, insert, swap in car) with the intention of having some operators that diversify- and some that intensify the solution. The weight of the operators (the chance of picking that operator this iteration) is recalculated each 100 iterations by looking at how that operator did during the last 100 iterations compared to the 100 iterations before that. The 2-opt and 3-opt operators picks 2 and 3 calls respectively and swaps them, they will repeat doing this until they get a solution with is good enough or they have run a set number of iterations. The insert operator will remove a call and try to reinsert it somewhere else. The swap in car operator will choose 2 numbers from the solutions in the same car (one part of a call, either pickup or delivery) and swaps them. The thought behind the swap operator is that is a good way of intensifying the solution but I don't think it works as good as I hoped for. The algorithm runs for 30000 iterations or for until the given time limit is reached.

Relevant System configuration

Intel i5-8250U CPU 1.700GHz

8GB RAM

Running python code with pypy3 <https://www.pypy.org/>