Instance	$\|V^+\|$	Random construct					Nearest Neighbour construct				
		Best cost	Avg cost	Dev (%)	Time (ms)	Best	Best cost	Avg cost	Dev (%)	Time (ms)	Best
vrpnc1a	50	1484.4	1710.0	15.2	0.18	0	766.9	766.9	0	1.26	1
vrpnc2a	75	2411.0	2737.7	13.6	0.25	0	1241.8	1241.8	0	2.76	1
vrpnc3b	100	3191.3	3558.4	11.5	0.29	0	1276.5	1276.5	0	4.73	1
vrpnc4a	150	4800.4	5305.9	10.5	0.42	0	1668.0	1668.0	0	10.33	1
vrpnc5b	199	6513.7	6990.4	7.3	0.56	0	2137.2	2137.2	0	17.79	1
vrpnc6b	50	1479.1	1700.5	15.0	0.18	0	796.7	796.7	0	1.40	1
vrpnc9a	150	4833.3	5385.6	11.4	0.46	0	1668.0	1668.0	0	10.16	1
vrpnc14a	100	3698.0	4173.2	12.9	0.30	0	1126.3	1126.3	0	4.66	1

Table 1: Comparison results between constructive methods (120 runs per instance)

Instance	$\|V^+\|$	Intraroute swap					Intraroute insertion				
		Best cost	Avg cost	Dev (%)	Time (s)	Best	Best cost	Avg cost	Dev (%)	Time (s)	Best
vrpnc1a	50	739.0	739.0	0	0.001	0	725.8	731.8	0.8	0.003	0
vrpnc2a	75	1218.4	1218.4	0	0.001	0	1196.1	1196.7	0.0	0.002	0
vrpnc3b	100	1261.6	1261.6	0	0.002	0	1242.2	1242.4	0.0	0.007	0
vrpnc4a	150	1598.2	1610.9	0.8	0.003	0	1579.1	1583.6	0.3	0.016	0
vrpnc5b	199	2104.2	2107.3	0.1	0.002	0	2073.5	2086.4	0.6	0.013	0
vrpnc6b	50	750.4	754.3	0.5	0.001	0	744.2	744.2	0	0.004	0
vrpnc9a	150	1600.2	1611.0	0.7	0.003	0	1579.1	1583.9	0.3	0.016	0
vrpnc14a	100	1101.4	1103.3	0.2	0.001	0	1083.0	1088.3	0.5	0.011	0

Instance	$\|V^+\|$	Interroute swap					Interroute insertion				
		Best cost	Avg cost	Dev (%)	Time (s)	Best	Best cost	Avg cost	Dev (%)	Time (s)	Best
vrpnc1a	50	766.9	766.9	0	0.002	0	611.9	679.2	11.0	0.216	1
vrpnc2a	75	1131.6	1144.7	1.2	0.018	0	1065.4	1122.0	5.3	0.447	1
vrpnc3b	100	1227.9	1227.9	0	0.044	0	995.5	1074.6	7.9	8.524	1
vrpnc4a	150	1468.6	1502.5	2.3	0.294	0	1294.3	1417.9	9.6	24.758	1
vrpnc5b	199	1853.6	1917.2	3.4	9.728	1	1847.1	1917.4	3.8	26.181	0
vrpnc6b	50	629.4	653.0	3.7	0.560	1	625.8	653.8	4.5	0.357	0
vrpnc9a	150	1284.9	1422.0	10.7	9.054	0	1295.5	1407.8	8.7	27.291	1
vrpnc14a	100	1126.3	1126.3	0	0.141	0	1126.3	1126.3	0	0.094	1

Instance	$\|V^+\|$	2-opt swap				
		Best cost	Avg cost	Dev (%)	Time (s)	Best
vrpnc1a	50	766.9	766.9	0	0.002	0
vrpnc2a	75	1241.8	1241.8	0	0.002	0
vrpnc3b	100	1276.5	1276.5	0	0.007	0
vrpnc4a	150	1668.0	1668.0	0	0.012	0
vrpnc5b	199	2137.1	2137.1	0	0.011	0
vrpnc6b	50	796.7	796.7	0	0.003	0
vrpnc9a	150	1668.0	1668.0	0	0.012	0
vrpnc14a	100	1126.3	1126.3	0	0.006	0

Table 2: Comparison results between local search algorithms (120 runs per instance)

Note: Each time in every table is the average time for each instance execution

Instance	$\|V^+\|$	More exhaustive searches first					Less exhaustive searches first				
		Best cost	Avg cost	Dev (%)	Time (s)	Best	Best cost	Avg cost	Dev (%)	Time (s)	Best
vrpnc1a	50	563.8	590.5	4.7	0.513	1	590.3	617.3	4.6	0.241	0
vrpnc2a	75	965.5	1000.4	3.6	0.928	0	927.5	988.4	6.6	0.673	1
vrpnc3b	100	960.0	994.4	3.6	4.416	0	934.5	984.8	5.4	2.193	1
vrpnc4a	150	1215.5	1286.3	5.8	14.936	0	1183.7	1230.2	3.9	7.129	1
vrpnc5b	199	1671.4	1730.5	3.5	19.589	1	1737.3	1770.2	1.9	9.362	0
vrpnc6b	50	583.7	591.9	1.4	0.684	1	612.1	617.6	0.9	0.302	0
vrpnc9a	150	1179.9	1286.9	9.1	15.955	0	1170.1	1228.0	4.9	9.731	1
vrpnc14a	100	1024.9	1040.4	1.5	2.312	1	1065.6	1066.3	0.0	0.342	0

Table 3: Results of the comparison between the execution order of local search algorithms in the VND algorithm (24 runs per instance)

Instance	$\|V^+\|$	All local searches					Variable Neighborhood Descent				
		Best cost	Avg cost	Dev (%)	Time (s)	Best	Best cost	Avg cost	Dev (%)	Time (s)	Best
vrpnc1a	50	618.0	641.7	3.8	0.157	0	590.3	617.3	4.6	0.241	1
vrpnc2a	75	966.8	1053.3	9.0	0.348	0	927.5	988.4	6.6	0.673	1
vrpnc3b	100	984.6	1032.0	4.8	2.096	0	934.5	984.8	5.4	2.193	1
vrpnc4a	150	1280.4	1331.1	4.0	6.854	0	1183.7	1230.2	3.9	7.129	1
vrpnc5b	199	1833.2	1886.6	2.9	8.230	0	1737.3	1770.2	1.9	9.362	1
vrpnc6b	50	623.3	636.6	2.1	0.201	0	612.1	617.6	0.9	0.302	1
vrpnc9a	150	1275.3	1327.4	4.1	8.855	0	1170.1	1228.0	4.9	9.731	1
vrpnc14a	100	1074.3	1074.9	0.1	0.266	0	1065.6	1066.3	0.0	0.342	1

Table 4: Results of the comparison between a simple sequence of all local search algorithms and the VND algorithm (24 runs per instance)

Instance	$\|V^+\|$	GVNS, $k_{max} = 0.1$					GVNS, $k_{max} = 0.2$				
		Best cost	Avg cost	Dev (%)	Time (s)	Best	Best cost	Avg cost	Dev (%)	Time (s)	Best
vrpnc1a	50	558.0	582.9	4.5	1.050	0	552.4	570.2	3.2	1.684	0
vrpnc2a	75	883.3	932.7	5.6	2.242	0	888.7	907.6	2.1	4.529	0
vrpnc3b	100	902.4	942.3	4.4	10.982	0	889.5	921.6	3.6	24.403	0
vrpnc4a	150	1131.6	1184.2	4.6	32.941	0	1129.4	1168.3	3.4	55.269	0
vrpnc5b	199	1563.1	1664.2	6.5	32.275	0	1562.3	1633.9	4.6	81.138	0
vrpnc6b	50	551.9	598.5	8.4	0.772	0	563.2	583.8	3.7	2.061	0
vrpnc9a	150	1145.6	1179.8	3.0	29.290	0	1126.0	1162.9	3.3	60.045	0
vrpnc14a	100	967.7	1012.8	4.7	2.313	0	948.7	997.4	5.1	5.847	0

Instance	$\|V^+\|$	GVNS, $k_{max} = 0.3$					GVNS, $k_{max} = 0.4$				
		Best cost	Avg cost	Dev (%)	Time (s)	Best	Best cost	Avg cost	Dev (%)	Time (s)	Best
vrpnc1a	50	550.7	568.9	3.3	2.289	0	550.9	564.9	2.5	3.597	0
vrpnc2a	75	888.7	904.2	1.7	8.749	0	880.2	902.6	2.5	8.199	0
vrpnc3b	100	892.0	922.3	3.4	32.211	0	900.0	923.7	2.6	40.473	0
vrpnc4a	150	1133.5	1166.2	2.9	80.987	0	1128.9	1161.5	2.9	104.256	0
vrpnc5b	199	1540.5	1615.5	4.9	127.915	0	1490.6	1611.3	8.1	190.69	1
vrpnc6b	50	553.4	577.1	4.3	2.651	1	561.6	572.6	2.0	4.158	0
vrpnc9a	150	1127.9	1162.0	3.0	90.501	0	1131.4	1169.4	3.4	97.215	0
vrpnc14a	100	957.5	989.6	3.3	8.754	0	955.4	976.0	2.2	11.586	0

Instance	$\|V^+\|$	GVNS, $k_{max} = 0.5$				
		Best cost	Avg cost	Dev (%)	Time (s)	Best
vrpnc1a	50	550.7	565.9	2.8	3.821	1
vrpnc2a	75	878.7	901.9	2.6	12.568	1
vrpnc3b	100	885.0	927.8	4.8	48.950	1
vrpnc4a	150	1126.5	1159.4	2.9	146.017	1
vrpnc5b	199	1547.3	1612.9	4.2	228.350	0
vrpnc6b	50	555.3	571.7	3.0	4.958	0
vrpnc9a	150	1121.7	1160.5	3.5	140.961	1
vrpnc14a	100	942.4	984.2	4.4	11.663	1

Table 5: Results of the comparison of GVNS algorithms based on their k_{max} parameter (24 runs per instance)

Instance	$\lvert V^+ \rvert$	VND					GVNS ($k_{max} = 0.5$)				
		Best cost	Avg cost	Dev (%)	Time (s)	Best	Best cost	Avg cost	Dev (%)	Time (s)	Best
vrpnc1a	50	590.3	617.3	4.6	0.241	0	550.7	565.9	2.8	3.821	1
vrpnc2a	75	927.5	988.4	6.6	0.673	0	878.7	901.9	2.6	12.568	1
vrpnc3b	100	934.5	984.8	5.4	2.193	0	885.0	927.8	4.8	48.950	1
vrpnc4a	150	1183.7	1230.2	3.9	7.129	0	1126.5	1159.4	2.9	146.017	1
vrpnc5b	199	1737.3	1770.2	1.9	9.362	0	1547.3	1612.9	4.2	228.350	1
vrpnc6b	50	612.1	617.6	0.9	0.302	0	555.3	571.7	3.0	4.958	1
vrpnc9a	150	1170.1	1228.0	4.9	9.731	0	1121.7	1160.5	3.5	140.961	1
vrpnc14a	100	1065.6	1066.3	0.0	0.342	0	942.4	984.2	4.4	11.663	1

Table 6: Results of the comparison between the VND algorithm and the GVNS algorithm (24 runs per instance)

Instance	$\|V^+\|$	All local searches					VND				
		Best cost	Avg cost	Dev (%)	Time (s)	Best	Best cost	Avg cost	Dev (%)	Time (s)	Best
vrpnc1a	50	613.5	643.8	4.9	0.17	0	586.3	622.1	6.1	0.236	0
vrpnc1b	50	616.4	634.0	2.9	0.166	0	593.9	618.1	4.1	0.212	0
vrpnc2a	75	978.8	1052.4	7.5	0.421	0	926.0	984.8	6.4	0.683	0
vrpnc2b	75	1029.2	1071.8	4.1	0.559	0	919.6	1002.4	9.0	0.587	0
vrpnc3a	100	977.7	1028.2	5.2	5.208	0	934.1	976.6	4.5	5.796	0
vrpnc3b	100	968.1	1043.4	7.8	3.173	0	932.9	982.2	5.3	3.526	0
vrpnc4a	150	1260.1	1327.3	5.3	8.909	0	1178.7	1234.6	4.7	9.339	0
vrpnc4b	150	1365.4	1432.6	4.9	4.832	0	1266.7	1317.2	4.0	6.85	0
vrpnc5a	199	1630.8	1689.5	3.6	12.788	0	1506.2	1568.9	4.2	14.728	0
vrpnc5b	199	1842.1	1900.7	3.2	8.424	0	1732.1	1775.6	2.5	9.002	0
vrpnc6a	50	632.2	642.5	1.6	0.124	0	603.2	610.1	1.1	0.32	0
vrpnc6b	50	612.1	631.3	3.1	0.212	0	596.1	617.6	3.6	0.288	0
vrpnc7a	75	974.3	1012.5	3.9	0.778	0	921.6	949.8	3.1	0.924	0
vrpnc7b	75	993.9	1054.1	6.0	0.782	0	930.8	991.7	6.5	0.917	0
vrpnc8a	100	996.9	1025.9	2.9	6.047	0	939.4	975.8	3.9	6.242	0
vrpnc8b	100	985.0	1043.2	5.9	2.933	0	938.0	980.2	4.5	3.497	0
vrpnc9a	150	1268.0	1327.0	4.7	9.258	0	1183.1	1238.3	4.7	9.46	0
vrpnc9b	150	1374.9	1433.8	4.3	4.857	0	1257.5	1320.4	5.0	6.845	0
vrpnc10a	199	1629.5	1696.4	4.1	11.501	0	1496.3	1570.3	4.9	14.535	0
vrpnc10b	199	1817.8	1902.5	4.7	8.436	0	1737.3	1779.0	2.4	8.393	0
vrpnc11a	120	1378.4	1440.1	4.5	1.096	0	1302.4	1366.9	5.0	2.386	0
vrpnc11b	120	1352.3	1389.5	2.7	1.37	0	1292.1	1323.6	2.4	3.021	0
vrpnc12a	100	1074.3	1075.2	0.1	0.26	0	1065.6	1066.0	0.0	0.334	0
vrpnc12b	100	1108.5	1140.2	2.9	1.728	0	1048.3	1068.3	1.9	2.848	0
vrpnc13a	120	1378.4	1442.4	4.6	1.251	0	1311.1	1379.4	5.2	2.48	0
vrpnc13b	120	1356.1	1386.4	2.2	1.634	0	1295.7	1328.4	2.5	2.854	0
vrpnc14a	100	1074.3	1074.8	0.0	0.293	0	1065.6	1066.2	0.1	0.328	0
vrpnc14b	100	1112.0	1142.2	2.7	1.792	0	1045.0	1062.4	1.7	2.85	0

Instance	$\|V^+\|$	GVNS ($k_{max} = 0.5$)				
		Best cost	Avg cost	Dev (%)	Time (s)	Best
vrpnc1a	50	550.7	563.7	2.4	4.671	1
vrpnc1b	50	552.1	573.1	3.8	6.427	1
vrpnc2a	75	881.6	905.1	2.7	14.824	1
vrpnc2b	75	889.3	918.9	3.3	12.196	1
vrpnc3a	100	874.9	903.3	3.3	56.469	1
vrpnc3b	100	876.1	920.1	5.0	54.927	1
vrpnc4a	150	1123.0	1167.8	4.0	138.402	1
vrpnc4b	150	1154.2	1216.1	5.4	151.152	1
vrpnc5a	199	1447.5	1497.7	3.5	278.136	1
vrpnc5b	199	1486.6	1604.1	7.9	264.668	1
vrpnc6a	50	550.1	559.1	1.6	5.507	1
vrpnc6b	50	555.3	572.1	3.0	5.934	1
vrpnc7a	75	885.6	901.7	1.8	12.727	1
vrpnc7b	75	897.9	920.8	2.6	15.24	1
vrpnc8a	100	882.9	904.4	2.4	55.084	1
vrpnc8b	100	883.3	918.2	3.9	54.358	1
vrpnc9a	150	1127.8	1165.7	3.4	136.076	1
vrpnc9b	150	1154.6	1218.8	5.6	149.752	1
vrpnc10a	199	1436.2	1500.5	4.5	251.643	1
vrpnc10b	199	1523.9	1600.7	5.0	264.785	1
vrpnc11a	120	1254.7	1331.3	6.1	53.976	1
vrpnc11b	120	1254.9	1311.8	4.5	60.864	1
vrpnc12a	100	935.0	979.2	4.7	16.679	1
vrpnc12b	100	983.7	1037.4	5.5	40.507	1
vrpnc13a	120	1245.2	1327.2	6.6	57.796	1
vrpnc13b	120	1259.2	1313.3	4.3	60.505	1
vrpnc14a	100	929.0	977.6	5.2	18.235	1
vrpnc14b	100	981.6	1033.7	5.3	38.666	1

Table 7: Results of the comparison of a sequence of local search algorithms, the VND algorithm, and the GVNS algorithm (48 runs per instance)

Instance	$\|V^+\|$	BKS	GVNS, $k_{max} = 0.5$				
			Best cost	Avg cost	Dev (%)	Time (s)	Best
vrpnc1a	50	550.7	550.7	563.7	2.4	4.671	0
vrpnc1b	50	551.9	552.1	573.1	3.8	6.427	0
vrpnc2a	75	868.6	881.6	905.1	2.7	14.824	0
vrpnc2b	75	878.7	889.3	918.9	3.3	12.196	0
vrpnc3a	100	860.4	874.9	903.3	3.3	56.469	0
vrpnc3b	100	859.4	876.1	920.1	5.0	54.927	0
vrpnc4a	150	1075.7	1123.0	1167.8	4.0	138.402	0
vrpnc4b	150	1092.1	1154.2	1216.1	5.4	151.152	0
vrpnc5a	199	1381.6	1447.5	1497.7	3.5	278.136	0
vrpnc5b	199	1373.3	1486.6	1604.1	7.9	264.668	0
vrpnc6a	50	557.5	550.1	559.1	1.6	5.507	0
vrpnc6b	50	557.5	555.3	572.1	3.0	5.934	0
vrpnc7a	75	920.5	885.6	901.7	1.8	12.727	0
vrpnc7b	75	930.7	897.9	920.8	2.6	15.24	0
vrpnc8a	100	876.7	882.9	904.4	2.4	55.084	0
vrpnc8b	100	875.3	883.3	918.2	3.9	54.358	0
vrpnc9a	150	1177.2	1127.8	1165.7	3.4	136.076	0
vrpnc9b	150	1175.7	1154.6	1218.8	5.6	149.752	0
vrpnc10a	199	1469.7	1436.2	1500.5	4.5	251.643	0
vrpnc10b	199	1470.8	1523.9	1600.7	5.0	264.785	0
vrpnc11a	120	1103.5	1254.7	1331.3	6.1	53.976	0
vrpnc11b	120	1199.0	1254.9	1311.8	4.5	60.864	0
vrpnc12a	100	899.6	935.0	979.2	4.7	16.679	0
vrpnc12b	100	950.8	983.7	1037.4	5.5	40.507	0
vrpnc13a	120	1556.5	1245.2	1327.2	6.6	57.796	0
vrpnc13b	120	1550.1	1259.2	1313.3	4.3	60.505	0
vrpnc14a	100	911.0	929.0	977.6	5.2	18.235	0
vrpnc14b	100	964.7	981.6	1033.7	5.3	38.666	0

Table 8: Results of the comparison between the GVNS algorithm and the best-known solution (BKS) (48 runs per instance)