Yugoslav Journal of Operations Research
25 (2015), Number 2, 169–184
DOI: 10.2298/YJOR140217011K

Invited survey

ADAPTIVE SEARCH TECHNIQUES FOR PROBLEMS IN VEHICLE ROUTING, PART II: A NUMERICAL COMPARISON

Stefanie KRITZINGER, Karl F. DOERNER
Department of Production and Logistics, Johannes Kepler University Linz, Austria
stefanie.kritzinger@jku.at, karl.doerner@jku.at

Fabien TRICOIRE, Richard F. HARTL
Department of Business Administration, University of Vienna, Austria
fabien.tricoire@univie.ac.at, richard.hartl@univie.ac.at

Received: January 2014 / Accepted: April 2014

Abstract: Research in the field of vehicle routing is often focused on finding new ideas and concepts in the development of fast and efficient algorithms for an improved solution process. Early studies introduce static tailor-made strategies, but trends show that algorithms with generic adaptive policies - which emerged in the past years - are more efficient to solve complex vehicle routing problems. In the first part of the survey, we presented an overview of recent literature dealing with adaptive or guided search techniques for problems in vehicle routing.

Keywords: Adaptive Strategies, Local Search, Variable Neighborhood Search, Vehicle Routing.

MSC: 90B06, 90C05, 90C08.

1. INTRODUCTION

As it is shown in Part I of this survey [10], different adaptive mechanisms can be used when solving vehicle routing problems (VRPs) with metaheuristics. The survey started with basic local search-based methods, e.g. adaptive tabu search or guided local search, followed by hybrid local search methods, e.g. iterated local search (ILS), adaptive variable neighborhood search (AVNS), and adaptive

large neighborhood search (ALNS). The survey concluded with population-based methods, e.g. ant colony optimization, memetic and genetic algorithms.

In this second part, we evaluate and analyze different adaptive strategies on the open VRP (OVRP) (see, e.g., [1, 18]) and the OVRP with time windows (OVRPTW) (see, e.g., [17]). For that purpose, we integrate the adaptive strategies into a solution framework for vehicle routing, which is based on variable neighborhood search (VNS). In Section 2, we present this VNS framework wherein the adaptive strategies are then integrated. In Section 3, different adaptive strategies based on recent literature in VNS are described. In Section 4, the experimental study is conducted. We provide a comparative summary of these results in Section 5.

2. SOLUTION METHOD

The VNS algorithm proposed by Mladenović and Hansen [13] has gained popularity because of its ability to solve combinatorial problems across a wide field of applications [7]. For example, VNS has been used to tackle many VRPs, including the periodic VRP [8], the dial-a-ride problem [14], the multi-depot VRP with time windows [16], and the multi-period orienteering problem with multiple time windows [22]. The basic steps of VNS are *initialization*, *shaking*, *local search*, and *acceptance decision*. A precise descriptions of VNS is available in the first part of the survey [10] and in prior research [5, 6, 7, 13]. In the following subsections, we describe the different components of our unified VNS (UVNS) algorithm that can solve several vehicle routing variants.

2.1. Initial solution

For the initial solution, we perform the cheapest insertion heuristic. We start with the fixed number of empty routes and fill them with customers by inserting the customer not already routed that results in the lowest increase in the total travel cost. We improve the starting solution with four local search procedures: 2-opt, Or-opt, cross-move, and 2-opt*. These methods, described in detail in Section 2.3, are executed according to a first improvement strategy until no improvement is obtained. The achieved solution then provides the first incumbent and the best found solution.

2.2. Shaking

For the shaking step, a set of neighborhoods must be defined, and the neighborhood operators are characterized by their ability to perturb the incumbent solution while keeping important parts of it unchanged.

We randomly take one among three shaking variants: a cross and icross-exchange, a sequence ruin with a reroute heuristic, and a random ruin with a reroute heuristic. The neighborhood size κ indicates the maximum length of the sequence or the maximum number of nodes moved from each route to some other. In all cases, we choose a random number between 1 and κ for the first randomly

chosen route, and a random number between 0 and κ for the second randomly chosen route.

The guiding idea of the cross-exchange operator [21] is to take two segments of different routes and exchange them; the icross-exchange operator [2] inverts the sequences. For cross and icross-exchange operators, we choose with the same probability between four possible variants of reinserting the segments either directly or being inverted. The second shaking operator is inspired by the large neighborhood search of Shaw [19]. We call it segment ruin and reroute. It consists of (i) selecting two routes randomly, (ii) removing a segment of nodes from each of the two routes, and (iii) iteratively reinserting the nodes removed at step (ii) in the solution. At each iteration of step (iii), one of the customers who is not in the solution is selected randomly, then inserted greedily. The third shaking operator, or random ruin and reroute, is similar to segment ruin and reroute except for the step (ii), when we select nodes randomly in the routes, not necessarily in sequence. The advantage of the latter ruin and reroute neighborhoods is the perturbation of more than two routes if the number of routes is high, or else a stronger perturbation in one route is done if the number of routes is small.

For an effective reduction of time window violations, we introduce an additional shaking step. If there are hard time window constraints that are violated, we move the customer with the highest time window violation to a randomly chosen route at a randomly chosen position. This special shaking step is performed every 1000 non-improving iterations if the best solution found so far has hard time window violations. It is performed before the regular shaking step.

Another special shaking step guarantees high perturbation of the incumbent feasible solution with a 2-opt* move (see Section 2.3). If there are $2000m$ non-improving iterations (m is the number of used vehicles),the last customers of two randomly chosen routes are exchanged. This shaking step is performed before the regular shaking step.

The shaking neighborhoods are scaled by the number of customers of a route k that are exchanged or moved. Let C_k denote the number of customers assigned to route k; then the maximum number of customers for each shaking move on route k is $\min(\kappa, C_k)$.

2.3. Local search

The solution obtained through shaking undergoes a local search procedure. The shaking steps focus on exchanging customers between routes, but the local search only searches for improvements among the routes that were modified in the shaking step.

We consider four intra-tour and inter-tour local search methods. After each shaking step, either 2-opt or a succession of cross-exchange and Or-opt moves is randomly chosen and performed; after that, 2-opt* is applied. Each method is performed in a first-improvement fashion until a local optimum is found.

In general, a 2-opt heuristic iteratively inverts sequences. To minimize CPU effort, we restrict the length of the inverted sequences to $\min(6, C_k - 1)$. A cross-exchange operation exchanges the sequences of customers between two routes .

Sequences up to a length of $\min(3, C_k - 1)$ are considered. An Or-opt local search iteratively moves subsequences up to a sequence length of three. A 2-opt* move exchanges the last parts of two routes. For a detailed description of local search methods for VRPs see [3].

2.4. Acceptance decision

After the shaking and the local search procedures have been performed, the current solution is compared with the incumbent solution to make the acceptance or rejection decision. If it accepts only improving solutions, the algorithm can easily get stuck, especially if the number of vehicles is restricted by the whole solution process. In many cases, it is also essential to have a strategy for accepting non-improving solutions (see e.g. Lourenço et al. [12]). We implement a more advanced acceptance decision for non-improving solutions, based on a threshold acceptance criterion used by Polacek et al. [16]. A solution yielding an improvement is always accepted. Ascending moves are accepted after a certain number of iterations, counted from the last accepted move, but only if the cost increase is below a certain threshold. In particular, we accept after 100 iterations without improvement a degradation of at most 10% of the current objective function value.

An important characteristic of our VNS algorithm is the ability to deal with infeasible solutions. Infeasibility occurs if the total capacity or tour duration exceeds a specific limit or if the time windows of the customers are violated. We penalize the degree of infeasibility of the set of routes and specify the evaluation function as:

$$f(x) = c(x) + \alpha c_\alpha(x) + \beta c_\beta(x) + \gamma c_\gamma(x). \tag{1}$$

The evaluation function $f(x)$ for the solution x is the sum of the total travel cost over all routes $c(x)$, the penalty terms for the violation of the capacity $c_\alpha(x)$, the violation of the route length $c_\beta(x)$, and the violation of time windows over all customers $c_\gamma(x)$, multiplied by the corresponding penalty parameters α, β, and γ. Following [16], we set the penalty parameters $\alpha = \beta = 100$. For problems with hard time windows, a penalty of 100 is not enough to guarantee feasible solutions in the end, so from the start, we choose a penalty of 200 as long as there is infeasibility due to tardiness. As soon as the solution becomes feasible in terms of tardiness, we increase the penalty to a high value to avoid undesired small time window violations. Through experiments, we found that a value of 1000 is high enough for the considered instances.

3. DIFFERENT ADAPTIVE STRATEGIES WITHIN VARIABLE NEIGHBORHOOD SEARCH

In this study we investigate different adaptive strategies based on adaptive VNS procedures we presented in Part I of this survey [10]. We focus on the following six adaptive mechanisms:

(A.1) *NhSize*: adapt the maximum neighborhood size,

(A.2) *Nh*: select the neighborhood size,
(A.3) *shaking*: select the shaking operator,
(A.4) *join_shaking_Nh*: select the shaking operator and the neighborhood size jointly,
(A.5) *indep_shaking_ls*: select the shaking and the local search operator independently,
(A.6) *join_shaking_ls*: select the shaking and the local search operator jointly.

For adapting the maximum neighborhood size in (A.1), we follow Hosny et al. [9]. The other cases (A.2) - (A.6) are solved and compared with two different adaptive mechanisms: The first adaptive mechanism of the VNS is performed with a scoring system. We call it AVNS-S. It is similar to the presented adaption mechanism of ALNS. Scores are added to the iterator x_i in the following way: (i) a score of six is added whenever a new overall best solution is found, (ii) a score of three is added if the current solution is improved, and (iii) a score of one is added if the solution is worse than the current but is accepted by the threshold acceptance criterion. The second adaptive mechanism is performed due to efficiency derived from Pillac et al. [15]. We call it AVNS-E. The efficiency of each neighborhood is measured by adding the improvement to the iterator x_i once if the current solution is improved, and twice, if the best found solution is improved. Neighborhoods with higher success are chosen frequently, while neighborhoods which lead to only few improvements are chosen rarely. For both mechanisms, the AVNS-S and the AVNS-E, we use a reaction factor $\rho = 0.1$ and the probabilities of choosing a neighborhood are uniformly distributed.

4. COMPUTATIONAL EXPERIMENTS

The algorithm is implemented in C++ and tested on two benchmark sets from prior literature: For the OVRP, we use the instances by Christofides et al. [4] with 50 to 199 customers, whereas for the OVRPTW, we use Solomon's Euclidean benchmark instances [20] with 100 customers clustered within a $[0, 100]^2$ square. We compare our results with the previous results obtained using the UVNS framework in [11] without adaptive mechanisms. For this study, we just consider the instances of group R, where the customer locations are randomly distributed, and the instances of class RC, where the customer locations are a mixture of the customer locations clustered in groups and those randomly distributed customer locations. We focus on these instances as they provide the highest potential for improvement.

All experiments are performed on an Intel(R) Xeon(R) CPU X5550 (2.67 GHz) running open SUSE 11.1. Most instances can be solved within a few seconds, but for instances with many customers to be served on a single route, several minutes may be necessary to receive results comparable to the best known or optimal solutions. Therefore, we stop the algorithm after ten minutes or $10000m^2$ non-improving iterations, where m is the number of vehicles, and the algorithm is run ten times on each instance.

In the following tables, we report the best and average performance of the particular adaptive mechanism and compare it with the best and average solutions of the UVNS in Section 2. The abbreviations of Tables 2 - 13 are explained in Table 1. We indicate in boldface the gap of the improved solution.

Table 1: Abbreviations of Tables 2 - 13

Abbreviation	Explanation
Avg.	average value
Cost	cost of the best found solution of UVNS
Best sol.	cost of the best found solution
Avg. sol.	average cost of all solutions obtained during all experiments
Best gap	gap from the best found solution to the best found solution of UVNS
Avg. gap	average gap from the average cost of all solutions obtained during all experiments to the best found solution of UVNS

Adapting the maximum neighborhood size (A.1)
A simple adaptive strategy is presented by Hosny et al. [9]. Besides an adaptable stopping condition controlled by the number of non-improving iterations, the maximum neighborhood size κ is not fixed, but it depends on the stage of the current VNS run considering multiple VNS runs. In the first run, κ is initialized with $2 \times \sqrt{n}$, where n is the total number of nodes. After a fixed number of iterations, or when no benefit seems to be realized, the current VNS run is stopped, the best found solution is chosen as initial solution for the next VNS run, and κ is reduced by one quarter of its initial value until the lower bound $\kappa/4$ is met. In other words, this multiple VNS run can be seen as one VNS run with reducing κ.

In our computational experiments, we perform ten independent VNS runs, each with a starting $\kappa = 2 \times \sqrt{n}$ and for the lower bound, we choose 8, the given κ in [11]. We reduce κ by its initial quarter after either two minutes of the solution process, or $1000 \times m^2$ of non-improving iterations, where m is the number of routes.

In Tables 2 and 3, we present the results of the VNS adapting the maximum neighborhood size. In OVRP, one of 14 instances can be improved by 0.21 %, but for five instances the best found solution of UVNS cannot be met. For larger instances, e.g., C05 with 199 customers and C09 with 150 customers, the best found solution is more than 2% and 1% worse than the best found solution of UVNS. Five of the 39 OVRPTW instances can be improved (see Table 3), e.g., R205 is improved by 0.50% and RC207 even by 0.73%.

Using the adaption of the maximum neighborhood size, several improvements are still possible but the average performance of 10 runs is not as promising.

Selecting the neighborhood size (A.2)
In this part, the shaking neighborhoods are scaled by the maximum number of customers of a route that are exchanged or moved. Instead of using the VNS defined strategy for adjusting the neighborhood size, the neighborhoods with high success should be called more often. This means, that the maximum number of customers is selected through this adaption strategy. We perform both adaptive

Table 2: Performance analysis on the OVRP instances by Christofides et al. [4] using *NhSize* (A.1)

	UVNS	AVNS			
	Cost	Best sol.	Avg. sol.	Best gap	Avg. gap
C01	416.06	416.06	416.06	**0.00%**	**0.00%**
C02	567.14	567.14	567.14	**0.00%**	**0.00%**
C03	640.42	640.42	641.84	**0.00%**	0.22%
C04	733.13	733.64	734.87	0.07%	0.24%
C05	907.53	912.77	928.02	0.58%	2.26%
C06	412.96	412.96	412.96	**0.00%**	**0.00%**
C07	583.19	583.19	583.30	**0.00%**	0.02%
C08	644.63	645.16	645.96	0.08%	0.21%
C09	757.96	758.24	767.50	0.04%	1.26%
C10	875.80	876.70	882.92	0.10%	0.81%
C11	682.12	682.12	682.39	**0.00%**	0.04%
C12	534.24	534.24	534.24	**0.00%**	**0.00%**
C13	904.04	902.11	905.31	**-0.21%**	0.14%
C14	591.87	591.87	591.87	**0.00%**	**0.00%**
Avg.	660.79	661.19	663.88	0.06%	0.47%

Table 3: Performance analysis on the OVRPTW instances by Solomon [20] using *NhSize* (A.1)

	UVNS	AVNS			
	Cost	Best sol.	Avg. sol.	Best gap	Avg. gap
R101	1192.85	1192.85	1192.85	**0.00%**	**0.00%**
R102	1079.39	1079.39	1079.39	**0.00%**	**0.00%**
R103	1016.78	1016.78	1016.81	**0.00%**	**0.00%**
R104	832.50	834.44	839.67	0.23%	0.86%
R105	1055.04	1055.04	1055.12	**0.00%**	**0.01%**
R106	1000.48	1001.14	1002.10	0.07%	0.16%
R107	910.75	910.75	914.36	**0.00%**	0.40%
R108	759.86	760.30	762.66	0.06%	0.37%
R109	934.15	934.15	934.64	**0.00%**	0.05%
R110	873.75	873.75	880.79	**0.00%**	0.81%
R111	895.21	896.48	906.90	0.14%	1.31%
R112	802.92	803.83	814.10	0.11%	1.39%
Avg.	946.14	946.57	949.95	0.05%	0.40%
RC101	1227.37	1227.37	1227.37	**0.00%**	**0.00%**
RC102	1185.43	1185.43	1190.48	**0.00%**	0.43%
RC103	918.65	918.65	918.65	**0.00%**	**0.00%**
RC104	787.02	787.02	789.14	**0.00%**	0.27%
RC105	1195.20	1195.20	1201.08	**0.00%**	0.49%
RC106	1071.83	1071.83	1075.42	**0.00%**	0.33%
RC107	860.62	860.62	862.68	**0.00%**	0.24%
RC108	831.09	833.03	836.58	0.23%	0.66%
Avg.	1009.65	1009.89	1012.68	0.02%	0.30%
R201	1182.43	1182.43	1185.45	**0.00%**	0.26%
R202	1150.24	1151.16	1151.48	0.08%	0.11%
R203	891.22	895.27	898.79	0.45%	0.85%
R204	801.23	802.95	819.31	0.21%	2.26%
R205	952.72	948.00	962.35	**-0.50%**	1.01%
R206	870.98	871.53	880.65	0.06%	1.11%
R207	854.40	858.33	887.57	0.46%	3.88%
R208	698.84	707.25	714.31	1.20%	2.21%
R209	851.69	851.69	862.99	**0.00%**	1.33%
R210	899.27	904.22	909.21	0.55%	1.11%
R211	853.65	851.80	869.39	**-0.22%**	1.84%
Avg.	909.70	911.33	921.95	0.18%	1.35%
RC201	1304.50	1310.31	1318.23	0.45%	1.05%
RC202	1289.04	1290.18	1312.69	0.09%	1.83%
RC203	993.22	993.76	1001.51	0.05%	0.83%
RC204	721.67	720.49	723.59	**-0.16%**	0.27%
RC205	1189.84	1189.84	1190.16	**0.00%**	0.03%
RC206	1088.85	1091.79	1095.35	0.27%	0.60%
RC207	1006.06	998.70	1007.75	**-0.73%**	0.17%
RC208	770.81	769.40	780.47	**-0.18%**	1.25%
Avg.	1045.50	1045.56	1053.72	0.01%	0.79%

Table 4: Performance analysis on the OVRP instances by Christofides et al. [4] using *Nh* (A.2)

	UVNS	AVNS-S				AVNS-E			
	Cost	Best sol.	Avg. sol.	Best gap	Avg. gap	Best sol.	Avg. sol.	Best gap	Avg. gap
C01	416.06	416.06	416.06	**0.00%**	**0.00%**	416.06	416.06	**0.00%**	**0.00%**
C02	567.14	567.14	567.14	**0.00%**	**0.00%**	567.14	567.14	**0.00%**	**0.00%**
C03	640.42	640.14	641.16	**-0.04%**	0.11%	640.42	641.24	**0.00%**	0.13%
C04	733.13	733.13	734.17	**0.00%**	0.14%	733.64	734.82	0.07%	0.23%
C05	907.53	914.54	926.79	0.77%	2.12%	908.36	922.94	0.09%	1.70%
C06	412.96	412.96	412.96	**0.00%**	**0.00%**	412.96	412.96	**0.00%**	**0.00%**
C07	583.19	583.19	583.19	**0.00%**	**0.00%**	583.19	583.24	**0.00%**	0.01%
C08	644.63	644.63	645.22	**0.00%**	0.09%	644.63	645.27	**0.00%**	0.10%
C09	757.96	757.91	761.35	**-0.01%**	0.45%	757.73	761.20	**-0.03%**	0.43%
C10	875.80	875.23	881.92	**-0.06%**	0.70%	876.81	881.14	0.12%	0.61%
C11	682.12	682.12	682.47	**0.00%**	0.05%	682.12	682.61	**0.00%**	0.07%
C13	534.24	534.24	534.24	**0.00%**	**0.00%**	534.24	534.24	**0.00%**	**0.00%**
C12	904.04	900.17	906.51	**-0.43%**	0.27%	902.11	906.87	**-0.21%**	0.31%
C14	591.87	591.87	591.87	**0.00%**	**0.00%**	591.87	591.87	**0.00%**	**0.00%**
Avg.	660.79	660.95	663.22	0.02%	0.37%	660.81	662.97	0.00%	0.33%

Table 5: Performance analysis on the OVRPTW instances by Solomon [20] using *Nh* (A.2)

	UVNS	AVNS-S				AVNS-E			
	Cost	Best sol.	Avg. sol.	Best gap	Avg. gap	Best sol.	Avg. sol.	Best gap	Avg. gap
R101	1192.85	1192.85	1192.85	**0.00%**	**0.00%**	1192.85	1192.85	**0.00%**	**0.00%**
R102	1079.39	1079.39	1079.39	**0.00%**	**0.00%**	1079.39	1079.39	**0.00%**	**0.00%**
R103	1016.78	1016.78	1016.78	**0.00%**	**0.00%**	1016.78	1016.78	**0.00%**	**0.00%**
R104	832.50	834.94	837.51	0.29%	0.60%	832.50	835.30	**0.00%**	0.34%
R105	1055.04	1055.04	1055.04	**0.00%**	**0.00%**	1055.04	1055.04	**0.00%**	**0.00%**
R106	1000.48	1000.95	1001.19	0.05%	0.07%	1000.48	1001.24	**0.00%**	0.08%
R107	910.75	910.75	913.47	**0.00%**	0.30%	910.75	914.36	**0.00%**	0.40%
R108	759.86	760.30	760.30	0.06%	0.06%	760.30	760.30	0.06%	0.06%
R109	934.15	934.15	934.41	**0.00%**	0.03%	934.15	934.30	**0.00%**	0.02%
R110	873.75	873.75	875.80	**0.00%**	0.23%	873.75	877.15	**0.00%**	0.39%
R111	895.21	895.21	897.29	**0.00%**	0.23%	895.21	898.06	**0.00%**	0.32%
R112	802.92	802.92	808.71	**0.00%**	0.72%	802.77	808.02	**-0.02%**	0.64%
Avg.	946.14	946.42	947.73	0.03%	0.17%	946.16	947.73	**0.00%**	0.17%
RC101	1227.37	1227.37	1227.37	**0.00%**	**0.00%**	1227.37	1227.37	**0.00%**	**0.00%**
RC102	1185.43	1185.43	1188.19	**0.00%**	0.23%	1185.43	1189.46	**0.00%**	0.34%
RC103	918.65	918.65	918.65	**0.00%**	**0.00%**	918.65	918.65	**0.00%**	**0.00%**
RC104	787.02	787.02	787.02	**0.00%**	**0.00%**	787.02	787.02	**0.00%**	**0.00%**
RC105	1195.20	1195.20	1196.36	**0.00%**	0.10%	1195.20	1196.59	**0.00%**	0.12%
RC106	1071.83	1071.83	1072.11	**0.00%**	0.03%	1071.83	1071.83	**0.00%**	**0.00%**
RC107	860.62	861.28	861.97	0.08%	0.16%	861.28	861.62	0.08%	0.12%
RC108	831.09	831.09	832.54	**0.00%**	0.18%	831.09	833.09	**0.00%**	0.24%
Avg.	1009.65	1009.73	1010.53	0.01%	0.09%	1009.73	1010.70	0.01%	0.10%
R201	1182.43	1182.43	1185.32	**0.00%**	0.24%	1182.43	1182.96	**0.00%**	0.04%
R202	1150.24	1151.16	1151.53	0.08%	0.11%	1151.12	1151.65	0.08%	0.12%
R203	891.22	892.51	895.96	0.14%	0.53%	895.24	896.54	0.45%	0.60%
R204	801.23	801.22	811.05	**0.00%**	1.23%	803.50	812.16	0.28%	1.36%
R205	952.72	952.72	960.03	**0.00%**	0.77%	953.33	958.61	0.06%	0.62%
R206	870.98	865.92	878.40	**-0.58%**	0.85%	873.67	879.66	0.31%	1.00%
R207	854.40	856.06	866.94	0.19%	1.47%	857.08	866.19	0.31%	1.38%
R208	698.84	699.08	706.74	0.03%	1.13%	699.15	703.55	0.04%	0.67%
R209	851.69	858.37	861.70	0.78%	1.18%	853.62	858.69	0.23%	0.82%
R210	899.27	895.37	905.21	**-0.43%**	0.66%	890.02	902.68	**-1.03%**	0.38%
R211	853.65	853.65	871.93	**0.00%**	2.14%	857.06	877.82	0.40%	2.83%
Avg.	909.70	909.86	917.71	0.02%	0.88%	910.56	917.32	0.10%	0.84%
RC201	1304.50	1303.73	1314.52	**-0.06%**	0.77%	1304.50	1316.43	**0.00%**	0.91%
RC202	1289.04	1289.04	1315.33	**0.00%**	2.04%	1290.18	1311.17	0.09%	1.72%
RC203	993.22	994.84	1001.22	0.16%	0.81%	993.22	1001.03	**0.00%**	0.79%
RC204	721.67	720.49	724.87	**-0.16%**	0.44%	720.38	724.65	**-0.18%**	0.41%
RC205	1189.84	1189.84	1190.38	**0.00%**	0.05%	1189.84	1189.91	**0.00%**	0.01%
RC206	1088.85	1092.40	1097.56	0.33%	0.80%	1087.97	1097.51	**-0.08%**	0.80%
RC207	1006.06	1006.06	1010.67	**0.00%**	0.46%	1001.46	1010.00	**-0.46%**	0.39%
RC208	770.81	772.22	787.21	0.18%	2.13%	770.60	784.00	**-0.03%**	1.71%
Avg.	1045.50	1046.08	1055.22	0.06%	0.93%	1044.77	1054.34	**-0.07%**	0.85%

mechanisms, the AVNS-S and the AVNS-E, and it turns out that for the considered instance classes, the average performance of the AVNS-E is slightly better. For the OVRP instances in Table 4, two instances can be improved compared to the UVNS, and for the OVRPTW in Table 5, six instances can be improved. Even R210 can be improved by 1.03%. For the instance class RC2 an average improvement of 0.07% is obtained.

Using the selection of neighborhood size, AVNS-E performs slightly better than AVNS-S. Especially for instance class RC2 an average improvement of 0.07% can be achieved with AVNS-E.

Selecting shaking operator (A.3)
In the original UVNS the shaking and local search operators are chosen randomly. In this study, we adapt the selection of the shaking operators, again with the AVNS-S and the AVNS-E. We select the shaking operator due to their past performance, either with the adaption based on scores or based on efficiency. In both cases, the selection of local search operators is still random. As it is shown in Tables 6 and 7, the OVRP and OVRPTW, the AVNS-E obtains slightly better results than the AVNS-S.

Using the selection of neighborhood size, AVNS-E performs slightly better than AVNS-S. Especially for instance class R2 an average improvement of 0.06% can be achieved with AVNS-E.

Selecting the shaking operator and the neighborhood size jointly (A.4)
A combination of choosing the neighborhood size and selecting the shaking operators, leads to these findings. In Tables 8 and 9 we show that the maximum number of customers of a route that are exchanged or moved has not a high influence on the shaking operator that is used, and vice versa.

Using the joint selection of the shaking operator and neighborhood size, AVNS-S performs better than AVNS-E for the OVRP on the contrary to the OVRPTW. Summarized an improvement of seven instances can be achieved either with AVNS-S or AVNS-E.

Selecting the shaking and the local search operators independently (A.5)
We are also interested in selecting both, the shaking operators as well as the local search operators due to their success in the previous performance. We study the selection of the operator classes independently, as it is usually done in the literature. As an acceptance decision is made after a shaking and a local search step, one can assume that the interplay between these operators will have a high impact on the decision. Therefore, it is not surprising that less improvement is obtained in Tables 10 and 11. One solution of the OVRP instances can be improved with the AVNS-S as well as the AVNS-E, and one solution of the OVRPTW instances can also be improved with the AVNS-S and the AVNS-E, respectively.

Using the independent selection of the shaking and local search operators, the improvements are not promising.

Table 6: Performance analysis on the OVRP instances by Christofides et al. [4] using *shaking* (A.3)

	UVNS	AVNS-S				AVNS-E			
	Cost	Best sol.	Avg. sol.	Best gap	Avg. gap	Best sol.	Avg. sol.	Best gap	Avg. gap
C01	416.06	416.06	416.06	**0.00%**	**0.00%**	416.06	416.06	**0.00%**	**0.00%**
C02	567.14	567.14	567.14	**0.00%**	**0.00%**	567.14	567.14	**0.00%**	**0.00%**
C03	640.42	640.42	642.34	**0.00%**	0.30%	640.86	641.74	0.07%	0.21%
C04	733.13	733.68	735.45	0.08%	0.32%	733.64	735.09	0.07%	0.27%
C05	907.53	922.74	933.28	1.68%	2.84%	914.02	930.39	0.72%	2.52%
C06	412.96	412.96	412.96	**0.00%**	**0.00%**	412.96	412.96	**0.00%**	**0.00%**
C07	583.19	583.19	583.24	**0.00%**	0.01%	583.19	583.34	**0.00%**	0.03%
C08	644.63	644.63	645.31	**0.00%**	0.10%	644.63	645.72	**0.00%**	0.17%
C09	757.96	757.95	761.96	**0.00%**	0.53%	758.24	765.37	0.04%	0.98%
C10	875.80	879.10	887.52	0.38%	1.34%	877.23	888.45	0.16%	1.44%
C11	682.12	682.12	682.69	**0.00%**	0.08%	682.12	682.83	**0.00%**	0.10%
C13	534.24	534.24	534.24	**0.00%**	**0.00%**	534.24	534.24	**0.00%**	**0.00%**
C12	904.04	902.11	908.90	**-0.21%**	0.54%	902.11	909.03	**-0.21%**	0.55%
C14	591.87	591.87	591.87	**0.00%**	**0.00%**	591.87	591.87	**0.00%**	**0.00%**
Avg.	660.79	662.02	664.50	0.19%	0.56%	661.31	664.59	0.08%	0.57%

Table 7: Performance analysis on the OVRPTW instances by Solomon [20] using *shaking* (A.3)

	UVNS	AVNS-S				AVNS-E			
	Cost	Best sol.	Avg. sol.	Best gap	Avg. gap	Best sol.	Avg. sol.	Best gap	Avg. gap
R101	1192.85	1192.85	1192.85	**0.00%**	**0.00%**	1192.85	1192.85	**0.00%**	**0.00%**
R102	1079.39	1079.39	1079.39	**0.00%**	**0.00%**	1079.39	1079.39	**0.00%**	**0.00%**
R103	1016.78	1016.78	1016.80	**0.00%**	**0.00%**	1016.78	1016.78	**0.00%**	**0.00%**
R104	832.50	834.44	837.75	0.23%	0.63%	832.50	836.31	**0.00%**	0.46%
R105	1055.04	1055.04	1055.04	**0.00%**	**0.00%**	1055.04	1055.04	**0.00%**	**0.00%**
R106	1000.48	1000.68	1001.10	0.02%	0.06%	1001.14	1001.19	0.07%	0.07%
R107	910.75	910.75	913.76	**0.00%**	0.33%	910.75	913.28	**0.00%**	0.28%
R108	759.86	760.30	762.33	0.06%	0.33%	760.30	762.84	0.06%	0.39%
R109	934.15	934.15	934.38	**0.00%**	0.02%	934.15	934.42	**0.00%**	0.03%
R110	873.75	873.75	875.69	**0.00%**	0.22%	873.75	875.64	**0.00%**	0.22%
R111	895.21	895.21	896.52	**0.00%**	0.15%	895.21	897.00	**0.00%**	0.20%
R112	802.92	803.93	807.07	0.13%	0.52%	803.38	809.29	0.06%	0.79%
Avg.	946.14	946.44	947.72	0.03%	0.17%	946.27	947.84	0.01%	0.18%
RC101	1227.37	1227.37	1227.37	**0.00%**	**0.00%**	1227.37	1227.37	**0.00%**	**0.00%**
RC102	1185.43	1185.43	1188.31	**0.00%**	0.24%	1185.43	1191.43	**0.00%**	0.51%
RC103	918.65	918.65	918.65	**0.00%**	**0.00%**	918.65	918.65	**0.00%**	**0.00%**
RC104	787.02	787.02	787.02	**0.00%**	**0.00%**	787.02	787.55	**0.00%**	0.07%
RC105	1195.20	1195.20	1196.13	**0.00%**	0.08%	1195.20	1196.59	**0.00%**	0.12%
RC106	1071.83	1071.83	1073.24	**0.00%**	0.13%	1071.83	1073.51	**0.00%**	0.16%
RC107	860.62	860.62	862.03	**0.00%**	0.16%	861.28	862.98	0.08%	0.27%
RC108	831.09	831.09	832.90	**0.00%**	0.22%	831.09	834.79	**0.00%**	0.45%
Avg.	1009.65	1009.65	1010.71	**0.00%**	0.10%	1009.73	1011.61	0.01%	0.19%
R201	1182.43	1182.43	1185.22	**0.00%**	0.24%	1182.43	1184.29	**0.00%**	0.16%
R202	1150.24	1151.14	1151.36	0.08%	0.10%	1149.59	1151.16	**-0.06%**	0.08%
R203	891.22	892.51	895.40	0.14%	0.47%	890.65	894.63	**-0.06%**	0.38%
R204	801.23	801.23	811.02	**0.00%**	1.22%	803.34	812.51	**0.26%**	1.41%
R205	952.72	952.83	958.00	**0.01%**	0.55%	949.38	954.12	**-0.35%**	0.15%
R206	870.98	871.76	877.06	**0.09%**	0.70%	870.98	875.78	**0.00%**	0.55%
R207	854.40	856.06	868.60	0.19%	1.66%	856.02	864.47	0.19%	1.18%
R208	698.84	699.08	704.64	0.03%	0.83%	699.15	703.34	**0.04%**	0.64%
R209	851.69	853.53	859.49	0.22%	0.92%	851.69	857.67	**0.00%**	0.70%
R210	899.27	899.21	903.75	**-0.01%**	0.50%	896.58	903.06	**-0.30%**	0.42%
R211	853.65	861.89	870.89	0.97%	2.02%	850.88	869.79	**-0.32%**	1.89%
Avg.	909.70	911.06	916.86	0.15%	0.79%	909.15	915.53	**-0.06%**	0.64%
RC201	1304.50	1311.79	1319.47	0.56%	1.15%	1310.31	1318.48	0.45%	1.07%
RC202	1289.04	1289.04	1314.93	**0.00%**	2.01%	1290.18	1311.08	0.09%	1.71%
RC203	993.22	993.76	1001.88	0.05%	0.87%	993.08	998.43	**-0.01%**	0.52%
RC204	721.67	720.49	724.61	**-0.16%**	0.41%	721.34	726.46	**-0.05%**	0.66%
RC205	1189.84	1189.84	1190.48	**0.00%**	0.05%	1189.84	1191.07	**0.00%**	0.10%
RC206	1088.85	1092.42	1094.70	0.33%	0.54%	1088.85	1093.36	**0.00%**	0.41%
RC207	1006.06	1006.06	1009.99	**0.00%**	0.39%	1001.46	1010.05	**-0.46%**	0.40%
RC208	770.81	775.96	781.97	0.67%	1.45%	782.53	784.16	1.52%	1.73%
Avg.	1045.50	1047.42	1054.75	0.18%	0.89%	1047.20	1054.14	0.16%	0.83%

Table 8: Performance analysis on the OVRP instances by Christofides et al. [4] using *join_shaking_Nh* (A.4)

	UVNS	AVNS-S				AVNS-E			
	Cost	Best sol.	Avg. sol.	Best gap	Avg. gap	Best sol.	Avg. sol.	Best gap	Avg. gap
C01	416.06	416.06	416.06	**0.00%**	**0.00%**	416.06	416.06	**0.00%**	**0.00%**
C02	567.14	567.14	567.14	**0.00%**	**0.00%**	567.14	567.14	**0.00%**	**0.00%**
C03	640.42	640.42	641.95	**0.00%**	0.24%	640.42	642.45	**0.00%**	0.32%
C04	733.13	734.39	736.76	0.17%	0.50%	734.97	736.81	0.25%	0.50%
C05	907.53	904.57	926.00	**-0.33%**	2.04%	916.09	923.97	0.94%	1.81%
C06	412.96	412.96	412.96	**0.00%**	**0.00%**	412.96	412.96	**0.00%**	**0.00%**
C07	583.19	583.19	583.45	**0.00%**	0.04%	583.19	583.35	**0.00%**	0.03%
C08	644.63	644.63	645.54	**0.00%**	0.14%	644.63	645.91	**0.00%**	0.20%
C09	757.96	759.48	767.67	0.20%	1.28%	758.17	763.86	0.03%	0.78%
C10	875.80	880.06	887.95	0.49%	1.39%	876.70	881.32	0.10%	0.63%
C11	682.12	682.12	682.75	**0.00%**	0.09%	682.12	683.40	**0.00%**	0.19%
C12	534.24	534.24	534.24	**0.00%**	**0.00%**	534.24	534.27	**0.00%**	0.01%
C13	904.04	902.11	907.75	**-0.21%**	0.41%	902.11	906.21	**-0.21%**	0.24%
C14	591.87	591.87	591.87	**0.00%**	**0.00%**	591.87	591.87	**0.00%**	**0.00%**
Avg.	660.79	660.95	664.43	0.02%	0.55%	661.48	663.54	0.10%	0.42%

Table 9: Performance analysis on the OVRPTW instances by Solomon [20] using *join_shaking_Nh* (A.4)

	UVNS	AVNS-S				AVNS-E			
	Cost	Best sol.	Avg. sol.	Best gap	Avg. gap	Best sol.	Avg. sol.	Best gap	Avg. gap
R101	1192.85	1192.85	1192.85	**0.00%**	**0.00%**	1192.85	1192.85	**0.00%**	**0.00%**
R102	1079.39	1079.39	1079.39	**0.00%**	**0.00%**	1079.39	1079.39	**0.00%**	**0.00%**
R103	1016.78	1016.78	1016.82	**0.00%**	**0.00%**	1016.78	1016.79	**0.00%**	**0.00%**
R104	832.50	832.50	840.61	**0.00%**	0.97%	834.44	839.41	0.23%	0.83%
R105	1055.04	1055.04	1055.19	**0.00%**	**0.01%**	1055.04	1055.04	**0.00%**	**0.00%**
R106	1000.48	1000.48	1001.73	**0.00%**	0.12%	1000.48	1001.32	**0.00%**	0.08%
R107	910.75	913.48	916.98	0.30%	0.68%	910.75	914.38	**0.00%**	0.40%
R108	759.86	760.30	764.73	0.06%	0.64%	760.30	764.29	0.06%	0.58%
R109	934.15	934.15	938.05	**0.00%**	0.42%	934.15	934.55	**0.00%**	0.04%
R110	873.75	873.75	883.34	**0.00%**	1.10%	873.75	875.83	**0.00%**	0.24%
R111	895.21	895.56	904.36	0.04%	1.02%	895.21	897.29	**0.00%**	0.23%
R112	802.92	807.21	817.41	0.54%	1.80%	803.38	809.86	0.06%	0.87%
Avg.	946.14	946.79	950.95	0.07%	0.51%	946.38	948.42	0.03%	0.24%
RC101	1227.37	1227.37	1227.37	**0.00%**	**0.00%**	1227.37	1227.37	**0.00%**	**0.00%**
RC102	1185.43	1185.43	1188.32	**0.00%**	0.24%	1185.43	1187.67	**0.00%**	0.19%
RC103	918.65	918.65	919.20	**0.00%**	0.06%	918.65	918.65	**0.00%**	**0.00%**
RC104	787.02	787.02	789.97	**0.00%**	0.37%	787.02	787.02	**0.00%**	**0.00%**
RC105	1195.20	1195.20	1197.57	**0.00%**	0.20%	1195.20	1196.13	**0.00%**	0.08%
RC106	1071.83	1071.83	1075.24	**0.00%**	0.32%	1071.83	1071.83	**0.00%**	**0.00%**
RC107	860.62	861.28	862.64	0.08%	0.23%	860.62	861.86	**0.00%**	0.14%
RC108	831.09	831.09	835.50	**0.00%**	0.53%	831.09	833.58	**0.00%**	0.30%
Avg.	1009.65	1009.73	1011.98	0.01%	0.23%	1009.65	1010.51	**0.00%**	0.09%
R201	1182.43	1182.43	1187.63	**0.00%**	0.44%	1182.43	1185.58	**0.00%**	0.27%
R202	1150.24	1151.16	1152.02	0.08%	0.15%	1150.67	1151.28	0.04%	0.09%
R203	891.22	894.17	899.00	0.33%	0.87%	889.12	895.39	**-0.24%**	0.47%
R204	801.23	806.05	816.93	0.60%	1.96%	809.61	814.07	1.05%	1.60%
R205	952.72	953.33	964.33	0.06%	1.22%	948.00	956.06	**-0.50%**	0.35%
R206	870.98	873.21	881.25	0.26%	1.18%	872.01	875.40	0.12%	0.51%
R207	854.40	868.54	887.51	1.66%	3.87%	857.08	871.97	0.31%	2.06%
R208	698.84	699.08	710.02	0.03%	1.60%	699.15	705.55	0.04%	0.96%
R209	851.69	851.69	860.17	**0.00%**	1.00%	851.69	856.94	**0.00%**	0.62%
R210	899.27	894.54	905.44	**-0.53%**	0.69%	899.99	903.70	0.08%	0.49%
R211	853.65	864.22	881.25	1.24%	3.23%	864.20	873.69	1.24%	2.35%
Avg.	909.70	912.58	922.32	0.32%	1.39%	911.27	917.24	0.17%	0.83%
RC201	1304.50	1314.32	1320.96	0.75%	1.26%	1304.50	1317.57	**0.00%**	1.00%
RC202	1289.04	1303.75	1340.41	1.14%	3.99%	1289.04	1319.41	**0.00%**	2.36%
RC203	993.22	994.25	1007.94	0.10%	1.48%	993.22	1000.51	**0.00%**	0.73%
RC204	721.67	720.15	732.51	**-0.21%**	1.50%	724.52	729.88	0.39%	1.14%
RC205	1189.84	1189.84	1190.48	**0.00%**	0.05%	1189.84	1191.39	**0.00%**	0.13%
RC206	1088.85	1092.42	1099.47	0.33%	0.98%	1092.42	1095.23	0.33%	0.59%
RC207	1006.06	1005.01	1015.88	**-0.10%**	0.98%	1005.01	1009.90	**-0.10%**	0.38%
RC208	770.81	775.58	788.81	0.62%	2.33%	777.85	785.08	0.91%	1.85%
Avg.	1045.50	1049.42	1062.06	0.37%	1.58%	1047.05	1056.12	0.15%	1.02%

Table 10: Performance analysis on the OVRP instances by Christofides et al. [4] using *indep_shaking_ls* (A.5)

	UVNS	AVNS-S				AVNS-E			
	Cost	Best sol.	Avg. sol.	Best gap	Avg. gap	Best sol.	Avg. sol.	Best gap	Avg. gap
C01	416.06	416.06	416.06	**0.00%**	**0.00%**	416.06	416.06	**0.00%**	**0.00%**
C02	567.14	567.14	567.14	**0.00%**	**0.00%**	567.14	567.14	**0.00%**	**0.00%**
C03	640.42	641.45	642.84	0.16%	0.38%	640.86	642.72	0.07%	0.36%
C04	733.13	734.93	738.89	0.24%	0.79%	734.92	736.85	0.24%	0.51%
C05	907.53	920.24	936.08	1.40%	3.15%	915.29	933.41	0.86%	2.85%
C06	412.96	412.96	412.96	**0.00%**	**0.00%**	412.96	412.96	**0.00%**	**0.00%**
C07	583.19	583.19	583.46	**0.00%**	0.05%	583.19	584.04	**0.00%**	0.15%
C08	644.63	644.63	645.92	**0.00%**	0.20%	644.63	646.08	**0.00%**	0.23%
C09	757.96	760.19	765.62	0.29%	1.01%	760.36	765.84	0.32%	1.04%
C10	875.80	884.75	892.93	1.02%	1.96%	881.07	891.57	0.60%	1.80%
C11	682.12	682.12	682.88	**0.00%**	0.11%	682.12	689.44	**0.00%**	1.07%
C12	534.24	534.24	534.24	**0.00%**	**0.00%**	534.24	534.26	**0.00%**	**0.00%**
C13	904.04	902.11	908.39	**-0.21%**	0.48%	902.44	913.01	**-0.18%**	0.99%
C14	591.87	591.87	591.87	**0.00%**	**0.00%**	591.87	591.87	**0.00%**	**0.00%**
Avg.	660.79	662.56	665.66	0.27%	0.74%	661.94	666.09	0.17%	0.80%

Table 11: Performance analysis on the OVRPTW instances by Solomon [20] using *indep_shaking_ls* (A.5)

	UVNS	AVNS-S				AVNS-E			
	Cost	Best sol.	Avg. sol.	Best gap	Avg. gap	Best sol.	Avg. sol.	Best gap	Avg. gap
R101	1192.85	1192.85	1192.90	**0.00%**	**0.00%**	1192.85	1192.85	**0.00%**	**0.00%**
R102	1079.39	1079.39	1079.39	**0.00%**	**0.00%**	1079.39	1079.39	**0.00%**	**0.00%**
R103	1016.78	1016.78	1016.80	**0.00%**	**0.00%**	1016.78	1016.79	**0.00%**	**0.00%**
R104	832.50	834.44	841.13	0.23%	1.04%	835.09	855.85	0.31%	2.80%
R105	1055.04	1055.04	1055.34	**0.00%**	0.03%	1055.04	1055.35	**0.00%**	0.03%
R106	1000.48	1001.14	1002.28	0.07%	0.18%	1001.14	1001.67	0.07%	0.12%
R107	910.75	910.75	916.41	**0.00%**	0.62%	910.75	916.49	**0.00%**	0.63%
R108	759.86	760.30	768.99	0.06%	1.20%	760.30	765.86	0.06%	0.79%
R109	934.15	934.15	940.94	**0.00%**	0.73%	934.15	937.15	**0.00%**	0.32%
R110	873.75	873.75	884.45	**0.00%**	1.23%	873.75	878.03	**0.00%**	0.49%
R111	895.21	895.21	914.43	**0.00%**	2.15%	895.21	908.56	**0.00%**	1.49%
R112	802.92	802.92	811.21	**0.00%**	1.03%	803.83	820.53	0.11%	2.19%
Avg.	946.14	946.39	952.02	0.03%	0.62%	946.52	952.38	0.04%	0.66%
RC101	1227.37	1227.37	1227.37	**0.00%**	**0.00%**	1227.37	1227.37	**0.00%**	**0.00%**
RC102	1185.43	1185.44	1203.12	**0.00%**	1.49%	1185.43	1198.77	**0.00%**	1.13%
RC103	918.65	918.65	918.65	**0.00%**	**0.00%**	918.65	919.02	**0.00%**	0.04%
RC104	787.02	787.02	792.60	**0.00%**	0.71%	787.02	792.49	**0.00%**	0.69%
RC105	1195.20	1195.20	1197.72	**0.00%**	0.21%	1195.20	1199.07	**0.00%**	0.32%
RC106	1071.83	1071.83	1074.91	**0.00%**	0.29%	1071.83	1079.06	**0.00%**	0.67%
RC107	860.62	860.62	862.57	**0.00%**	0.23%	861.28	862.94	0.08%	0.27%
RC108	831.09	831.09	836.95	**0.00%**	0.71%	831.09	839.14	**0.00%**	0.97%
Avg.	1009.65	1009.65	1014.24	**0.00%**	0.45%	1009.73	1014.73	0.01%	0.50%
R201	1182.43	1182.43	1187.86	**0.00%**	0.46%	1184.88	1187.32	0.21%	0.41%
R202	1150.24	1151.14	1152.40	0.08%	0.19%	1151.39	1152.19	0.10%	0.17%
R203	891.22	895.24	899.23	0.45%	0.90%	895.57	898.10	0.49%	0.77%
R204	801.23	805.59	822.31	0.54%	2.63%	808.74	820.76	0.94%	2.44%
R205	952.72	954.55	962.06	0.19%	0.98%	954.66	964.04	0.20%	1.19%
R206	870.98	867.55	879.76	**-0.39%**	1.01%	876.38	881.31	0.62%	1.19%
R207	854.40	861.22	885.35	0.80%	3.62%	871.57	893.01	2.01%	4.52%
R208	698.84	704.92	711.75	0.87%	1.85%	706.60	713.95	1.11%	2.16%
R209	851.69	857.63	863.28	0.70%	1.36%	854.72	862.36	0.36%	1.25%
R210	899.27	902.67	907.29	0.38%	0.89%	902.12	909.63	0.32%	1.15%
R211	853.65	862.97	883.55	1.09%	3.50%	882.36	897.96	3.36%	5.19%
Avg.	909.70	913.26	923.17	0.39%	1.48%	917.18	925.51	0.82%	1.74%
RC201	1304.50	1310.31	1319.57	0.45%	1.16%	1320.21	1322.45	1.20%	1.38%
RC202	1289.04	1290.18	1330.26	0.09%	3.20%	1290.63	1342.41	0.12%	4.14%
RC203	993.22	1001.24	1007.75	0.81%	1.46%	993.22	1005.07	**0.00%**	1.19%
RC204	721.67	722.03	728.18	0.05%	0.90%	720.15	728.20	**-0.21%**	0.90%
RC205	1189.84	1189.84	1192.48	**0.00%**	0.22%	1189.84	1191.18	**0.00%**	0.11%
RC206	1088.85	1092.66	1097.37	0.35%	0.78%	1091.79	1101.76	0.27%	1.19%
RC207	1006.06	1006.06	1018.34	**0.00%**	1.22%	1008.58	1021.64	0.25%	1.55%
RC208	770.81	774.31	790.35	0.45%	2.53%	779.97	787.50	1.19%	2.16%
Avg.	1045.50	1048.33	1060.54	0.27%	1.44%	1049.30	1062.53	0.36%	1.63%

Table 12: Performance analysis on the OVRP instances by Christofides et al. [4] using *join_shaking_ls* (A.6)

	UVNS	AVNS-S				AVNS-E			
	Cost	Best sol.	Avg. sol.	Best gap	Avg. gap	Best sol.	Avg. sol.	Best gap	Avg. gap
C01	416.06	416.06	416.06	**0.00%**	**0.00%**	416.06	416.06	**0.00%**	**0.00%**
C02	567.14	567.14	567.14	**0.00%**	**0.00%**	567.14	567.14	**0.00%**	**0.00%**
C03	640.42	639.74	641.63	**-0.11%**	0.19%	640.14	641.89	**-0.04%**	0.23%
C04	733.13	733.13	734.67	**0.00%**	0.21%	733.85	734.94	0.10%	0.25%
C05	907.53	908.94	920.26	0.16%	1.40%	907.00	923.75	**-0.06%**	1.79%
C06	412.96	412.96	412.96	**0.00%**	**0.00%**	412.96	412.96	**0.00%**	**0.00%**
C07	583.19	583.19	583.30	**0.00%**	0.02%	583.19	583.24	**0.00%**	0.01%
C08	644.63	644.63	645.25	**0.00%**	0.10%	644.63	645.24	**0.00%**	0.09%
C09	757.96	757.91	762.80	**-0.01%**	0.64%	759.35	763.78	0.18%	0.77%
C10	875.80	878.53	885.09	0.31%	1.06%	877.47	883.86	0.19%	0.92%
C11	682.12	682.12	682.39	**0.00%**	0.04%	682.12	682.53	**0.00%**	0.06%
C12	534.24	534.24	534.24	**0.00%**	**0.00%**	534.24	534.24	**0.00%**	**0.00%**
C13	904.04	902.11	907.49	**-0.21%**	0.38%	902.11	906.72	**-0.21%**	0.30%
C14	591.87	591.87	591.87	**0.00%**	**0.00%**	591.87	591.87	**0.00%**	**0.00%**
Avg.	660.79	660.90	663.23	0.02%	0.37%	660.87	663.44	0.01%	0.40%

Table 13: Performance analysis on the OVRPTW instances by Solomon [20] using *join_shaking_ls* (A.6)

	UVNS	AVNS-S				AVNS-E			
	Cost	Best sol.	Avg. sol.	Best gap	Avg. gap	Best sol.	Avg. sol.	Best gap	Avg. gap
R101	1192.85	1192.85	1192.85	**0.00%**	**0.00%**	1192.85	1192.85	**0.00%**	**0.00%**
R102	1079.39	1079.39	1079.39	**0.00%**	**0.00%**	1079.39	1079.39	**0.00%**	**0.00%**
R103	1016.78	1016.78	1016.81	**0.00%**	**0.00%**	1016.78	1016.79	**0.00%**	**0.00%**
R104	832.50	835.09	842.06	0.31%	1.15%	834.437	836.225	0.23%	0.45%
R105	1055.04	1055.04	1055.34	**0.00%**	**0.03%**	1055.04	1055.04	**0.00%**	**0.00%**
R106	1000.48	1000.95	1001.89	0.05%	0.14%	1000.36	1001.01	**-0.01%**	0.05%
R107	910.75	910.75	915.60	**0.00%**	0.53%	910.747	913.635	**0.00%**	0.32%
R108	759.86	760.30	764.40	0.06%	0.60%	760.304	760.97	0.06%	0.15%
R109	934.15	934.15	934.78	**0.00%**	0.07%	934.149	936.367	**0.00%**	0.24%
R110	873.75	873.75	880.54	**0.00%**	0.78%	873.748	875.761	**0.00%**	0.23%
R111	895.21	895.56	910.90	0.04%	1.75%	895.209	897.738	**0.00%**	0.28%
R112	802.92	803.93	814.64	0.13%	1.46%	803.641	808.036	0.09%	0.64%
Avg.	946.14	946.55	950.77	0.04%	0.49%	946.39	947.82	0.03%	0.18%
RC101	1227.37	1227.37	1227.66	**0.00%**	**0.02%**	1227.37	1227.37	**0.00%**	**0.00%**
RC102	1185.43	1185.43	1186.81	**0.00%**	0.12%	1185.43	1190.13	**0.00%**	0.40%
RC103	918.65	918.65	919.20	**0.00%**	0.06%	918.65	918.65	**0.00%**	**0.00%**
RC104	787.02	787.02	790.43	**0.00%**	0.43%	787.02	787.85	**0.00%**	0.11%
RC105	1195.20	1195.20	1198.41	**0.00%**	0.27%	1195.20	1196.36	**0.00%**	0.10%
RC106	1071.83	1071.83	1075.10	**0.00%**	0.31%	1071.83	1071.83	**0.00%**	**0.00%**
RC107	860.62	861.28	863.05	0.08%	0.28%	860.62	861.56	**0.00%**	0.11%
RC108	831.09	831.09	833.05	**0.00%**	0.24%	833.03	835.95	0.23%	0.59%
Avg.	1009.65	1009.73	1011.71	0.01%	0.20%	1009.89	1011.21	0.02%	0.15%
R201	1182.43	1182.43	1184.13	**0.00%**	0.14%	1182.43	1184.94	**0.00%**	0.21%
R202	1150.24	1151.16	1151.43	0.08%	0.10%	1150.67	1151.51	0.04%	0.11%
R203	891.22	893.72	895.34	0.28%	0.46%	891.08	895.66	**-0.02%**	0.50%
R204	801.23	805.19	812.71	0.49%	1.43%	800.87	810.14	**-0.04%**	1.11%
R205	952.72	952.72	954.84	**0.00%**	0.22%	952.72	956.54	**0.00%**	0.40%
R206	870.98	868.59	875.84	**-0.27%**	0.56%	868.18	873.55	**-0.32%**	0.30%
R207	854.40	857.02	870.60	0.31%	1.90%	869.43	879.83	1.76%	2.98%
R208	698.84	699.44	705.44	0.09%	0.94%	699.44	704.21	**0.09%**	0.77%
R209	851.69	853.30	858.55	0.19%	0.81%	856.73	860.39	0.59%	1.02%
R210	899.27	895.11	902.49	**-0.46%**	0.36%	902.15	906.95	0.32%	0.85%
R211	853.65	851.59	873.93	**-0.24%**	2.38%	865.20	873.65	1.35%	2.34%
Avg.	909.70	910.02	916.85	0.04%	0.79%	912.63	917.94	0.32%	0.91%
RC201	1304.50	1303.73	1317.20	**-0.06%**	0.97%	1303.73	1314.68	**-0.06%**	0.78%
RC202	1289.04	1289.04	1312.13	**0.00%**	1.79%	1289.04	1299.59	**0.00%**	0.82%
RC203	993.22	994.84	1003.21	0.16%	1.01%	993.76	1003.54	0.05%	1.04%
RC204	721.67	721.12	723.49	**-0.08%**	0.25%	720.15	724.72	**-0.21%**	0.42%
RC205	1189.84	1189.84	1190.53	**0.00%**	0.06%	1189.84	1190.78	**0.00%**	0.08%
RC206	1088.85	1088.85	1094.26	0.00%	0.50%	1090.57	1094.67	0.16%	0.53%
RC207	1006.06	1003.36	1008.87	**-0.27%**	0.28%	1005.01	1011.94	**-0.10%**	0.58%
RC208	770.81	773.43	780.06	0.34%	1.20%	780.68	785.31	1.28%	1.88%
Avg.	1045.50	1045.53	1053.72	**0.00%**	0.79%	1046.60	1053.15	0.11%	0.73%

Selecting shaking and local search operators jointly (A.6)
As it is already mentioned, we discuss the selection of the shaking and local search operators jointly in this section. In contrast to two independent adaption progresses, we select a pair of one shaking and one local search operator at every iteration based on their previous performance. This strategy allows that the right local search operator is chosen according to the selected shaking operator. Tables 12 and 13 show that both adaptive strategies, the score-based and the efficiency-based mechanism, yield a high number of improved solutions and the best average performance over all considered instances.

Using the joint selection of the shaking and local search operators, the highest number of improved instances can be obtained.

5. CONCLUSION

This research paper addresses a numerical study that discusses different adaptive strategies. It shows that the inclusion of an adaptive mechanism within a local search-based algorithm can improve the solutions. We show that some adaptive strategies lead to promising results, but some mechanisms do not achieve the expected results. In Tables 14 and 15 we summarize the performance of the six considered adaptive strategies.

Table 14: Summary of the performance analysis on the OVRP instances by Christofides et al. [4]

	adaptive mechanism	number of improved sol.	Best gap	Avg. gap
(A.1) *NhSize*	AVNS	1	0.06%	0.47%
(A.2) *Nh*	AVNS-S	4	0.02%	0.37%
	AVNS-E	2	0.00%	0.33%
(A.3) *shaking*	AVNS-S	1	0.19%	0.56%
	AVNS-E	1	0.08%	0.57%
(A.4) *join_shaking_Nh*	AVNS-S	2	0.02%	0.55%
	AVNS-E	1	0.10%	0.42%
(A.5) *indep_shaking_ls*	AVNS-S	1	0.27%	0.74%
	AVNS-E	1	0.17%	0.80%
(A.6) *join_shaking_ls*	AVNS-S	3	0.02%	0.37%
	AVNS-E	3	0.01%	0.40%

For both, the OVRP and the OVRPTW, the independent selection of the shaking and local search operators (A.5) achieve the worst results in case of the number of improved instances, as well as the solution quality. But the joint selection of the shaking and local search operators (A.6) obtains the best results. Also the selection of the neighborhood (A.2) for both problems, and the selection of the shaking operator (A.3) for the OVRPTW should be considered for further research.

The following interesting fact can be noticed: if the shaking and local search operators (or the shaking operator and the neighborhood size) are adapted inde-

Table 15: Summary of the performance analysis on the OVRPTW instances by Solomon [20]

	adaptive mechanism	number of improved sol.	Best gap	Avg. gap
(A.1) *NhSize*	AVNS	5	0.07%	0.71%
(A.2) *Nh*	AVNS-S	4	0.03%	0.52%
	AVNS-E	6	0.01%	0.49%
(A.3) *shaking*	AVNS-S	2	0.09%	0.49%
	AVNS-E	8	0.03%	0.46%
(A.4) *join_shaking_Nh*	AVNS-S	3	0.19%	0.93%
	AVNS-E	3	0.09%	0.55%
(A.5) *indep_shaking_ls*	AVNS-S	1	0.17%	1.00%
	AVNS-E	1	0.31%	1.13%
(A.6) *join_shaking_ls*	AVNS-S	6	0.02%	0.57%
	AVNS-E	7	0.12%	0.49%

pendently, the performance of the AVNS-S is better than the performance of the AVNS-E, e.g., the overall best gap for AVNS-S of the *join_shaking_Nh* for the OVRP instance class, is 0.02%, while the overall best gap for AVNS-E is 0.10%. From the number of improved solutions of the OVRPTW instance class, we show that the AVNS-E yields higher solution quality compared to AVNS-S.

We conclude from the numerical study that an effective adaption mechanism should change just one parameter, or a part of the algorithm and not different independent ones simultaneously. An adaption mechanism implemented in an algorithm yields substantial improvements, but adapting independent operators does not lead to satisfying results.

As one of the next steps, the usage of efficiency based roulette wheel adaption may be considered and tested in future ALNS research.

Acknowledgements. This work received support from the Austrian Science Fund (FWF) under grant and L628-N15 (Translational Research Programs).

REFERENCES

[1] Bodin, L., Golden, B., Assad, A., Ball, M., "Routing and scheduling of vehicles and crews: the state of the art", *Computers & Operations Research*, 10(2) (1983) 195-211.

[2] Bräysy, O., "A Reactive Variable Neighborhood Search for the Vehicle-Routing Problem with Time Windows", *INFORMS Journal on Computing*, 15(4) (2003) 347-368.

[3] Bräysy, O., Gendreau, M., "Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms", *Transportation Science*, 39(1) (2005) 104-118.

[4] Christofides, N., Mingozzi, A., Toth, P., "The vehicle routing problem", in: N. Christofides, A. Mingozzi, P. Toth, C. Sandi (eds.) *Combinatorial Optimization*, Wiley Chichester New York, (1979) 313-338.

[5] Hansen, P., Mladenović, N., "Variable Neighborhood Search", in: P. M. Pardalos, M. G. C. Resende (eds.) *Handbook of Applied Optimization*, Oxford University Press New York, (2000) 221-234.

[6] Hansen, P., Mladenović, N., "Variable Neighborhood Search: Principles and applications", *European Journal of Operational Research*, 130(3) (2001) 449-467.
[7] Hansen, P., Mladenović, N., Moreno Pérez, J. A., "Variable neighborhood search: methods and applications", *Annals of Operations Research*, 175(3) (2010) 367-407.
[8] Hemmelmayr, V. C., Doerner, K. F., Hartl, R. F., "A variable neighborhood search heuristic for periodic routing problems", *Computers & Operations Research*, 39(16) (2012) 3215-3228.
[9] Hosny, M. I., Mumford, C. L., "Solving the One-Commodity Pickup and Delivery Problem Using an Adaptive Hybrid VNS/SA Approach", in: R. Schaefer, C. Cotta, J. Kolodziej, G. Rudolph (eds.) *Parallel Problem Solving from Nature, PPSN XI, Lecture Notes in Computer Science*, Springer Berlin Heidelberg, vol. 6239 (2010) 189-198.
[10] Kritzinger, S., Tricoire, F., Doerner, K. F., Hartl, R. F., "Adaptive search techniques for problems in vehicle routing, Part I: A survey", to appear in *Yugoslav Journal of Operations Research* 25(1) (2015) 3-31.
[11] Kritzinger, S., Tricoire, F., Doerner, K. F., Hartl, R. F., Stützle, T., "A Unified Framework for Routing Problems with a Fixed Fleet Size", *Tech. Rep. JKU-PLM-2012-006*, Johannes Kepler University Linz, Austria, (2012).
[12] Lourenço, H. R., Martin, O. C., Stützle, T., "Iterated Local Search: Framework and Applications", in: M. Gendreau, J.-Y. Potvin (eds.) *Handbook of Metaheuristics, Second Edition*, International Series in Operations Research & Management Science, Springer Science+Business Media LLC, vol. 146 (2010) 363-397.
[13] Mladenović, N., Hansen, P., "Variable Neighborhood Search", *Computers & Operations Research*, 24(11) (1997) 1097-1100.
[14] Parragh, S. N., Doerner, K. F., Hartl, R. F., "Variable neighborhood search for the dial-a-ride problem", *Computers & Operations Research*, 37(6) (2010) 1129-1138.
[15] Pillac, V., Guéret, C., Medaglia, A. L., "An event-driven optimization framework for dynamic vehicle routing", *Decision Support Systems*, 54(1) (2012) 414-423.
[16] Polacek, M., Hartl, R. F., Doerner, K. F., Reimann, M., "A Variable Neighborhood Search for the Multi Depot Vehicle Routing Problem with Time Windows", *Journal of Heuristics*, 10(6) (2004) 613-627.
[17] Repoussis, P. P., Tarantilis, C. D., Ioannou, G., "The open vehicle routing problem with time windows", *Journal of the Operational Research Society*, 58(3) (2007) 355-367.
[18] Schrage, L., "Formulation and structure of more complex/realistic routing and scheduling problems", *Networks*, 11(2) (1981) 229-232.
[19] Shaw, P., "Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems", in: M. Maher, J. F. Puget (eds.) *Principles and Practice of Constraint Programming - CP98, Lecture Notes in Computer Science*, Springer Berlin Heidelberg, vol. 1520 (1998) 417-431.
[20] Solomon, M., "Algorithms for the vehicle routing and scheduling problems with time constraints", *Operations Research*, 45(2) (1987) 254-265.
[21] Taillard, É. D., Badeau, P., Gendreau, M., Potvin, J.-Y., "A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows", *Transportation Science*, 31(2) (1997) 170-186.
[22] Tricoire, F., Romauch, M., Doerner, K. F., Hartl, R. F., "Heuristics for the multi-period orienteering problem with multiple time windows", *Computers & Operations Research*, 37(2) (2009) 351-367.