Final Assignment

Runar Fosse

The algorithm ran is ALNS, with the deterministic record-to-record acceptance criteria. Every testcase (except the first two) run for 15 minutes concurrently. If there has not been an improvement upon the current best solution after 7500 iterations, the escape algorithm will run.

I have 6 different operators, made from the combinations of these different removal/insertion heuristics:

Removal heuristics:
- Shaw removal: Remove a random amount of similar calls.
- Costly removal: Remove a random amount of the current most costly calls.
- Random removal: Remove a random amount of randomly selected calls.

Insertion heuristics:
- Greedy insertion: Greedily insert the current cheapest call to insert into its position.
- Regret-k insertion: Insert the call with the highest calculated regret-k into its best position.

My escape algorithm works by first having a small chance (relative to instance size) to reset the incumbent back to the current best solution seen. This helps prevent a case where several escape algorithms move the solution so far away from the current optimum that it gets stuck. Then I perform a small step 25 times, moving between 2% and 6% of all calls with a random removal - greedy insertion operator, accepting all intermediate solutions.

The record-to-record acceptance criteria has been modified and tuned such that it generally fits every instance size very well.

Instance: Call_7_Vehicle_3				
	Average objective	Best objective	Improvement (%)	Running time
Final Metaheuristic	1134176.00	1134176.00	65.022906 %	30.000 s
Best solution: [4, 4, 2, 2, 0, 7, 7, 0, 1, 5, 5, 3, 3, 1, 0, 6, 6]				

Instance: Call_18_Vehicle_5				
	Average objective	Best objective	Improvement (%)	Running time
Final Metaheuristic	2374420.00	2374420.00	73.499132 %	120.000 s
Best solution: [4, 4, 15, 15, 11, 11, 16, 16, 0, 6, 6, 5, 18, 5, 14, 17, 17, 14, 18, 0, 9, 8, 8, 9, 13, 13, 0, 7, 7, 3, 3, 10, 1, 10, 1, 0, 12, 12, 0, 2, 2]				

Instance: Call_35_Vehicle_7				
	Average objective	Best objective	Improvement (%)	Running time
Final Metaheuristic	4893734.00	4893734.00	73.386004 %	900.000 s
Best solution: [34, 23, 23, 14, 34, 17, 14, 17, 27, 27, 28, 28, 20, 20, 0, 16, 16, 30, 7, 7, 30, 33, 33, 0, 19, 19, 24, 24, 22, 22, 25, 25, 31, 31, 0, 4, 4, 15, 15, 3, 1, 3, 21, 21, 1, 0, 8, 11, 11, 18, 8, 18, 5, 5, 2, 29, 2, 29, 0, 9, 6, 9, 35, 6, 13, 35, 13, 26, 32, 32, 26, 0, 12, 12, 10, 10, 0]			

Instance: Call_80_Vehicle_20				
	Average objective	Best objective	Improvement (%)	Running time
Final Metaheuristic	10481208.00	10340869.00	77.885796 %	900.010 s
Best solution: [25, 68, 25, 62, 62, 68, 20, 20, 16, 16, 0, 22, 22, 34, 59, 34, 17, 59, 27, 27, 17, 33, 33, 0, 60, 60, 64, 64, 0, 54, 63, 63, 54, 0, 74, 79, 74, 28, 28, 58, 79, 58, 0, 18, 18, 35, 42, 35, 3, 3, 42, 80, 80, 7, 7, 0, 66, 66, 5, 5, 77, 77, 0, 39, 61, 39, 61, 55, 46, 55, 36, 50, 46, 14, 36, 50, 14, 0, 8, 8, 49, 49, 72, 73, 73, 10, 72, 10, 0, 15, 70, 15, 23, 23, 11, 11, 45, 45, 70, 69, 48, 48, 69, 0, 21, 37, 37, 21, 0, 4, 4, 71, 71, 65, 12, 65, 12, 0, 1, 26, 26, 1, 0, 76, 44, 76, 44, 0, 51, 43, 51, 43, 52, 56, 56, 52, 0, 32, 41, 41, 29, 32, 29, 78, 78, 2, 2, 24, 24, 0, 53, 30, 53, 30, 9, 19, 9, 19, 47, 47, 6, 6, 0, 38, 38, 31, 31, 0, 40, 40, 13, 13, 75, 75, 0, 57, 67, 67, 57, 0]			

Instance: Call_130_Vehicle_40				
	Average objective	Best objective	Improvement (%)	Running time
Final Metaheuristic	16568530.50	16424758.00	78.496880 %	900.028 s
Best solution: [3, 3, 121, 41, 41, 121, 107, 107, 37, 28, 36, 28, 77, 36, 37, 77, 0, 60, 60, 102, 102, 82, 45, 45, 82, 65, 127, 65, 127, 0, 105, 105, 62, 62, 63, 83, 83, 63, 0, 27, 27, 124, 67, 124, 92, 67, 46, 92, 46, 0, 126, 126, 22, 22, 26, 26, 0, 42, 96, 96, 50, 42, 118, 118, 9, 17, 17, 9, 50, 0, 25, 25, 59, 7, 7, 59, 64, 64, 0, 5, 73, 5, 99, 99, 73, 10, 53, 30, 10, 53, 30, 0, 103, 103, 4, 4, 0, 15, 15, 16, 16, 97, 97, 0, 80, 80, 0, 123, 74, 123, 74, 34, 68, 70, 34, 70, 68, 20, 20, 0, 0, 49, 49, 33, 33, 125, 81, 125, 12, 81, 12, 0, 23, 23, 55, 109, 109, 55, 0, 88, 88, 130, 130, 110, 110, 48, 48, 0, 84, 98, 98, 114, 114, 84, 90, 18, 90, 122, 18, 91, 94, 94, 122, 91, 0, 11, 11, 120, 120, 61, 104, 61, 51, 51, 104, 0, 0, 129, 129, 117, 117, 39, 39, 0, 115, 113, 113, 115, 0, 87, 69, 87, 52, 69, 35, 52, 35, 0, 72, 112, 112, 72, 0, 106, 66, 106, 66, 0, 75, 75, 0, 21, 32, 32, 21, 0, 6, 44, 6, 44, 0, 2, 2, 78, 78, 0, 85, 85, 89, 43, 89, 43, 0, 0, 119, 119, 0, 19, 19, 128, 128, 0, 38, 38, 1, 31, 31, 1, 0, 86, 86, 58, 29, 58, 29, 40, 24, 24, 40, 0, 100, 79, 100, 76, 79, 76, 0, 108, 108, 0, 116, 116, 57, 57, 0, 8, 95, 8, 47, 95, 56, 56, 47, 71, 14, 71, 14, 0, 93, 93, 13, 101, 13, 101, 0, 54, 111, 54, 111, 0]			

Instance: Call_300_Vehicle_90				
	Average objective	Best objective	Improvement (%)	Running time
Final Metaheuristic	36118010.30	35882927.00	78.946547 %	900.041 s

Best solution: [0, 206, 206, 0, 217, 217, 0, 62, 62, 253, 253, 0, 96, 263, 96, 7, 7, 275, 275, 263, 0, 273, 200, 200, 273, 0, 10, 92, 92, 10, 0, 0, 0, 15, 15, 271, 271, 183, 185, 183, 58, 185, 58, 0, 80, 80, 0, 0, 118, 41, 41, 118, 11, 11, 0, 126, 44, 126, 154, 154, 44, 0, 54, 54, 0, 0, 0, 0, 0, 212, 181, 148, 212, 181, 148, 3, 3, 0, 248, 248, 77, 90, 77, 90, 0, 233, 233, 0, 35, 215, 270, 35, 215, 270, 0, 0, 23, 23, 279, 279, 0, 36, 155, 155, 195, 36, 195, 0, 222, 222, 293, 209, 209, 293, 0, 111, 89, 89, 111, 0, 246, 246, 72, 220, 220, 72, 0, 0, 68, 68, 274, 274, 119, 119, 4, 4, 156, 156, 0, 123, 269, 269, 123, 28, 70, 28, 70, 166, 166, 267, 267, 0, 193, 193, 285, 285, 205, 205, 97, 97, 0, 43, 43, 177, 177, 0, 211, 79, 79, 122, 211, 153, 122, 153, 0, 137, 137, 32, 46, 46, 297, 32, 297, 296, 296, 0, 276, 276, 52, 52, 8, 8, 0, 180, 180, 109, 240, 74, 74, 240, 109, 204, 204, 107, 107, 106, 106, 0, 241, 225, 48, 225, 48, 64, 241, 64, 0, 143, 143, 230, 230, 170, 170, 16, 16, 192, 192, 292, 198, 198, 292, 0, 158, 114, 158, 114, 108, 108, 125, 139, 139, 125, 117, 117, 0, 244, 244, 94, 300, 300, 94, 219, 227, 115, 227, 219, 115, 0, 242, 242, 144, 165, 144, 165, 251, 251, 0, 168, 168, 202, 202, 56, 18, 56, 140, 140, 18, 0, 150, 61, 150, 61, 149, 149, 191, 196, 191, 196, 0, 250, 286, 141, 250, 286, 141, 21, 163, 163, 21, 0, 29, 29, 49, 257, 257, 49, 266, 259, 259, 266, 0, 264, 83, 264, 254, 254, 83, 0, 159, 159, 199, 81, 199, 84, 187, 81, 17, 17, 187, 84, 0, 152, 136, 136, 152, 167, 167, 255, 27, 27, 255, 101, 22, 22, 101, 0, 105, 295, 105, 194, 164, 295, 194, 235, 164, 235, 33, 33, 0, 142, 142, 262, 262, 289, 102, 102, 289, 130, 130, 0, 278, 277, 278, 210, 210, 277, 98, 98, 127, 120, 120, 127, 100, 100, 161, 161, 0, 221, 91, 221, 91, 14, 104, 14, 104, 0, 232, 272, 232, 272, 290, 63, 290, 60, 60, 258, 258, 63, 0, 37, 66, 66, 281, 37, 281, 47, 47, 129, 186, 186, 129, 0, 138, 138, 24, 24, 234, 234, 34, 50, 34, 298, 298, 50, 0, 65, 65, 229, 229, 5, 226, 73, 226, 5, 99, 99, 73, 0, 133, 133, 172, 207, 172, 207, 176, 283, 283, 176, 0, 218, 218, 256, 256, 184, 124, 124, 184, 282, 282, 231, 231, 0, 39, 288, 203, 182, 39, 55, 203, 182, 288, 55, 0, 75, 228, 228, 75, 0, 171, 178, 178, 171, 216, 216, 190, 190, 0, 85, 12, 85, 20, 20, 12, 0, 299, 69, 69, 299, 13, 13, 0, 201, 87, 87, 128, 201, 173, 173, 128, 0, 162, 151, 162, 151, 157, 2, 2, 157, 0, 103, 30, 53, 103, 51, 59, 59, 30, 53, 93, 51, 93, 0, 265, 110, 265, 189, 110, 112, 112, 189, 0, 0, 213, 145, 213, 145, 0, 67, 57, 67, 57, 0, 147, 147, 175, 175, 0, 31, 236, 31, 236, 0, 238, 238, 0, 19, 224, 19, 224, 0, 25, 1, 25, 86, 86, 6, 6, 1, 0, 247, 247, 249, 237, 249, 237, 0, 174, 284, 284, 174, 0, 280, 287, 287, 280, 132, 113, 113, 132, 0, 116, 116, 243, 131, 131, 42, 243, 38, 38, 252, 42, 252, 0, 76, 76, 0, 223, 268, 268, 223, 0, 245, 26, 245, 26, 0, 134, 134, 188, 188, 71, 9, 71, 9, 197, 146, 146, 197, 0, 121, 88, 121, 294, 294, 88, 0, 208, 78, 78, 208, 261, 45, 95, 45, 95, 261, 0, 179, 179, 169, 291, 169, 291, 160, 160, 0, 214, 239, 214, 239, 0, 82, 135, 135, 82, 260, 40, 260, 40, 0]