DEPARTMENT OF ENGINEERING MANAGEMENT

Meta-analysis of metaheuristics: Quantifying the effect of adaptiveness in adaptive large neighborhood search

Renata Turkeš, Kenneth Sörensen, Lars Magnus Hvattum, Eva Barrena, Hayet Chentli, Leandro Coelho, Iman Dayarian, Axel Grimault, Anders Gullhav, Çağatay Iris, Merve Keskin, Alexander Kiefer, Richard Lusby, Geraldo Mauri, Marcela Monroy-Licht, Sophie Parragh, Juan-Pablo Riquelme-Rodríguez, Alberto Santini, Vinicius Gandra Martins Santos & Charles Thomas

UNIVERSITY OF ANTWERP
Faculty of Business and Economics
City Campus
Prinsstraat 13, B.226
B-2000 Antwerp
Tel. +32 (0)3 265 40 32
Fax +32 (0)3 265 47 99
www.uantwerpen.be

FACULTY OF BUSINESS AND ECONOMICS

DEPARTMENT OF ENGINEERING MANAGEMENT

Meta-analysis of metaheuristics: Quantifying the effect of adaptiveness in adaptive large neighborhood search

Renata Turkeš, Kenneth Sörensen, Lars Magnus Hvattum, Eva Barrena, Hayet Chentli, Leandro Coelho, Iman Dayarian, Axel Grimault, Anders Gullhav, Çağatay Iris, Merve Keskin, Alexander Kiefer, Richard Lusby, Geraldo Mauri, Marcela Monroy-Licht, Sophie Parragh, Juan-Pablo Riquelme-Rodríguez, Alberto Santini, Vinicius Gandra Martins Santos & Charles Thomas

RESEARCH PAPER 2019-002
OCTOBER 2019

University of Antwerp, City Campus, Prinsstraat 13, B-2000 Antwerp, Belgium
Research Administration – room B.226
phone: (32) 3 265 40 32
fax: (32) 3 265 47 99
e-mail: joeri.nys@uantwerpen.be

D/2019/1169/002

Meta-analysis of Metaheuristics:
Quantifying the Effect of Adaptiveness in Adaptive Large Neighborhood Search

Renata Turkeš Kenneth Sörensen Lars Magnus Hvattum
Eva Barrena Hayet Chentli Leandro Coelho
Iman Dayarian Axel Grimault Anders Gullhav
Çağatay Iris Merve Keskin Alexander Kiefer
Richard Lusby Geraldo Mauri Marcela Monroy-Licht
Sophie Parragh Juan-Pablo Riquelme-Rodríguez
Alberto Santini Vinicius Gandra Martins Santos
Charles Thomas

Research on metaheuristics has focused almost exclusively on (novel) algorithmic development and on competitive testing, both of which have been frequently argued to yield very little generalizable knowledge. The main goal of this paper is to promote meta-analysis — a systematic statistical examination that combines the results of several independent studies —as a more suitable way to obtain problem- and implementation-independent insights on metaheuristics. Meta-analysis is widely used in several scientific domains, most notably the medical sciences (e.g., to establish the efficacy of a certain treatment). To the best of our knowledge this is the first meta-analysis in the field of metaheuristics.

To illustrate the approach, we carry out a meta-analysis to gain insights into the importance of the adaptive layer in adaptive large neighborhood search (ALNS). Although ALNS has been widely used to solve a broad range of problems, it has not yet been established whether or not adaptiveness actually contributes to the performance of an ALNS algorithm.

A total of 134 studies were identified through Google Scholar or personal e-mail correspondence with researchers in the domain, 63 of which fit our eligibility criteria. After sending requests for data to the authors of the eligible studies, we obtained results for 25 different implementations of ALNS, which were analysed using a random-effects model.

On average, the addition of an adaptive layer in an ALNS algorithm improves the objective function value by 0.14% (95% confidence interval 0.07 to 0.22%). Although the adaptive layer can (and in a limited number of studies does) have an added value, it also adds considerable complexity and can therefore only be recommended in some very specific situations. These findings underline the importance of evaluating the contribution of metaheuristic components, and of knowledge over competitive testing.

1 Motivation for meta-analysis

Traditionally, research in the domain of metaheuristics has put great emphasis on competitive testing (the so-called “horse race" [55]), the main goal of which is the development of an algorithm that performs as well as possible (and preferably better than other algorithms) on a set of benchmark problem instances. However, as has been frequently argued, this type of studies generates very little understanding, since we only learn which algorithm performs better, but not *why*. For example, if a study shows that a variable neighborhood search with swap, insert and 2-opt moves, reactive tabu list and ejection chain perturbation yields solutions that are, on average, closer to the best known solution for a set of vehicle routing problem instances, than an evolutionary algorithm with hybrid intelligent crossover, roulette wheel selection, and mutation using Or-opt, we are most often left without any clarification which of the heuristic components is crucial for this good performance.

What is more, we cannot even be confident about which of the two given heuristic frameworks performs better for the given problem, as the comparison results might be significantly influenced by, e.g., the coding skills of the authors (a use of clever data structures and memory might significantly reduce the computation time), or some other algorithmic components that might not have even been mentioned in the articles. For the aforementioned example, for instance, we are not sure if the horse race winner would be the same if the heuristics were reimplemented by another author, or if variable neighborhood search with the same neighborhoods but with a different acceptance criterion would still outperform the evolutionary algorithm.

A result of this almost exclusive focus on competition is that the metaheuristics literature, even after several decades, has not been able to answer simple questions such as “Does a stochastic acceptance criterion outperform a deterministic one in local search algorithms?”, “Is a tabu list of variable size better than a fixed one?”, or — the question we will answer in this paper — “Does the adaptive layer of an adaptive large neigbhorhood search (ALNS) algorithm improve its performance?”.

In order to learn something about a particular component, it needs to be studied in isolation. For instance, a comparison of our imaginary variable neighborhood search with and without the ejection chain perturbation could be a first step in learning if the ejection chain perturbation improves the algorithmic performance for the problem under study.

In this way, the focus is shifted from the horse race to gaining knowledge, but the literature is sparse with respect to such studies. One example is [108], which investigates the acceptance criteria for adaptive large neighborhood search, by comparing a range of alternatives. However, the comparisons are carried out on set of instances from only three problems (capacitated vehicle routing problem, capacitated minimum spanning tree problem, quadratic assignment problem), and the authors themselves acknowledge that the conclusions drawn from the experiments will not necessarily apply to all other implementations, or other unrelated problems; they also acknowledge that the relative merit of the acceptance criteria may also change based on the number of iterations that can be run. Some knowledge is therefore generated, but limited to one or a few problems, the particular implementation, and the conclusions could even be misleading if they miss important interactions with some other factors (e.g., stopping criterion, or local search move). Obviously, the findings would be more trustworthy if they were independently confirmed in a number of articles by different authors, and on a variety of problems.

Meta-analysis of the literature is one way to obtain such insights related to a clearly formulated question, that do not depend on a particular problem, or on the particular implementation. Meta-analysis refers to a systematic review of the literature, combined with the use of statistical techniques to integrate and summarize the results of several independent studies [84]. The term "meta-analysis" was coined in 1976 by the statistician Gene V. Glass, who stated "my major interest currently is in what we have come to call…the meta-analysis of research. The term is a bit grand, but it is precise and apt…Meta-analysis refers to the analysis of analyses" [32]. By combining information from all relevant studies, meta-analyses can provide more precise insights than those derived from the individual studies included within a review.

The historical roots of meta-analysis can be traced back to 17th century studies of astronomy [99], but it gained the greatest popularity in medicine. The increasingly overwhelming amount of information is making it very difficult, if not impossible to stay current [128], and therefore clinicians read meta-analyses to keep up to date with their field and they are often used as a starting point for developing clinical practice guidelines [84]. A paper by statistician Karl Pearson published in 1904 [118], which collated data from several studies of typhoid inoculation, is seen as the first time a meta-analytic approach was used to aggregate the outcomes of multiple clinical studies. Since then, many meta-analyses have been carried out, e.g., on the effect of antibiotics on antimicrobial resistance [21], chemotherapy in adult high-grade glioma [38], accuracy of neutrophil helatinase-associated lipocalin in diagnosis and prognosis in acute kidney injury [43], ischemic stroke risk with oral contraceptives [31]. Meta-analyses are very helpful as they synthesize research, and therefore present a high level view on the evidence. Moreover, they are less subject to bias, and can help to identify patterns among study results as well as potential reasons for discrepant results.

The aim of this paper is to promote meta-analysis as a methodology that can help us answer general questions related to metaheuristics. Despite its popularity in medicine, to the best of our knowledge meta-analysis has not been applied in the domain of metaheuristics (or operations research in general, for that matter). We have only identified two systematic reviews of metaheuristic techniques for generating software tests [3], and of genetic programming for software engineering predictive modeling [2], but no meta-analysis was performed. Meta-analysis has great potential in generating knowledge which is problem- and implementation-independent, and less influenced by confounding factors (many of which we would not necessarily even be aware of). For example, the comparison of two algorithms can be severely influenced by the given computation time, but in case of a meta-analysis we consider multiple studies with different stopping criteria, so that the results become more reliable. To illustrate the approach, we carry out meta-analysis of the adaptive layer in the adaptive large neighborhood search (ALNS).

The ALNS metaheuristic was introduced by Røpke and Pisinger [104], and is based on the principle of large neighborhood search (LNS) [115], where the problem is iteratively relaxed and re-optimized, i.e., a part of the solution is "destroyed" and then "repaired". LNS is similar to the ruin and recreate paradigm presented in [114], or the ripup and reroute paradigm applied in [24]. Røpke and Pisinger [104] proposed to alternate between *a number of different* destroy and repair heuristics, arguing that the overall robustness of the search can be increased. However, since it is unknown in advance which heuristics are best suited for the instance under consideration, and since the performance of a heuristic might vary during the search (e.g., some methods might be well suited in the first iterations of the search while others lead to better results in later iterations [52]), the ALNS metaheuristic selects the heuristics in a probabilistic and *adaptive* fashion, according to their past performance [10].

Even though it seems reasonable to assume that the adaptive layer improves the performance of an ALNS heuristic by selecting destroy and repair heuristics that perform well for the problem instance, this behavior has not yet been established as a fact, let alone quantified. The goal of the meta-analysis in this paper is therefore to answer the following research question:

> How much does the ALNS adaptive layer help to improve heuristic performance?

The remainder of the paper is organized as follows. In Section 2, we describe in detail the ALNS, with a focus on the adaptive layer and our research question. Section 3 provides an overview of the literature related to the topic of interest. We then proceed to describe how we selected the studies included in the meta-analysis, and identify their main properties in Section 4. Before carrying out a statistical analysis to summarize the results of the studies, we first examine some sources of bias, both in individual and across studies in Section 5. The meta-analysis is described in Section 6, and then discussed in detail in Section 7, in order to identify the main factors that influence the importance of the ALNS adaptive layer. A summary of findings, limitations and suggestions for future research are given in Section 8.

2 A in ALNS: the adaptive layer

Adaptive large neighborhood search (ALNS) has been applied to a broad range of routing [6–8, 13, 18, 20, 25, 27, 29, 33, 39, 41, 48, 51, 56–58, 66, 69–71, 74, 90–93, 97, 101, 105, 106, 113, 125] and scheduling problems [4, 11, 15, 22, 46, 87, 97, 98, 100–102, 104, 119, 129], but also to the location problem [17], lot sizing [89], assignment problem [53, 133], or a combination of (the aforementioned) problems: inventory routing [5, 44, 86, 116, 117, 132], production routing [1], location routing [62], or routing and scheduling [10].

In ALNS, a solution is iteratively destroyed and repaired through the application of *several* heuristics $h \in \mathcal{H}$. In order to select the heuristic to use, a weight is assigned to each destroy heuristic $h \in \mathcal{D}$ and each repair heuristic $h \in \mathcal{R}$. First, weights are set to some initial values, which are usually equal. An *adaptive* weight adjustment procedure is proposed in order to update these weights based on the performance of each heuristic. The entire search is divided into a number of segments. The score π_h of each heuristic is set to zero at the start of each segment, and is increased by:

$$\delta = \begin{cases} \delta_1, & \text{if the solution is a new global best} \\ \delta_2, & \text{if the solution is better than the current, and not accepted before} \\ \delta_3, & \text{if the solution is worse than current, accepted, and not accepted before.} \end{cases} \tag{1}$$

In Røpke and Pisinger [104] where ALNS was first introduced, the authors suggested employing a simulated annealing Metropolis acceptance criterion, which accepts worse solution with a certain probability. This criterion has indeed been widely used for ALNS, but several other acceptance criteria have also been applied in the literature. Note from Equation (1) that only previously unvisited solutions are rewarded, in order to encourage heuristics that are able to explore new parts of the solution space. This means that such an implementation of ALNS requires keeping track of visited solutions, and checking if a solution has been visited before.

At the end of each segment, new weights are calculated using the aforementioned recorded scores. The weight w_h^{s+1} of the heuristic h in segment $s + 1$ is calculated as follows:

$$w_h^{s+1} = (1 - r)w_h^s + r\frac{\pi_h}{\theta_h}, \tag{2}$$

where θ_h is the number of times heuristic h was used during the last segment (if $\theta_h = 0$, the weight w_h remains unchanged), and r is the reaction factor. The reaction factor r controls how quickly the weight adjustment procedure reacts to changes in the effectiveness of the heuristic: if $r = 0$, the weights remain unchanged, and if $r = 1$ the weights are determined by the performance in the last segment [104]. The number of segments, the reaction factor r and the scores δ_1, δ_2 and δ_3 are (meta)parameters of the adaptive layer of ALNS, which need to be tuned. In each iteration, two roulette wheel selection principles are applied to calculate the probabilities for choosing respectively a destroy and repair heuristic:

$$\frac{w_h^s}{\sum_{l\in\mathcal{D}} w_l^s}, \quad \frac{w_h^s}{\sum_{l\in\mathcal{R}} w_l^s}.$$

Evidently, the adaptive layer adds considerable complexity to the heuristic, and requires considerable additional implementation and computational effort. The question of whether the adaptive layer actually improves the performance of an ALNS heuristic is therefore of more than just theoretical interest. Should we be able to demonstrate that the adaptive layer adds little or no performance, then keeping track of heuristic weights, visited solutions, and tuning the parameters δ_1, δ_2, δ_3, r, and the number of segments, could all be avoided.

In this research, we are interested in comparing ALNS with the tuned r and the non-adaptive version with $r = 0$, with fixed heuristic weights. The non-adaptive version of ALNS is sometimes referred to as LNS, but because LNS is more often interpreted as a single pair of a destroy and repair heuristic, and in order to avoid any confusion, we refer to it as (¬A)LNS.

To compare the adaptive and non-adaptive version of an ALNS heuristic, we calculate the *average improvement of the objective function caused by the adaptive layer A*. Let us assume that we are considering a maximization problem with the objective function f, and let $\{I_1, I_2, \dots, I_N\}$ denote the set of available problem instances. Let $x_r^*(I)$ be the best solution for instance I found by the ALNS (with some non-zero value of the reaction factor r), and let $x_0^*(I)$ be the best solution found by the non-adaptive (¬A)LNS with $r = 0$ for the same problem instance. Since ALNS is not a deterministic algorithm, several runs with different outcomes can be performed on a single instance. For each instance, we therefore calculate the average objective function values across a number of runs, $\overline{f}(x_r^*(I))$ and $\overline{f}(x_0^*(I))$. For each instance, we then calculate the percentage improvement in the average objective function value yielded with the adaptive layer:

$$\frac{\overline{f}(x_r^*(I)) - \overline{f}(x_0^*(I))}{\overline{f}(x_0^*(I))}.$$

We are interested in the average improvement with the adaptive layer, across the set of available instances:

$$A = \frac{1}{N} \sum_{I \in \{I_1, I_2, \dots, I_N\}} \frac{\overline{f}(x_r^*(I)) - \overline{f}(x_0^*(I))}{\overline{f}(x_0^*(I))}. \tag{3}$$

If we are considering a minimization problem, the average improvement in the objective function is calculated as

$$A = -\frac{1}{N} \sum_{I \in \{I_1, I_2, \dots, I_N\}} \frac{\overline{f}(x_r^*(I)) - \overline{f}(x_0^*(I))}{\overline{f}(x_0^*(I))}. \tag{4}$$

Our research question now translates to:

> What is the percentage improvement A in the objective function obtained by ALNS compared to the non-adaptive variant (¬A)LNS?

Each study included in our meta-analysis is summarized with A. In meta-analysis for medical research, each study is described with its "effect size" of therapy: the mean difference of the outcome variable between treated and untreated subjects. In our case, the subjects (i.e., patients) correspond to problem instances, the treatment corresponds to the ALNS, no treatment corresponds to the non-adaptive alternative, the outcome variable is the objective function value, and the effect size is the average improvement A. The difference is that in medicine there are two groups of patients, those who receive the treatment and those who do not (control group), whereas we can evaluate both the non-adaptive and adaptive algorithm on each instance.

We note that $A > 0$ means that the ALNS on average improves over the non-adaptive alternative (¬A)LNS, which is not necessarily equivalent to saying that ALNS has a smaller average gap from the best known solution than (¬A)LNS (Table 1). There is a number of ways one could define what it means for one algorithm to be better than another one, but we chose to investigate if $A > 0$, as we believe this to be the most straightforward translation of our research question.

Table 1: Algorithms I and II have respectively 4.54% and 5% average gaps from the best known solution. On the other hand, algorithm II finds a solution which is 10% better for instance I_1, but 10% worse for instance I_2, so that the average improvement of algorithm II *relative* to algorithm I is 0%. If algorithms I and II correspond respectively to the (¬A)LNS and ALNS, $A = 0\%$ measures the average improvement of ALNS *over* the non-adaptive variant, i.e., it represents the added value of the adaptive layer.

maximization problem	algorithm I	algorithm II
instance I_1	10	11
instance I_2	100	90

Before proceeding to the next section, we also note that the implementations of ALNS in the literature vary with respect to a number of algorithmic components. Besides the differences in the problem-dependent destroy and repair heuristics, there are many differences with respect to other problem-independent components, e.g., stopping or acceptance criteria. A lot of articles do not employ the above weight adjustment formula (2), or assign scores to the heuristics in a different way. For example, Røpke and Pisinger proposed to update the scores of a destroy and repair heuristic by the same amount using Equation (1), since we cannot tell whether it was the destroy or the repair heuristic that was the reason for the “success". However, rather than evaluating the heuristics independently (with two separate roulette wheel mechanisms), one might rather weigh each heuristic pair separately. Another example of a difference between ALNS implementations is the presence or absence of noise. Since the repair heuristics are commonly greedy with respect to some criteria (e.g., lowest increase in the objective function, greatest regret), they are quite myopic, which prompted Røpke and Pisinger to suggest randomizing the insertion heuristics so that they do not always make the move that seems best locally. This can be achieved by adding a noise term to the objective function. Noise is considered in some, but not all of the studies which consider ALNS. Some of destroy heuristics are also greedy (e.g., worst components are removed), so that noise can similarly be incorporated in the destroy heuristics. We describe each of these important differences in greater detail in Section 4 (and further discuss their impact on the importance of adaptiveness in Section 7).

3 Justification for review

Insights regarding the importance of the adaptive layer in ALNS in the literature are few and far between. As mentioned, the research goal underlying a large majority of the papers in the domain of metaheuristics (and therefore also ALNS papers) is to introduce an algorithm which outperforms all others on a set of benchmark instances. The experimental results found in such papers therefore invariably focus on the comparison either with other algorithms introduced in the literature, or with an exact solver, a lower or upper bound, etc. Only a limited number of articles compare ALNS with the non-adaptive version (¬A)LNS, in order to assess the added value of the adaptive layer.

In several papers, the adaptive layer was found to be unhelpful. For example, it was shown not to have any added value in [127], and the authors therefore decided to refrain from using it, and rather choose the heuristics randomly according to a uniform distribution. Similarly, the adaptive layer was removed in [34, 68, 94, 123], since extensive tests showed that no (significant) advantage could be identified when employing the adaptive mechanism. A complete removal of the adaptive layer was shown to exhibit only a very slight deterioration of results in [28]: with a careful offline heuristic selection, the adaptive mechanism did not need to be implemented to further guide the heuristic selection online. The adaptive layer was shown to have only a negligible impact on the solutions in [88]. In [63], the authors demonstrated that (¬A)LNS with equal weights performs on average better than ALNS with the weights independently adjusted, but (sometimes) worse than the ALNS with pairwise weights.

Several papers found the adaptive layer to behave in perhaps unexpected ways. The frequency of using different destroy and repair heuristics did not vary significantly across heuristics, but also not across sets of instances in [25, 62, 91]. The average selection probabilities of the heuristics did not change much for different values of the reaction factor in [46, 51, 64]. This means that ALNS was shown to be robust with respect to changes in the meta-parameter r.

In a few articles the adaptive layer was shown to be beneficial. For example, ALNS was shown to be considerably better than the non-adaptive versions in [15], with the weights fixed in three different ways. ALNS was also demonstrated to outperform (¬A)LNS (for majority of instances) in [37, 79]. ALNS was somewhat better than equiprobable (¬A)LNS in [60], although (¬A)LNS with predefined probabilities (tuned by using ALNS), as well as (¬A)LNS with uniform probabilities and with four repair heuristics excluded, seemed to perform better than ALNS, with or without the four aforementioned repair heuristics.

There are many more papers which do not explicitly evaluate the importance of the adaptive layer, but which do investigate the performance of different destroy and repair heuristics. The final probabilities of choosing different heuristics, averaged across instances, were shown to be (more or less) different in [14, 23, 27, 74, 76, 78]. According to the authors, significant variations between the min and max final selection probabilities in the case of some heuristics in [23] suggest the importance of the adaptation layer. In [1, 30, 46, 64, 109, 110], we can also see that different heuristics are used with somewhat different frequencies across different (groups of) instances. Intuitively, this is an indication that there is some benefit to the adaptive layer, but we note immediately here that this is not necessarily the case: although significant differences between heuristic selection probabilities were shown in [109, 110], we will see later (Table 4) that ALNS does not outperform (¬A)LNS for the algorithms introduced in these articles. In [42], the selection probabilities can be seen to fluctuate throughout different phases of the algorithm, for a single instance, which the authors interpret as a benefit of using the adaptive strategy. ALNS is implemented in [124] to solve 10 different combinatorial optimization problems, and not surprisingly, the difference between the weights of the heuristics vary greatly, since only a few heuristics are particularly useful for each of the problems. Moreover, the authors even showed that the LNS with the best pair of one destroy and one repair heuristic outperforms ALNS.

Furthermore, to tune the reaction factor r, a number of different values are considered in [111, 126], with significant differences in the performance of ALNS. In [72], the average frequencies of the heuristics change considerably for an instance, for different values of the reaction factor.

To summarize, there are some articles that explicitly show that the adaptive layer has no added value, whereas others demonstrate that adaptiveness helps to improve the algorithmic performance. In addition, many articles look into the selection probabilities of different heuristics, which give some indications (although not evidence) that the adaptiveness might be beneficial in some cases. The discrepant results (without any insights about the reasons behind the sources of disagreement) confirm the relevance of our research question.

4 Identification, selection and main properties of studies

A Google Scholar search of “adaptive large neighborhood search" (without the quotes) returned 332 000 records, which is too many to process. In addition, most of the time ALNS is only mentioned in the literature review of the articles. We therefore limited the search only to the records that contain “adaptive large neighborhood search" in the title, identifying 129 records in total. The search was performed on April 19, 2019. Out of those, 34 records were excluded, as they were either citations only, duplicates, could not be found, were not in English, or did not actually contain ALNS in the title. The remaining 95 records were accessed to assess their eligibility.

We restricted our selection to the articles that:

- describe the weight adjustment mechanism used in sufficient detail,
- employ a weight adjustment formula which includes a parameter, whose value could be set to a certain value so that the adaptive layer is switched off, and
- employ a roulette wheel mechanism to choose between heuristics.

The following 25 articles were excluded because the weight adjustment mechanism was not described in detail: [9, 10, 17, 20, 29, 36, 40, 41, 44, 45, 47, 57, 61, 65, 77, 86, 90, 106, 112, 116, 117, 119, 122, 130, 132]. An additional 5 articles [1, 22, 92, 98, 133] were excluded because the adaptive layer cannot be removed by setting a parameter to a certain value, and [58] was excluded as a roulette wheel principle is not employed (rather, a heuristic with the highest weight is chosen). [82] was excluded because there is no ALNS introduced in the paper (rather, solutions from a population are chosen in an adaptive fashion).

Out of the 63 eligible studies, only [79] included the comparisons between ALNS and the non-adaptive (¬A)LNS. For the remaining 62, the main authors (Turkeš, Sörensen, Hvattum) of this review e-mailed the request for data. We asked for the objective function value per instance (for a number of runs) for ALNS with the value of the reaction factor r which was used in the paper, and with $r = 0$ (so that the weights remain unchanged, see Section 2), whereas the remaining parameters should have the same value as mentioned in the paper, for both the adaptive and non-adaptive variant. It is precisely for this reason (i.e., having to request data from earlier work), that we decided to consider the inclusion criteria above, as they allowed us to describe in a straightforward manner the information we were interested in. As obtaining the requested results is typically non-trivial and may require making changes to existing code and rerunning experiments, the approached researchers were invited as co-authors of this paper.

A few contacted authors suggested including results also for additional articles (not identified with Google Scholar, or with no ALNS in the title: [59, 107, 108, 110, 123]), which we decided to consider in order to have as many potential studies as possible. These studies therefore did not need to satisfy our eligibility criteria, since the researchers were already clear on precisely what type of information was needed. The flowchart in Figure 1 describes the process in greater detail, and is created according to the PRISMA guidelines (Preferred Reporting Items for Systematic Reviews and Meta-Analyses, established by an international group of experts) [84].

The included studies describe ALNS implementations for a variety of problems, and differ further on a number of study characteristics, that might have an influence on the importance of the adaptive layer:

- Article - Although ALNS might be used to study the same problem, and even if the same values of other parameters were used, the results might depend on the particular implementation, or some other confounding factors that have not been identified.

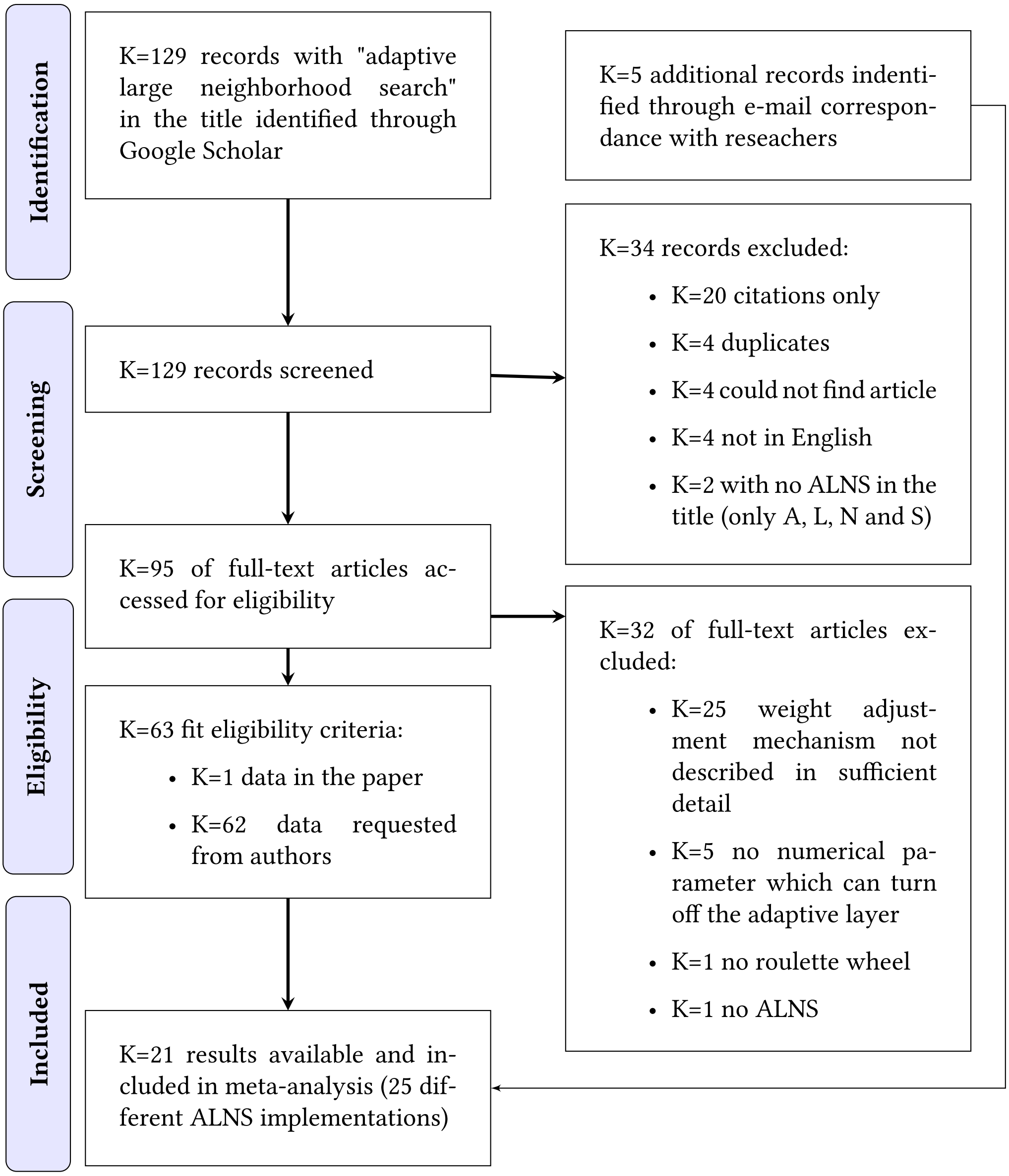

Identification
Screening
Eligibility
Included
K=129 records with "adaptive large neighborhood search" in the title identified through Google Scholar
K=5 additional records indentified through e-mail correspondance with reseachers
K=34 records excluded:
• K=20 citations only
• K=4 duplicates
• K=4 could not find article
• K=4 not in English
• K=2 with no ALNS in the title (only A, L, N and S)
K=129 records screened
K=95 of full-text articles accessed for eligibility
K=32 of full-text articles excluded:
• K=25 weight adjustment mechanism not described in sufficient detail
• K=5 no numerical parameter which can turn off the adaptive layer
• K=1 no roulette wheel
• K=1 no ALNS
K=63 fit eligibility criteria:
• K=1 data in the paper
• K=62 data requested from authors
K=21 results available and included in meta-analysis (25 different ALNS implementations)

Figure 1: PRISMA flowchart describes the process of identification and selection of studies to be included in the meta-analysis.

- Problem - The importance of the adaptive layer might depend on the particular (type of the) problem(s) under study.

- Objective function - The particular formulation of the objective function can have an influence on the importance of adaptiveness. For example, in some of the studies [23, 123] included in the analysis, the objective function includes some fixed costs, which can be large. If the objective function corresponds to, e.g., total routing costs, and two algorithms return solutions with with the cost of 100 and 90, the latter algorithm improves over the former by 10%. If, however, the objective function corresponds to the sum of a fixed vehicle cost of 1000 and the total routing costs, the objective function of the two solutions would amount to 1100 and 1090, so that the latter algorithm outperforms the former by only 0.9%. It is also important to know if the goal is to maximize or minimize the objective function, in order to know if formula (3) or (4) should be employed to assess the added value of adaptiveness.

- Number of destroy and repair heuristics - The greater the number of heuristics, the greater we can expect the importance of the adaptive layer.
- Weight adjustment formula - There is a great difference in the weight adjustment formula used for ALNS in the literature. The interpretation of scores δ does not always correspond to the description given in (1), e.g., one does not necessarily keep track of the visited solutions and rewards therefore any accepted solution, or differentiates between only two, or even four or more different scores. In some articles, the computation time of the heuristics is incorporated when recalculating the heuristic weights, e.g., in [1, 16, 42, 124, 129, 131]. In a few papers, there are no constant scores that are used to update the weights, but rather the score for a heuristic in a segment is calculated as $\frac{\Delta f}{\Delta t}$, where Δf corresponds to the difference in the objective function, and Δt is the computation time of the heuristic, e.g., in [124]. The latter is simpler since no score-related parameters need to be tuned, and it differentiates between heuristics which produce solutions which are almost as good as the current solution and the ones that produce solutions which are much better (or worse). The former, however, allows to not only reward improvements in the performance related to a better objective function value, but also to reward diversification, if it is shown by parameter tuning to be more important. In [102], the weight adjustment formula also incorporates a similarity index to reward very different solutions.
- Heuristic evaluation - The weights of destroy and repair heuristics can be updated independently or pairwise, and the choice of the approach could influence the importance of the adaptive layer.
- Initial weights, and scores - It is not only the reaction factor r, but also the ratio between the initial weights and the scores, which determines the importance of past performance.
- Reaction factor - The value of the reaction factor determines the importance of past performance.
- Number of segments - The number of times that the heuristic weights are updated plays a role in the adaptive layer.
- Noise - If noise is added to the objective function, the diversification (and hence, possibly the adaptive layer as well) might become less crucial.

- Acceptance criterion - We only mention the acceptance criterion used, but of course there could be even further differences that affect the importance of the adaptive layer (e.g., starting temperature and the cooling rate if the simulated annealing acceptance criterion is used).
- Stopping criterion - If a very large amount of computation time (or number of iterations) is available, we expect the differences to be less, since even in the non-adaptive version, the most successful heuristics would be chosen sufficiently many times. Reversely, the more limited the computation time, we expect greater importance of the adaptive layer. In addition, the choice of the stopping criterion can be an important source of bias, which we discuss in greater detail in the next section.

Table 2 gives an overview of the studies included in the meta-analysis, with respect to the aforementioned study characteristics.

5 Bias

Before carrying out the analysis, it is important to acknowledge the potential sources of bias, both in individual studies and across studies.

5.1 Bias in individual studies

The first source of bias is the stopping criterion: most of the articles that describe an ALNS adopt the number of iterations as the termination criterion proposed in the original ALNS paper [104]. Since the average computation times of different destroy and repair heuristics generally differ, and since the heuristic selection probabilities in ALNS and (¬A)LNS also vary, the computation time for the two variants of the algorithms might be significantly different for a given iteration limit. It is rather difficult to make any general claims about which of the variants benefits from such a stopping criterion, but it is to be expected that more sophisticated heuristics tend to produce better results at the cost of larger runtime, and that these heuristics would be selected more often in ALNS. This means that, on average, an iteration of the non-adaptive variant is expected to be faster. If the stopping criterion would be computation time, it would therefore be common for the non-adaptive variant to run a greater number of iterations, and the additional iterations could help to improve the quality of the best found solution. This bias is pronounced more strongly if the weight adjustment mechanism does not incorporate heuristic runtimes, which is the case for most of the studies included in our meta-analysis (Table 2).

Table 2: Studies included in the meta-analysis differ with respect to a number of characteristics.

Article	Problem	Objective	Number of destroy heuristics	Number of repair heuristics	Weight adjustment formula	Heuristic evaluation	Initial weights	Scores	Reaction factor	Number of segments	Noise	Acceptance criterion	Stopping criterion
[11]	Train timetabling problem with dynamic demand	Minimizing average passenger waiting time	3	3	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Independent	1	(10, 5, 2)	0.7	200 iterations in a segment	✗	Simulated annealing Metropolis criterion	Temperature of 0.01, with a cooling rate as a function of maximum 70 000 iterations
[19]	Profitable tour problem with simultaneous pick-up and delivery services	Maximizing difference between collected profit and routing cost	6 4	4 3	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Partially independent	1	(33, 9, 0)	0.2	9 000	✓	Simulated annealing Metropolis criterion	900 000 iterations
[23]	Multi-period vehicle routing problem	Minimizing the sum of fixed vehicle costs and routing costs	9	3	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Pairwise	1	(2, 1, 1, 1)	0.25	The length of a segment is adjusted dynamically.	✗	Simulated annealing Metropolis criterion	Temperature, minimum 25 000 iterations
[37]	Pick-up and delivery problem with resource synchronization	Minimizing the sum of travel costs, costs of service times and vehicle utilization costs	7	10	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Independent	1	(33, 20, 13)	0.1	100 iterations in a segment	✓	Simulated annealing Metropolis criterion	600 seconds
[42]	Multi-tier service deployment problem in clouds	Minimizing the total cost of placing replicas in the public cloud	6	3	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Independent	1	(42, 31, 22)	0.5	100 iterations in a segment	✗	Simulated annealing Metropolis criterion	Instance-specific number of iterations
[54]	Berth allocation and quay crane assignment problem	Minimizing overall costs	4	4	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Independent	1	(2, 5, 10)	0.8	270	✓	Simulated annealing Metropolis criterion	270 000 iterations

Studies included in the meta-analysis differ with respect to a number of characteristics (continued).

Article	Problem	Objective	Number of destroy heuristics	Number of repair heuristics	Weight adjustment formula	Heuristic evaluation	Initial weights	Scores	Reaction factor	Number of segments	Noise	Acceptance criterion	Stopping criterion
[59]	Electric vehicle routing problem with time windows	Minimizing number of vehicles (and next the total distance traveled)	16	7	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Independent	1	(25, 20, 21)	0.25	125 or 4, depending on the category of heuristics	✗	Simulated annealing Metropolis criterion	25 000 iterations
[60]	Curriculum-based course timetabling problem	Minimization of weighted sum of integer penalties	10	8	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$ $w_l^{s+1} = (1-r)w_l^s + r\frac{\pi_l}{\theta_l}(\frac{T_0}{T_l})$ $h \in \mathcal{D}, l \in \mathcal{R}$, T_l average computation time of heuristic l, T_0 average computation time of an arbitrarily chosen reference repair heuristic	Independent	1	(30, 15, 18)	0.16	100 iterations in a segment	✓	Simulated annealing Metropolis criterion	Instance-specific number of iterations
[63]	Service technician routing and scheduling problem	Minimizing sum of total routing and outsourcing costs	5	6	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Pairwise	1	(33, 9, 13)	0.1	250	✓	Simulated annealing Metropolis criterion	25 000 iterations
[75]	Dynamic patient admission scheduling problem	Minimizing sum of penalties for soft constraints violations	3	2	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Independent	1	(25, 15, 5)	0.1	100 iterations in a segment	✓	Simulated annealing Metropolis criterion	Temperature 0.5
[79]	Multi-depot multi-period vehicle routing problem with a heterogeneous fleet	Minimization of total delivery cost	4	1	$w_h^{s+1} = w_h^s - \delta$ $w_l^{s+1} = w_l^s + \frac{\delta}{\lvert\mathcal{D}\rvert}$ h failed to improve the current best solution, $l \in \mathcal{D}\backslash\{h\}$	Independent (= Pairwise)	1	0.01	0.5	1 iteration in a segment, hence at most 100 segments	✗	Hill climbing	100 iterations, or 10 iterations without improvement
[81]	Discrete and continuous berth allocation problem	Minimizing the sum of service times and penalties for deadline violations	4	3	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Independent	1	(50, 10, 7)	0.05	50 iterations in a segment	✓	Simulated annealing Metropolis criterion	120 seconds

Studies included in the meta-analysis differ with respect to a number of characteristics (continued).

Article	Problem	Objective	Number of destroy heuristics	Number of repair heuristics	Weight adjustment formula	Heuristic evaluation	Initial weights	Scores	Reaction factor	Number of segments	Noise	Acceptance criterion	Stopping criterion
[85]	Rural postman problem with time windows	Minimizing cost	3	2	$w_h^{s+1} = (1-r)\frac{\sum_{t=1}^{s-1} w_h^t}{s-1} + r\frac{\pi_h}{\theta_h}$	Independent	1	(15, 8, 2)	0.7	100, with heuristic weights re-initialized after every 10 segments	✗	Simulated annealing Metropolis criterion	25 000 iterations
[93]	Truck and trailer routing problem with time windows	Minimizing total routing cost	4	5	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Pairwise	1	(33, 9)	0.1	500	✓	Simulated annealing Metropolis criterion	50 000 iterations
[103]	Periodic capacitated arc routing problem with inventory constraints - *gfbj* instances	Minimizing the sum of penalty cost of humidity level difference, and routing cost	8	8	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	There is a single repair heuristic corresponding to each destroy heuristic.	1/8	(15, 10, 5)	0.8	100	-	Simulated annealing Metropolis criterion	25 000 iterations or 7 200 seconds
	Periodic capacitated arc routing problem with inventory constraints - *mine* instances							(25, 10, 5)	0.1				
[108]	Capacitated vehicle routing problem	Minimizing distance	3	1	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Independent	1	((10, 4, 2)	0.01	150 000	✗	Record-to-record travel	150 000 iterations
	Capacitated minimum spanning tree problem	Minimizing sum of edge costs	2	1									
	Quadratic assignment problem	Minimizing cost	1	3									
[107]	Orienteering problem	Maximize collected prize	3	4	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Independent	1	(15.38, 5.34, 3.04)	0.57	1 iteration in a segment	✗	Record-to-record travel	5-minutes timeout

Studies included in the meta-analysis differ with respect to a number of characteristics (continued).

Article	Problem	Objective	Number of destroy heuristics	Number of repair heuristics	Weight adjustment formula	Heuristic evaluation	Initial weights	Scores	Reaction factor	Number of segments	Noise	Acceptance criterion	Stopping criterion
[109]	Gate matrix layout problem	Minimizing number of tracks	5	4	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Independent	1	(15, 25, 5)	0.66	60 iterations in a segment, hence at most 13 segments	✓	Simulated annealing Metropolis criterion	Temperature 0.01, or objective function value equal to a trivial lower bound, or 100 consecutive iterations without improvement (but minimum 400 in total), or 800 iterations
[110]	Cutwidth minimization problem	Minimizing maximum number of edges crossing any consecutive pair of vertices	5	4	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Independent	1	(50, 15, 25)	0.85	200 iterations in a segment, hence at most 15 segments	✓	Simulated annealing Metropolis criterion	Temperature 0.01, or 3 000 iterations
[123]	Fleet size and mix dial-a-ride problem	Minimizing the sum of fixed vehicle costs and times and distance costs	5	4	$w_h^{s+1} = (1-r)w_h^s + r\frac{\pi_h}{\theta_h}$	Independent	1	(33, 20, 15)	0.1	100 iterations in a segment	✓	Record-to-record travel	Instance-dependent computation time (16, 40 or 100 minutes)
[124]	10 different combinatorial optimization problems	Minimizing different objectives	30	36	$w_h = (1-r)\frac{L}{T}T_h + rL_h$ L_h, T_h local and total efficiency of heuristic h L, T local and total efficiency of all heuristics	Independent	1	$\frac{\Delta f}{\Delta t}$	0.95	1 iteration in a segment	✗	Hill climbing	240 seconds

Regardless of the frequency of the heuristics in the adaptive and non-adaptive algorithm, the iteration limit as the termination criterion could bias the comparison in favor of ALNS in yet another way. Indeed, no information about the heuristic scores or weights needs to be updated in (¬A)LNS, but more importantly there is no need to memorize visited solutions, or check if a solution has been visited before, which is another reason why the non-adaptive algorithm runs faster. This is another reason why, if both were limited by the same amount of computation time, (¬A)LNS would be able to perform more iterations and possibly find a better solution within the additional time gained. In other words, the lack of the adaptivity could be compensated by the additional iterations gained, which is not achieved if the algorithm is stopped with a given number of iterations. Note, however, that the computational steps related to the adaptive layer are still carried out in the non-adaptive variant considered in our meta-analysis: we chose to ask the authors to run the comparison of ALNS with (¬A)LNS with $r = 0$, as this is more straightforward than asking them to remove the code related to the adaptive layer, but these two non-adaptive variants are not equivalent. The simplified non-adaptive variant (¬A)LNS with $r = 0$ carries out the algorithmic steps concerned with adaptiveness, although this has no influence on the selection of heuristics. It is precisely this simplification that is the second source of bias in our meta-analysis, and it rather occurs when the stopping criterion is the computation time. In this case, non-adaptive (¬A)LNS without A-code saves some time compared to ALNS due to the lack of adaptivity, but (¬A)LNS with $r = 0$ achieves no such gains.

The most fair comparison would therefore be limited by computation time, and would consider ALNS without the adaptiveness code as the appropriate (¬A)LNS. In Table 3, the aforementioned arguments are supported with the ALNS introduced in [60]. We are not particularly concerned with these sources of bias, but we find it important to note that in most cases they can make the A defined in (3) an exaggerated estimate of the importance of the adaptive layer.

We also note that we are comparing ALNS to the equiprobable (¬A)LNS, whereas there might be much more successful non-adaptive variants. If the selection probabilities of the included heuristics would be tuned (offline), it is to be expected that this non-adaptive (¬A)LNS would have a better performance than the (¬A)LNS where the selection probabilities are equal.

On the other hand, for some problems, there might be a bias in favor of the non-adaptive alternative. Indeed, if many simple instances are considered, for which both ALNS and the non-adaptive (¬)ALNS can find the optimal solution, the adaptive layer cannot improve the performance. If the adaptive layer has an added value, it is to be expected that it will be greater for large instances. A similar issue arises if the problem is very constrained so that only a limited number of feasible solutions exist. For these instances, the non-adaptive variant is as successful as ALNS. However, we decided to also include in the results the percentage of instances for which ALNS is better or worse than (and hence also equal to) (¬)ALNS. Together with standard deviation, the percentage of instances for which ALNS and the non-adaptive variant return the same solution quality, gives an indication about how much the average importance of adaptiveness in a study could at most be skewed

Table 3: Two sources of bias in individual studies are illustrated with the results for the ALNS introduced in [60] for the curriculum-based course timetabling (minimization) problem, averaged across 21 instances and 5 runs. The first is the number of iterations as the stopping criterion: if the stopping criterion would be time, a smaller difference between ALNS and the non-adaptive alternative is expected. The additional computation load of the adaptive layer is the second source of bias: the (¬A)LNS with $r = 0$ is a conservative estimate of the non-adaptive version of (¬A)LNS (with the code related to the adaptive layer removed), since savings in computation time could improve the performance.

Stopping criterion	Algorithm	Number of iterations	Computation time (s)	Objective function value
Number of iterations	ALNS	425 714.29	160.26	75.42
	(¬A)LNS with r=0	425 714.29	136.20	79.50
	(¬A)LNS without A-code	425 714.29	126.92	79.29
Computation time	ALNS	478 846.76	192.00	75.12
	(¬A)LNS with r=0	598 570.31	192.00	78.09
	(¬A)LNS without A-code	686 910.50	192.00	78.21

(made smaller) because of the simple instances where ALNS could not ever outperform the non-adaptive alternative.

5.2 Bias across studies

Next to the bias in individual studies, there are also three potential sources of bias across studies: publication, search, and selection bias [128]. In the remainder of the section, we define each of these types of bias, and explain how they apply to the meta-analysis carried out in this paper.

The first type of bias, the publication bias, is caused by the fact that positive studies are more likely to be published. Medical research is often sponsored by pharmaceutical companies, which obviously prefer results in favor of their developed treatments. For example, trials of single alkylating agents versus multiple agent cytotoxic chemotherapy in the treatment of ovarian cancer have been analyzed [83]. Published trials yielded significant results in favor of the multiple-agent therapy, but that finding was not supported when the results of all trials — both those published and those registered but not published — were analyzed. Another study suggested that intervention effects reported in journals were 33% greater than those reported in doctoral dissertations [83]. Similarly, meta-analyses limited to published trials, compared with those that included both published and grey literature, overestimated the treatment effect by an average of 12% [83]. A similar (if not worse) bias exists in metaheuristic research: the focus lies in designing an algorithm that outperforms another, or at least on demonstrating the benefits of certain components, whereas negative results are difficult to publish. Such selective publishing of studies with statistically significant or directionally positive results can lead to inflated estimates of efficacy in the meta-analyses. This bias occurs in the identification phase, and should be minimized with an effort to identify unpublished studies. We minimized the publication bias by not restricting the review to studies published in journals: we also include conference proceedings, bachelor, master and PhD theses, and working papers (some of which have in the meantime been published in a journal). We also do not make any restrictions with respect to the number of citations. For practical purposes we did, however, consider only studies in English. Language-restricted meta-analyses, as compared with language-inclusive ones, have shown to over-estimate the treatment effect by only 2% on average [83], and we expect the impact of language or geographical region to be way less important for metaheuristic compared to clinical research.

The second type of bias, the search bias, also occurs in the study identification phase and is related to missing relevant studies with a faulty search. Search bias is minimized with a set of key words that is as complete as possible (and the search engine and key words should explicitly be mentioned). We restricted the search to Google Scholar and the records with ALNS in the title for practical purposes, which resulted in search bias, due to an incomplete retrieval of ALNS-related research. Furthermore, if the ALNS has been compared to the non-adaptive alternative in order to assess the added value of the adaptive layer, and if it did not prove to have any benefits, the authors would most likely remove the adaptive framework. Most often this would not even be mentioned in the paper (in order to publish, the focus is unfortunately almost exclusively on positive results), but even if this is the case, such articles would not have ALNS in the title and would therefore not be identified with our search. This has been confirmed to us through personal correspondence. For example, according to Fabien Lehéudé, the adaptive layer was evaluated and shown to be useful in [33, 37] (ALNS is in the title), but was not significantly helpful in [34, 35, 68, 123] (no ALNS in the title), although this is not always explicitly shown or mentioned in the paper. Olivier Péton also confirmed that, although ALNS was shown to be slightly better than (¬A)LNS in [80], on a number of other occasions there was no significant difference between the two algorithms and eventually the adaptive layer was removed (so that there would be no ALNS in the title). We hope to have decreased this bias somewhat through personal correspondence with the researchers, who directed us towards other papers relevant to the topic under study.

Finally, selection bias, as the name suggests, can occur in the study selection phase due to poorly defined eligibility criteria. We minimized the risk of bias with a clear definition of inclusion criteria, given in Section 4.

A funnel plot is sometimes used for meta-analyses in clinical research to assess the identification (publication and search) and selection bias. The funnel plot is a scatter plot of the individual studies in a meta-analysis, with the treatment effect on the x-axis and some measure of study precision, such as the sample size, the inverse variance, or the standard error, on the y-axis [120]. In the absence of bias, the funnel plot is expected to show a symmetrical distribution about the mean effect size. However, asymmetry in a funnel plot is not necessarily resulting from a bias, but could also be a consequence of, e.g., overestimation of treatment effects in smaller studies of inadequate methodological quality [120]. Asymmetry of the funnel plot, either visually interpreted or statistically tested, was shown not to accurately predict bias [67], and many authors have by now argued that visual interpretation of funnel plots is too subjective to be useful [50]. For these reasons, we refrained from using a funnel plot in this paper.

6 Statistical analysis

Most meta-analyses are based on one of two statistical models: fixed-effect or random-effects model. The fixed-effect model assumes that there is one true effect size (hence the term *fixed-effect*) which underlies all studies in the analysis, and that all differences in observed effects are due to sampling error. The summary effect is then an estimate of this common effect size. The random-effects model assumes that the true effect could vary from study to study. For example, in clinical research, the effect size might be higher (or lower) in studies where the participants are older, or more educated, or healthier than in others, or when a more intensive variant of treatment is used. The effect sizes in the included studies are assumed to represent a random sample of the effect sizes which could have been observed (hence the term *random-effects*). The summary effect is then an estimate of the mean (of the distribution) of these effects [12].

In our meta-analysis of the ALNS adaptive layer, we cannot assume that the importance of the adaptive layer is the same for each study. Indeed, there are a number of study characteristics which we identified in Section 4 that could influence the importance of adaptiveness. In other words, the studies we considered do not investigate the same population, but rather different populations of instances for different problems. For this reason, our primary statistical analysis is based on a random-effects model.

For both the fixed-effect and the random-effects model, the summary estimate is a weighted average of the effect sizes in the individual studies. The individual study results are weighted by some measure of their precision, most commonly with the inverse of their variance. In statistics, inverse-variance weighting is a method of aggregating two or more random variables to minimize the variance of the weighted average.

The only difference between the fixed-effect and random-effects model lies in the source(s) of variance. Under the fixed-effect model there is one level of sampling - subjects are sampled from a population, and therefore there is only one source of variance. By contrast, under the random-effects model there are two levels of sampling and thus two sources of variance. In our example of meta-analysis of meta-heuristics, we sample instances of a problem under consideration in a study, but we also sample problems (or studies) from the population of problems. The first source of variance is thus the difference between the observed effect of the study and the true effect of the study. This within-study variance is commonly estimated with the squared standard error:

$$V_i = \frac{\sigma_i^2}{N_i},$$

(σ_i is the standard deviation, and N_i is the number of problem instances in study i). It makes sense to weigh studies with the inverse variance: we assign more weight to the studies which include a greater number of instances, and for which the dispersion of the effect size across instances is small.

The second source of variance is the variance of the effect sizes across the population of studies. This between-study variance is commonly estimated using the DerSimonian and Laird method [26] in the following way:

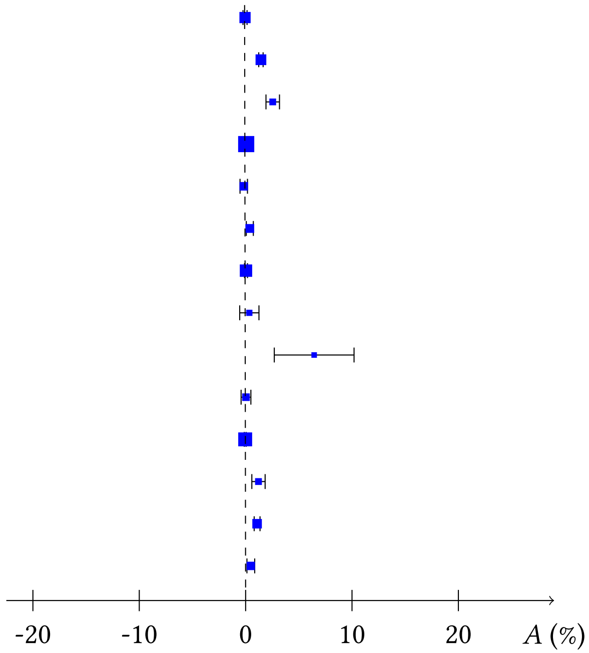
$$T^2 = \frac{Q - df}{C},$$

where statistic Q is a sum of squares of the effect size estimates about their mean, weighted by the inverse of variance V_i, $df = K - 1$ and K is the number of studies included in the meta-analysis, and C is simply a factor which puts the standardized variation between studies $Q - df$ back into the same metric that had been used to report the within-study variance. For more details on these statistics, see [12]. Whereas the only variance in the fixed-effect model is V_i, the total variance under the random effects model is $V_i + T^2$, so that the weight of study i is calculated as:

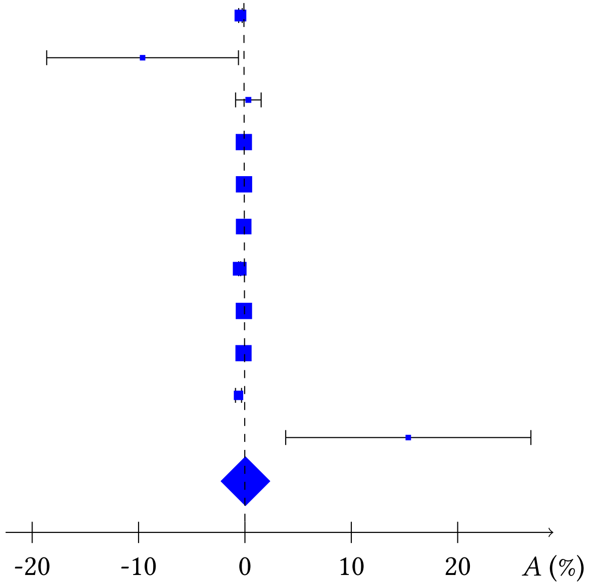
$$W_i = \frac{1}{V_i + T^2}.$$

The relative weights assigned under a random-effects model are more balanced than those assigned under fixed effect model, due to the addition of the between-study variance T^2. This also makes sense: when assigning weights to the different studies under the fixed-effect model, we can largely ignore the information in the smaller studies since we have better information about the common effect size in the larger studies. Under the random-effects model, each study provides information about a different effect size, and we want to be sure that all these effect sizes are represented in the summary estimate. This means that we cannot disregard a small study by giving it a very small weight. The estimate provided by that study may be imprecise, but it is information about an effect that no other study has estimated [12].

Table 4: On average, the adaptive layer helps to improve the ALNS performance by $A = 0.14\%$. The summary effect A, represented with a diamond, is a weighted average of the added value of adaptiveness across a number of studies included in the meta-analysis. The mean effect in each of the studies is represented with a square (with a 95% confidence interval around it), where the size of the square reflects the weight of the study. The percentage of problem instances for which ALNS outperforms the non-adaptive variant (¬A)LNS, or the other way around, provides additional information on the behaviour of the adaptive layer in each of the included studies.

Article	Mean A	Standard deviation	N	Weight	Forest plot	% better	% worse
[11]	0.04	1.26	108	32.23		45.37	29.63
[19] ALNS(6,4)	1.53	1.43	117	29.64		87.18	4.27
[19] ALNS(4,3)	2.64	3.78	117	7.21		84.62	10.26
[23]	0.15	0.16	62	59.99		95.16	1.61
[37]	-0.08	0.35	3	17.18		33.33	66.67
[42]	0.49	1.49	60	18.71		56.67	35.00
[54]	0.13	0.53	30	38.91		46.67	43.33
[59]	0.45	3.64	56	3.95		25.00	8.93
[60]	6.54	8.89	21	0.26		85.71	4.76
[63]	0.13	3.12	144	11.90		45.14	45.14
[75]	0.06	1.19	296	47.43		45.95	52.70
[79]	1.31	1.03	9	7.45		100.00	0.00
[81]	1.72	1.29	60	22.76		51.67	0.00
[85]	0.59	3.23	233	16.37		8.58	2.58

On average, the adaptive layer helps to improve the ALNS performance by $A = 0.14\%$. The summary effect A, represented with a diamond, is a weighted average of the added value of adaptiveness across a number of studies included in the meta-analysis. The mean effect in each of the studies is represented with a square (with a 95% confidence interval around it), where the size of the square reflects the weight of the study. The percentage of problem instances for which ALNS outperforms the non-adaptive variant (¬A)LNS, or the other way around, provides additional information on the behaviour of the adaptive layer in each of the included studies.

Article	Mean A	Standard deviation	N	Weight	Forest plot	% better	% worse
[93]	-0.32	0.47	18	35.33		5.56	83.33
[103] *gfbj*	-9.52	21.69	22	0.05		4.61	96.80
[103] *mine*	0.42	2.02	10	2.37		40.00	60.00
[108] CVRP	-0.00	0.17	159	60.80		49.06	42.14
[108] CMST	0.01	0.12	104	60.98		21.15	21.15
[108] QAP	-0.02	0.36	99	56.86		23.23	20.20
[107]	-0.40	1.31	324	46.38		25.93	63.27
[109]	0.00	0.38	1 455	61.11		10.93	10.45
[110]	-0.03	1.99	11 705	60.23		15.67	15.90
[123]	-0.50	0.64	14	22.04		21.43	78.57
[124]	15.46	26.41	20	0.03		75.00	10.00
Average	0.14						

The results of our statistical analysis are summarized in the forest plot shown in Table 4. To improve the clarity of this section and the results, we include a toy example of a step-by-step meta-analysis in the Appendix (Table 5). For most of the studies included in the meta-analysis, the average objective function value is calculated across 5 runs, but for some 10 or 20 runs were carried out for each instance. For more details, interested readers are referred to the complete results per study and per instance, which are publicly available on the corresponding author's website.

The results suggest that adaptiveness helps to improve ALNS performance on average by $A = 0.14292\%$ (with a standard error of 0.03738, and a 95% confidence interval between 0.06819 and 0.21764). The null hypothesis being tested is that the mean effect is zero, i.e., that the ALNS adaptive layer has no added value, with a p-value of 0.00018. In other words, if the null hypothesis is true, the probability of finding that $A \geq 0.14292\%$ is 0.00018. This means that there is strong evidence (at level of significance 0.001) against the null hypothesis.

7 Discussion

We start this section by providing more information on the two important outliers in the forest plot (Table 4), with $A_i = 15.46\%$ [124] and $A_i = 6.45\%$ [60] (subscript i denotes study i). This helps us to further extend the discussion about the importance of adaptiveness in ALNS.

In [124], ALNS is envisioned as a black box optimizer which can solve any (or, a very broad range of) combinatorial problem(s). This ALNS includes 30 destroy and 36 repair heuristics, and each of them is known to work well for one of the 10 different problems which were considered in the computational experiments. It is therefore not surprising that the adaptive layer was able to improve the performance by 15.46%. Indeed, Stefan Røpke, who introduced ALNS, and Alberto Santini, agree that the adaptive layer is expected to be important if ALNS is going to be applied to different problems (e.g., vast range of routing problems), or a diverse set of instances for one problem, where we expect that some destroy or repair heuristics would perform very poorly or very good on some subsets of instances (e.g., on symmetric vehicle routing problem). Røpke affirmed that, if ALNS is used to solve a particular problem and one does not plan to implement destroy or repair heuristics that are targeted to a specific type of instances, the adaptive layer is expected not to matter, or matter very little. In [124], the authors explicitly mention that different heuristics are targeting different subsets of problem instances, e.g., for one of the destroy heuristics, the authors state that it should be efficient on problems where successive decision variables are related to each other (such as lot sizing problems), whereas another should be efficient on problems where values represent resources shared between variables (such as bin-packing problems).

We also note that a different weight adjustment procedure is employed in [124]. We can see in Table 2 that the weights of the heuristics are updated with the gradient $\frac{\Delta f}{\Delta t}$ of the objective function over time (evaluated on the current best solution). However, we also note that the authors do not employ the weight adjustment formula (2). The authors propose a variation in the weight-update mechanism, based on a so called sliding evaluation window, which is especially useful when the different heuristics exhibit unequal running times. For this study, the ALNS with the standard weight adjustment formula (2) outperforms the equiprobable (¬A)LNS by 10.69%. This means that, if there are less useful heuristics which also take a lot of computation time, and the computation time is not taken properly into account when updating the heuristic weights, the adaptive layer can be expected to have a greater importance.

Furthermore, in [124], LNS with the single best performing pair of destroy and repair heuristic outperforms any other (adaptive, or non-adaptive) variant. This means that neither of the two considered variants of the adaptive layer were intelligent enough, or fast enough, to focus solely on the heuristics which are successful for a given problem instance (by assigning them a probability of being selected close to one).

That the choice of heuristics seems to be crucial for ALNS performance is also indicated in the second outlier. In [60], ALNS with 10 destroy and 8 repair heuristics is considered, and it is shown that it improves over the non-adaptive (¬A)LNS by 6.54% (Table 4). However, we see from an analysis in the article that the equiprobable non-adaptive variant with a subset of only 10 destroy and 4 (most successful) repair heuristics even outperforms ALNS with the same 10 destroy and 4 repair heuristics. The importance of the adaptive layer is thus lower if the set of heuristics is already narrowed down to the successful ones. In the latter comparison, the iteration limit is adjusted to reflect the differences in the computational effort, and the more appropriate non-adaptive variant without the adaptiveness-related code is considered, which also speaks to the importance of two sources of bias discussed in Section 5.1. The author pointed out that the deviation from the other ALNS studies might partially also be accredited to the problem characteristics. For the considered timetabling problem, rather artificial integer penalties are used for soft constraint violations in the objective function, so that the percentage deviation in the objective function value between two solutions with one additional penalty is quite high.

In any case, we note that results are robust with respect to the outliers: the summary effect A remains 0.14 even when the three outliers [124], [60] and [103] (*gfbj* instances) are excluded from meta-analysis (A changes from 0.14292 to 0.13707).

There are further indications that, if the set of heuristics is narrowed down, the adaptive layer might not be so useful. In [103], the authors show that ALNS with all 8 heuristics performs worse for almost any instance than some other strict subsets of heuristics, which suggests that the adaptive mechanism is not able to adjust the weights in an optimal way (i.e., focusing only on the subset of heuristics which perform well for the given instance). In the literature review in Section 2, we mention that ALNS was shown to be better than the three non-adaptive versions, with weights fixed in different ways in [15]. However, we can also see in that paper that the non-adaptive versions (with the set of smart heuristics) are better than many versions of ALNS with a greater number of heuristics. In [28], the results obtained by ALNS are clearly worse compared to the configuration with an offline selection of heuristics by irace (a package for automatic configurations of algorithmic parameters [73]). This suggests that adaptiveness cannot effectively compensate for a poor offline selection of heuristics. The effectiveness of ALNS therefore primarily lies in the choice of heuristics, which are usually based on proven performance methods for the problem at hand [109]. It therefore comes as no surprise that for [110], where the choice of heuristics is narrowed down with irace to 5 destroy and 4 repair heuristics (from respectively 10 and 5), the importance of adaptiveness is $A_i = -0.03\%$. On a related note, [95] shows that adapting Ant Colony Optimization parameters online is ineffective as long as the values of these parameters are carefully tuned offline (for the traveling salesman problem and quadratic assignment problem, and a particular implementation).

In our literature review and meta-analysis, there is one example where the findings above are not confirmed. In [19], the final algorithm which is run for every instance is a random choice between an ALNS with 6 destroy and 4 repair heuristics, and an ALNS with a subset of only 4 destroy and 3 repair heuristics (as one is better with respect to solution quality, and the other with respect to computation time). To have a proper evaluation of the added value of the adaptive layer, we asked the authors to compare the adaptive and non-adaptive algorithm for the two versions of ALNS separately (rather than randomly choosing between them). We can see in Table 4 that $A_i = 2.64\%$ for ALNS(4,3), and $A_i = 1.53\%$ for ALNS(6,4), which contradicts the conclusion above suggesting a lower importance of the adaptive layer when the choice of heuristics is narrowed down. However, the results we received also show that ALNS(6,4) finds solutions which are, on average, 3.30% better than ALNS(4,3), implying that the choice of heuristics is not reduced well (with respect to the solution quality) in this case.

Different weight adjustment mechanisms have been used in the literature, and some might be more successful than others (for certain types of problems, or other characteristics). In [76], the selection probabilities look completely different for two runs of the ALNS on the same instance, and the authors therefore conclude that the current weight adjustment formula is far from optimal. Additionally, the success percentages (the number of times the new best, or a better solution was found) of the most frequently used heuristics are not necessarily the highest [30, 76, 78]. This effect can be explained by the fact that some heuristics tend to have success in the earlier iterations of the algorithm, thus increasing the weight and the probability to be selected for further iterations in which they might be far less successful. It is indeed easier to find better solutions in the earlier stages of the algorithm, and one way to compensate for this is to include the current iteration number in the score, to give more importance to the rewards at the end of the search [77]. Some authors also argue that the assumption that past success implies future success might be very wrong (e.g., diversification might be more important in the early phase of the algorithm) [76]. Other weight adjustment mechanisms have been proposed in the literature, e.g., the selection of heuristics is guided in [49] by selection strategies for the multi armed bandit problem. In [13], the authors discuss four different scoring mechanisms, but also two different selection principles, with the goal of making the differences between heuristic weights matter (gradually) more or less than in the common roulette wheel principle. To the best of our knowledge, there exist no studies that investigate this particular topic, and we believe it is an interesting question for future research (possibly taking some inspiration from machine learning).

The studies included in our meta-analysis differ further with respect to a number of characteristics (Table 2), which could have an influence on the importance of the adaptive layer, such as the problem type, objective function, acceptance and stopping criterion, and so on, or any other confounding factor. The number of studies is too small to carry out additional sensitivity or subgroup analyses which would investigate the relationship between these factors and the added value of adaptiveness. For example, it was shown in [23] that there is a 0.23 – 0.29% degradation if the weights of destroy and repair heuristics are considered individually rather than in pairs, but next to [63, 93], this is the only study included in our meta-analysis with pairwise rather than independent evaluation of heuristics, so this question cannot be properly answered from our analysis.

However, both detailed results per study and per instance, and the summary results and calculations for our meta-analysis, are made publicly available on:

`http://antor.uantwerpen.be/members/renata-turkes/`

so that the analysis can be replicated, and more importantly, so that the results of other studies could be included. When more articles become available which evaluate the effect of adaptiveness, this meta-analysis can easily be extended and further insights can be obtained. For instance, further sensitivity analyses, sub-group analyses, or meta-regression could then help to study the effect of different study characteristics (problem type, number of heuristics, reaction factor, scores or stopping criterion, noise) and their interactions on the importance of adaptiveness in ALNS.

This article also highlights the importance of sharing code and instances. Most of the articles which introduce ALNS for a problem focus on the comparison of ALNS performance compared to the best known solutions from the literature, but these computational experiments cannot be replicated or verified, and any other question the reader might have can rarely be answered from the articles themselves.

8 Conclusions, limitations and future work

The goal of this paper is to promote meta-analysis as a good tool of obtaining knowledge about metaheuristic frameworks. Meta-analysis is a systematic approach to identifying, apraising, synthesizing, and combining the results of relevant studies to arrive at conclusions about a body of research [121]. In addition, meta-analysis has the capacity to contrast results from different studies, identify patterns among study results, sources of disagreement among those results, or other interesting relationships. Meta-analyses have gained great popularity in clinical research, but we believe them to be a great tool of generating new insights also in Operations Research. To illustrate the approach, we have carried out a meta-analysis of the importance of ALNS adaptive layer.

Our meta-analysis suggests that the benefit of the adaptive layer is minimal: adaptiveness helps to improve ALNS performance only by 0.14%. This number is far lower what one might expect when considering the popularity of this metaheuristic framework. This has been confirmed by a number of researchers in the domain through personal correspondence, who affirmed that the adaptive layer was not beneficial even for the majority of the articles they worked on (and was eventually removed, e.g., in [34, 35, 68, 123]). If ALNS is developed, it is therefore important to properly evaluate if the adaptive layer is beneficial, rather than blindly copying the algorithmic structure from some other implementation. Without a doubt, the same is true for any other seemingly crucial algorithmic components. Different selection probabilities across heuristics do not imply that the adaptive layer is advantageous: the differences among heuristic weights were shown to be statistically significant in [109], but the effect of adaptiveness is $A_i = 0.00\%$. We suggest to also explicitly mention these outcomes (of evaluation of algorithmic components, either positive or negative) and thereby contribute to our knowledge and understanding of metaheuristics.

There is continued evidence that, if the mostly useless (and in particular useless and slow) heuristics are removed, the adaptive layer does not have a major impact. Indeed, it has already been shown in [28] that a priori heuristic selection is crucial and that the roulette wheel mechanism cannot effectively compensate for the lack of it (for an ALNS focused on a multi-trip vehicle routing problem). In other words, eliminating some of the destroy and repair heuristics offline, before the online selection through the roulette wheel, is essential. If we therefore aim to develop a good performing algorithm (e.g., win a horse race) and we know how the instances can be expected to look like, whereas no specialized heuristics are targeting a particular subset of instances (e.g., a subset corresponding to one of the considered problems, one variant of the problem, or a particular type of instances even of a single problem), the adaptive layer will probably not have a major impact. In this case, the heuristic selection probabilities can be tuned, or even left equal. The final algorithm can therefore be significantly simplified by removing both some of the heuristics and the adaptive layer: keeping track of heuristic scores and weights, and visited solutions, or checking if a solution has been visited before (thereby also reducing the heuristic computational effort), and the parameters of the adaptive layer (scores, reaction factor and number of segments) do not need to be tuned.

There are some reasons why one might want to include "useless" (and slow) heuristics in the solution algorithm. One example is when such a heuristic is targeted to a special case, to a particular subset of problem instances: although it performs poorly for many instances, this heuristic can be very important for others. To a certain extent, the adaptive layer can guide the selection of heuristics for a given instance, and in this extreme scenario it can be sufficient to improve the algorithmic performance. In this scenario, however, it would also be possible to assign a small (fixed) weight to the heuristic which targets the particular special case, or to tune the heuristic weights offline. Another example is when not enough test instances are available to know the properties of instances to be solved, and the robustness of the algorithm is a priority.

The choice of heuristics therefore has a crucial influence on the importance of the adaptive layer. For example, in an extreme case where thousands of destroy and repair heuristics would be included in ALNS, but with most of them being ineffective for the given problem, we can expect that ALNS would significantly outperform (¬A)LNS. No matter the outcome of our meta-analysis of this particular question, we cannot therefore ever conclude that adaptiveness could never be beneficial. The outcome does, however, suggest that, depending on one's main purpose for developing a solution algorithm which incorporates a number of destroy and repair heuristics, and how well these heuristics are selected, the adaptive layer can be redundant.

Although the choice of heuristics seems to be crucial for the importance of ALNS adaptiveness, the studies included in our meta-analysis differ with respect to a number of characteristics (Table 2), which could have an influence on the importance of the adaptive layer, such as the problem type, objective function, weight adjustment mechanism, and so on, or other confounding factors. Unfortunately, we only received the results from a limited number of studies, which did not make it possible to carry out additional analyses of the relationship between these factors and the added value of adaptiveness. If results from other studies become available, this question could be revisited to gain further insights.

It would also be interesting to carry out more meta-analyses to answer other questions. An example would be an analysis of the importance of having a variable length tabu list in reactive tabu search (some answers are provided in [96] for the quadratic assignment and the maximum clique problem). A better understanding of the acceptance criterion has been indicated as the key to improving the ALNS performance [47], but only for three problems and a specific implementation with a fixed stopping criterion and other properties, and it might therefore be interesting to investigate this further by considering multiple studies. There is a number of limitations of meta-analyses (e.g., the different sources of bias discussed in Section 5, although we note that these are also present in any other literature review), but we have demonstrated in this paper that it can help us gain a better understanding of metaheuristics.

Author contributions

Renata Turkeš carried out the literature review, identified and selected the studies, communicated with the authors of eligible studies via e-mail, collected and processed the data, performed the meta-analysis and wrote the manuscript. Kenneth Sörensen and Lars Magnus Hvattum came up with the idea of performing meta-analysis to obtain general insights about metaheuristics, and assisted with writing of the paper. In addition, Kenneth Sörensen suggested to investigate the importance of the ALNS adaptive layer as an example of an interesting question that can be studied with a meta-analysis. Eva Barrena, Hayet Chentli, Leandro Coelho, Iman Dayarian, Axel Grimault, Anders Gullhav, Çağatay Iris, Merve Keskin, Alexander Kiefer, Richard Lusby, Geraldo Mauri, Marcela Monroy-Licht, Sophie Parragh, Juan-Pablo Riquelme-Rodríguez, Alberto Santini, Vinicius Gandra Martins Santos and Charles Thomas sent the results from their earlier papers, which were included in the meta-analysis. This often meant that additional experiments were carried out in order to obtain the comparison of ALNS with the non-adaptive variant. Alexander Kiefer carried out further experiments that provided better insights into the bias in individual studies (Table 3).

Acknowledgments

We would like to acknowledge Simona Mancini and Oscar Tellez who explicitly analyzed the added value of the ALNS adaptive layer and included the detailed experimental results per instance in their articles. This made it possible for us to include their results in our meta-analysis without having to send a request for data. We are also thankful to Michel Gendreau, Richard Hartl, Fabien Lehuédé, Olivier Péton, and Stefan Røpke who shared their insights about adaptive large neighborhood search, most of which have been included in this paper.

Appendix

Table 5: A toy example of a step-by-step meta-analysis with two included studies. The information in the highlighted rectangle corresponds to the data collected from the articles, or requested from authors: objective function value for ALNS and the non-adaptive variant, for a number of problem instances and across a number of runs. This information is used to estimate the mean importance A_i of the adaptive layer, i.e., the average percentage improvement of ALNS upon the non-adaptive algorithm, for a study i. The summary effect A is a weighted average of the effect across studies, with the weight calculated as the inverse variance. In a random-effects model, the variance is calculated as the sum of within-study variance (square of standard error, which incorporates both the standard deviation across problem instances within a study, and a number of instances) and across study variance (estimated with poor precision when the number studies is very small). The summary effect A in this example is weighted more strongly towards A_1 than A_2, since the weight of the study S_2 is very small: indeed, the adaptive layer improves the algorithmic performance by 0.64% for one instance, 5.71% for another instances, but by −0.33% for the last instance, and we are therefore less confident about the true effect of adaptiveness in this study (i.e., the standard deviation is large, and hence the weight is small, indicating that the estimate A_2 is less precise).

Study S_i	Instance I_{ij}	Run	$f(x_0^*(I_{ij}))$	$f(x_r^*(I_{ij}))$	$\bar{f}(x_0^*(I_{ij}))$	$\bar{f}(x_r^*(I_{ij}))$	$100 \times [\bar{f}(x_r^*(I_{ij})) - \bar{f}(x_0^*(I_{ij}))]/\bar{f}(x_0^*(I_{ij}))$	Effect A_i	Standard deviation σ_i	Number of instances N_i	Within-study variance $V_i = \frac{\sigma_i}{N_i}$	Between-study variance T^2	Weight $W_i = \frac{1}{V_i + T^2}$	Normalized weight $\bar{W}_i$	Weighted effect $\bar{W}_i \times A_i$
S_1	I_{11}	1	856.0	863.0	855.00	866.50	1.35	1.00	0.50	2	0.13	0	7.93	0.97	0.97
		2	854.0	870.0											
	I_{12}	1	40.0	39.0	39.00	39.25	0.64								
		2	38.0	39.5											
S_2	I_{21}	1	1200.0	1208.0	1200.00	1206.5	0.64	1.97	3.26	3	3.55		0.28	0.03	0.07
		2	1200.0	1205.0											
	I_{22}	1	10.0	10.5	10.5	11.1	5.71								
		2	11.0	11.7											
	I_{23}	1	301.0	299.0	300.5	299.5	-0.33								
		2	300.0	300.0											
															A = 1.04

Table 6: Important notation and acronyms.

Notation	
$\mathcal{H}$	set of (destroy or repair) heuristics
$\mathcal{D}$	set of destroy heuristics
$\mathcal{R}$	set of repair heuristics
h	(destroy or repair) heuristic (can be referred also as l)
s	segment, a number of consecutive iterations
δ	score added to a heuristic in each iteration it has been called
π_h	total score of heuristic h at the end of the current segment
θ_h^s	number of times heuristic h has been called during the current segment
r	reaction factor, which controls the importance of heuristic performance in the last segment
w_h^s	weight of heuristic h updated after segment s (calculated as $(1-r)w_h^{s-1} + r\pi_h/\theta_h$)
f	objective function
$\overline{f}(\cdot)$	average objective function value across a number of runs
x_r^*	best solution found by ALNS, with tuned r
x_0^*	best solution found by non-adaptive (¬A)LNS, with $r = 0$
K	number of studies in the meta-analysis
S_i	study S_i (an individual study in the meta-analysis)
I_{ij}	instance j in study S_i
A_i	improvement with the adaptive layer in study S_i, averaged across instances and runs
N_i	number of instances in study S_i
σ_i	standard deviation in study S_i
V_i	variance within study S_i (calculated as σ^2/N)
T^2	variance across studies
W_i	weight of study S_i (calculated as inverse total variance $1/(V_i + T^2)$)
A	improvement with the adaptive layer, weighted across studies
Acronyms	
LNS	large neighborhood search (single pair of destroy and repair heuristics)
ALNS	adaptive large neighborhood search
(¬A)LNS	ALNS without the adaptive layer
ALNS($\lvert\mathcal{D}\rvert$, $\lvert\mathcal{R}\rvert$)	ALNS with $\lvert\mathcal{D}\rvert$ destroy and $\lvert\mathcal{R}\rvert$ repair heuristics

References

[1] Yossiri Adulyasak, Jean-François Cordeau, and Raf Jans. Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science*, 48(1):20–45, 2012.

[2] Wasif Afzal and Richard Torkar. On the application of genetic programming for software engineering predictive modeling: A systematic review. *Expert Systems with Applications*, 38(9):11984–11997, 2011.

[3] Wasif Afzal, Richard Torkar, and Robert Feldt. A systematic review of search-based testing for non-functional system properties. *Information and Software Technology*, 51(6):957–976, 2009.

[4] Eduardo López Aguilar, Yannick Kergosien, Vincent Boyer, and Jean-Charles Billaut. An adaptive large neighborhood search for an appointment scheduling problem in

health care. In *The French Operations Research & Decision Support Society ROADEF 2017*.

[5] Deniz Aksen, Onur Kaya, F Sibel Salman, and Özge Tüncel. An adaptive large neighborhood search algorithm for a selective and periodic inventory routing problem. *European Journal of Operational Research*, 239(2):413–426, 2014.

[6] Mahdi Alinaghian and Nadia Shokouhi. Multi-depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhood search. *Omega*, 76:85–99, 2018.

[7] Yasemin Arda, Yves Crama, and Véronique François. An adaptive large neighborhood search for a vehicle routing problem with multiple trips and driver shifts. In *The French Operations Research & Decision Support Society ROADEF 2013*.

[8] Nabila Azi, Michel Gendreau, and Jean-Yves Potvin. An adaptive large neighborhood search for a vehicle routing problem with multiple routes. *Computers & Operations Research*, 41:167–173, 2014.

[9] Lukas Bach, Geir Hasle, and Christian Schulz. Adaptive large neighborhood search on the graphics processing unit. *European Journal of Operational Research*, 275(1): 53–66, 2019.

[10] Rikard Bakkehaug, Jørgen G Rakke, Kjetil Fagerholt, and Gilbert Laporte. An adaptive large neighborhood search heuristic for fleet deployment problems with voyage separation requirements. *Transportation Research Part C: Emerging Technologies*, 70: 129–141, 2016.

[11] Eva Barrena, David Canca, Leandro Coelho, and Gilbert Laporte. Single-line rail rapid transit timetabling under dynamic passenger demand. *Transportation Research Part B: Methodological*, 70:134–150, 2014.

[12] Michael Borenstein, Larry V Hedges, Julian PT Higgins, and Hannah R Rothstein. *Introduction to meta-analysis*. John Wiley & Sons, 2011.

[13] Simen Braaten, Ola Gjønnes, Lars Magnus Hvattum, and Gregorio Tirado. Heuristics for the robust vehicle routing problem with time windows. *Expert Systems with Applications*, 77:136–147, 2017.

[14] Maurizio Bruglieri, Ferdinando Pezzella, and Ornella Pisacane. An adaptive large neighborhood search for relocating vehicles in electric carsharing services. *Discrete Applied Mathematics*, 253:185–200, 2019.

[15] Elivelton Ferreira Bueno. *Mathematical Modeling and Optimization Approaches for Scheduling the Regular-Season Games of the National Hockey League*. PhD thesis, École polytechnique de Montréal, 2014.

[16] David Canca, Alicia De-Los-Santos, Gilbert Laporte, and Juan A Mesa. The railway network design, line planning and capacity problem: An adaptive large neighborhood search metaheuristic. In *Advanced Concepts, Methodologies and Technologies for Transportation and Logistics*, pages 198–219. Springer, 2016.

[17] S Chaharsooghi, F Momayezi, and N Ghaffarinasab. An adaptive large neighborhood search heuristic for solving the reliable multiple allocation hub location problem under hub disruptions. *International Journal of Industrial Engineering Computations*, 8(2):191–202, 2017.

[18] Shifeng Chen, Rong Chen, Gai-Ge Wang, Jian Gao, and Arun Kumar Sangaiah. An adaptive large neighborhood search heuristic for dynamic vehicle routing problems. *Computers & Electrical Engineering*, 67:596–607, 2018.

[19] Hayet Chentli, Rachid Ouafi, and Wahiba Ramdane Cherif-Khettaf. A selective adaptive large neighborhood search heuristic for the profitable tour problem with simultaneous pickup and delivery services. *RAIRO-Operations Research*, 52(4):1295–1328, 2018.

[20] Jonas Mark Christensen and Stefan Røpke. An adaptive large neighborhood search-based three-stage matheuristic for the vehicle routing problem with time windows. In *Odysseus 2015; Sixth International Workshop on Freight Transportation and Logistics*, 2015.

[21] Ceire Costelloe, Chris Metcalfe, Andrew Lovering, David Mant, and Alastair D Hay. Effect of antibiotic prescribing in primary care on antimicrobial resistance in individual patients: systematic review and meta-analysis. *The BMJ*, 340:c2096, 2010.

[22] Luciano Perdigão Cota, Frederico Gadelha Guimarães, Fernando B de Oliveira, and Marcone J Freitas Souza. An adaptive large neighborhood search with learning automata for the unrelated parallel machine scheduling problem. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pages 185–192. IEEE, 2017.

[23] Iman Dayarian, Teodor Gabriel Crainic, Michel Gendreau, and Walter Rei. An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 95:95–123, 2016.

[24] William A Dees and Patrick G Karger. Automated rip-up and reroute techniques. In *19th Design Automation Conference*, pages 432–439. IEEE, 1982.

[25] Emrah Demir, Tolga Bektaş, and Gilbert Laporte. An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223(2):346–359, 2012.

[26] Rebecca DerSimonian and Nan Laird. Meta-analysis in clinical trials. *Controlled clinical trials*, 7(3):177–188, 1986.

[27] Uğur Emeç, Bülent Çatay, and Burcin Bozkaya. An adaptive large neighborhood search for an e-grocery delivery routing problem. *Computers & Operations Research*, 69:109–125, 2016.

[28] Véronique François, Yasemin Arda, Yves Crama, and Gilbert Laporte. Large neighborhood search for multi-trip vehicle routing. *European Journal of Operational Research*, 255(2):422–441, 2016.

[29] Véronique François, Yasemin Arda, and Yves Crama. Adaptive large neighborhood search for multi-trip vehicle routing with time windows. *Transportation Science*, 2017.

[30] Veaceslav Ghilas, Emrah Demir, and Tom Van Woensel. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines. *Computers & Operations Research*, 72:12–30, 2016.

[31] Leslie Allison Gillum, Sai Kumar Mamidipudi, and S Claiborne Johnston. Ischemic stroke risk with oral contraceptives: a meta-analysis. *Jama*, 284(1):72–78, 2000.

[32] Gene V Glass. Primary, secondary, and meta-analysis of research. *Educational Researcher*, 5(10):3–8, 1976.

[33] Philippe Grangier, Michel Gendreau, Fabien Lehuédé, and Louis-Martin Rousseau. An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. *European Journal of Operational Research*, 254(1):80–91, 2016.

[34] Philippe Grangier, Michel Gendreau, Fabien Lehuédé, and Louis-Martin Rousseau. A matheuristic based on large neighborhood search for the vehicle routing problem with cross-docking. *Computers & Operations Research*, 84:116–126, 2017.

[35] Philippe Grangier, Michel Gendreau, Louis-Martin Rousseau, and Fabien Lehuédé. *The Vehicle Routing Problem with Cross-docking and Resource Constraints*. Journal of Heuristics (to appear), 2019.

[36] Axel Grimault, Nathalie Bostel, Willy Lambert, and Fabien Lehuédé. An adaptive large neighborhood search for a full truckload routing problem in public works. In *Odysseus 2015*, 2015.

[37] Axel Grimault, Nathalie Bostel, and Fabien Lehuédé. An adaptive large neighborhood search for the full truckload pickup and delivery problem with resource synchronization. *Computers & Operations Research*, 88:1–14, 2017.

[38] Glioma Meta-analysis Trialists GMT Group. Chemotherapy in adult high-grade glioma: a systematic review and meta-analysis of individual patient data from 12 randomised trials. *The Lancet*, 359(9311):1011–1018, 2002.

[39] Timo Gschwind and Michael Drexl. Adaptive large neighborhood search with a constant-time feasibility test for the dial-a-ride problem. *Transportation Science*, 2019.

[40] Wenjuan Gu, Diego Cattaruzza, Maxime Ogier, and Frédéric Semet. Adaptive large neighborhood search for the commodity constrained split delivery vrp. In *OR 2018 International Conference on Operations Research*, 2018.

[41] Wenjuan Gu, Diego Cattaruzza, Maxime Ogier, and Frédéric Semet. Adaptive large neighborhood search for multicommodity vrp. In *Odysseus 2018, the Seventh International Workshop on Freight Transportation and Logistics*, 2018.

[42] Anders N Gullhav, Jean-François Cordeau, Lars Magnus Hvattum, and Bjørn Nygreen. Adaptive large neighborhood search heuristics for multi-tier service deployment problems in clouds. *European Journal of Operational Research*, 259(3):829–846, 2017.

[43] Michael Haase, Rinaldo Bellomo, Prasad Devarajan, Peter Schlattmann, Anja Haase-Fielitz, and NGAL Meta-analysis Investigator Group. Accuracy of neutrophil gelatinase-associated lipocalin (ngal) in diagnosis and prognosis in acute kidney injury: a systematic review and meta-analysis. *American Journal of Kidney Diseases*, 54(6):1012–1024, 2009.

[44] Mina Hadianniasar. An adaptive large neighborhood search heuristic for the inventory routing problem with time windows. Master's thesis, University of Arkansas, 2015.

[45] Makbul Hajad, Viboon Tangwarodomnukun, Chorkaew Jaturanonda, and Chaiya Dumkum. Laser cutting path optimization using simulated annealing with an adaptive large neighborhood search. *The International Journal of Advanced Manufacturing Technology*, pages 1–12, 2019.

[46] Xiaolu He, Lei and, Gilbert Laporte, Yingwu Chen, and Yingguo Chen. An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling. *Computers & Operations Research*, 100:12–25, 2018.

[47] Ahmad Hemmati and Lars Magnus Hvattum. Evaluating the importance of randomization in adaptive large neighborhood search. *International Transactions in Operational Research*, 24(5):929–942, 2017.

[48] Vera C Hemmelmayr, Jean-François Cordeau, and Teodor Gabriel Crainic. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & Operations Research*, 39(12):3215–3228, 2012.

[49] Gregor Hendel. Adaptive large neighborhood search for mixed integer programming. *Zuse Institute Berlin Report 18-60*, 2018.

[50] Julian PT Higgins and Sally Green. *Cochrane handbook for systematic reviews of interventions*, volume 4. John Wiley & Sons, 2011.

[51] Julian Hof and Michael Schneider. An adaptive large neighborhood search with path relinking for a class of vehicle-routing problems with simultaneous pickup and delivery. *Networks*, DOI:10.1002/net.21879.

[52] Andreas Hottenrott. An adaptive large neighborhood search algorithm for the tail assignment problem of airlines. Master's thesis, RWTH Aachen University, 2015.

[53] Çağatay Iris and Jasmine Siu Lee Lam. Recoverable robustness in weekly berth and quay crane planning. *Transportation Research Part B: Methodological*, 122:365–389, 2019.

[54] Çağatay Iris, Dario Pacino, and Stefan Røpke. Improved formulations and an adaptive large neighborhood search heuristic for the integrated berth allocation and quay crane assignment problem. *Transportation Research Part E: Logistics and Transportation Review*, 105:123–147, 2017.

[55] D.S. Johnson. A theoretician's guide to the experimental analysis of algorithms. *Data structures, near neighbor searches, and methodology: fifth and sixth DIMACS implementation challenges*, 59:215–250, 2002.

[56] Laila Kakar and Dennis Huisman. Implementation of an adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. Master's thesis, Erasmus School of Economics, 2014.

[57] Surendra Reddy Kancharla and Gitakrishnan Ramadurai. An adaptive large neighborhood search approach for electric vehicle routing with load-dependent energy consumption. *Transportation in Developing Economies*, 4(2):10, 2018.

[58] Klemens Katterbauer, Ceyda Oguz, and Sibel Salman. Hybrid adaptive large neighborhood search for the optimal statistic median problem. *Computers & Operations Research*, 39(11):2679–2687, 2012.

[59] Merve Keskin and Bülent Çatay. Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 65:111–127, 2016.

[60] Alexander Kiefer, Richard F Hartl, and Alexander Schnell. Adaptive large neighborhood search for the curriculum-based course timetabling problem. *Annals of Operations Research*, 252(2):255–282, 2017.

[61] Gabriel Kleinoscheg, Harald Burgsteiner, Martin Bernroider, Maria Obermayer, et al. Improving emergency medical dispatching by means of the adaptive large neighborhood search. In *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 293–296. IEEE, 2014.

[62] Çağrı Koç. A unified-adaptive large neighborhood search metaheuristic for periodic location-routing problems. *Transportation Research Part C: Emerging Technologies*, 68:265–284, 2016.

[63] Attila A Kovacs, Sophie N Parragh, Karl F Doerner, and Richard F Hartl. Adaptive large neighborhood search for service technician routing and scheduling problems. *Journal of Scheduling*, 15(5):579–600, 2012.

[64] Attila A Kovacs, Sophie N Parragh, and Richard F Hartl. A template-based adaptive large neighborhood search for the consistent vehicle routing problem. *Networks*, 63 (1):60–81, 2014.

[65] Simon Kristiansen and Thomas R Stidsen. Adaptive large neighborhood search for student sectioning at Danish high schools. In *Proceedings of the Ninth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)*, 2012.

[66] Gilbert Laporte, Roberto Musmanno, and Francesca Vocaturo. An adaptive large neighbourhood search heuristic for the capacitated arc-routing problem with stochastic demands. *Transportation Science*, 44(1):125–135, 2010.

[67] Joseph Lau, John PA Ioannidis, Norma Terrin, Christopher H Schmid, and Ingram Olkin. The case of the misleading funnel plot. *Bmj*, 333(7568):597–600, 2006.

[68] Fabien Lehuédé, Renaud Masson, Sophie N Parragh, Olivier Péton, and Fabien Tricoire. A multi-criteria large neighbourhood search for the transportation of disabled people. *Journal of the Operational Research Society*, 65(7):983–1000, 2014.

[69] Baoxiang Li, Dmitry Krushinsky, Tom Van Woensel, and Hajo A Reijers. An adaptive large neighborhood search heuristic for the share-a-ride problem. *Computers & Operations Research*, 66:170–180, 2016.

[70] Yuan Li, Haoxun Chen, and Christian Prins. Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests. *European Journal of Operational Research*, 252(1):27–38, 2016.

[71] Ran Liu, Yangyi Tao, and Xiaolei Xie. An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and synchronized visits. *Computers & Operations Research*, 101:250–262, 2019.

[72] Xiaolu Liu, Gilbert Laporte, Yingwu Chen, and Renjie He. An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time. *Computers & Operations Research*, 86:41–53, 2017.

[73] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, and Thomas Stützle. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58, 2016.

[74] Zhixing Luo, Hu Qin, Dezhi Zhang, and Andrew Lim. Adaptive large neighborhood search heuristics for the vehicle routing problem with stochastic demands and weight-related cost. *Transportation Research Part E: Logistics and Transportation Review*, 85:69–89, 2016.

[75] Richard Martin Lusby, Martin Schwierz, Troels Martin Range, and Jesper Larsen. An adaptive large neighborhood search procedure applied to the dynamic patient admission scheduling problem. *Artificial Intelligence in Medicine*, 74:21–31, 2016.

[76] Roman Lutz. Adaptive large neighborhood search. Bachelor thesis, Ulm University, 2015.

[77] Jean-Baptiste Mairy, Yves Deville, and Pascal Van Hentenryck. Reinforced adaptive large neighborhood search. In *The Seventeenth International Conference on Principles and Practice of Constraint Programming (CP 2011)*, page 55, 2011.

[78] Setareh Majidi, Seyyed-Mahdi Hosseini-Motlagh, and Joshua Ignatius. Adaptive large neighborhood search heuristic for pollution-routing problem with simultaneous pickup and delivery. *Soft Computing*, 22(9):2851–2865, 2018.

[79] Simona Mancini. A real-life multi depot multi period vehicle routing problem with a heterogeneous fleet: Formulation and adaptive large neighborhood search based matheuristic. *Transportation Research Part C: Emerging Technologies*, 70:100–112, 2016.

[80] Renaud Masson, Fabien Lehuédé, and Olivier Péton. An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science*, 47(3):344–355, 2013.

[81] Geraldo Regis Mauri, Glaydston Mattos Ribeiro, Luiz Antonio Nogueira Lorena, and Gilbert Laporte. An adaptive large neighborhood search for the discrete and continuous berth allocation problem. *Computers & Operations Research*, 70:140–154, 2016.

[82] Elham Mofid-Nakhaee and Farnaz Barzinpour. A multi-compartment capacitated arc routing problem with intermediate facilities for solid waste collection using hybrid adaptive large neighborhood search and whale algorithm. *Waste Management & Research*, 37(1):38–47, 2019.

[83] David Moher, Deborah J Cook, Susan Eastwood, Ingram Olkin, Drummond Rennie, and Donna F Stroup. Improving the quality of reports of meta-analyses of randomised controlled trials: the QUOROM statement. *Oncology Research and Treatment*, 23(6):597–602, 2000.

[84] David Moher, Alessandro Liberati, Jennifer Tetzlaff, and Douglas G Altman. Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. *Annals of Internal Medicine*, 151(4):264–269, 2009.

[85] Marcela Monroy-Licht, Ciro Alberto Amaya, and André Langevin. Adaptive large neighborhood search algorithm for the rural postman problem with time windows. *Networks*, 70(1):44–59, 2017.

[86] Adam Montjoy and Jeffrey W Herrmann. Adaptive large neighborhood search for the inventory slack routing problem. In *IIE Annual Conference. Proceedings*, page 1. Institute of Industrial and Systems Engineers (IISE), 2010.

[87] Laurent Flindt Muller. An adaptive large neighborhood search algorithm for the resource-constrained project scheduling problem. In *MIC 2009: The VIII Metaheuristics international conference*, 2009.

[88] Laurent Flindt Muller. An adaptive large neighborhood search algorithm for the multi-mode RCPSP. *DTU Management 2011, No. 3*, 2011.

[89] Laurent Flindt Muller and Simon Spoorendonk. A hybrid adaptive large neighborhood search algorithm applied to a lot-sizing problem. *DTU Management 2010, No. 1*, 2010.

[90] Salma Naccache, Jean-François Côté, and Leandro Callegari Coelho. *An Adaptive Large Neighborhood Search for the Multi-Pickup and Delivery Problem with Time Windows*. CIRRELT, 2017.

[91] Quang Ngoc Nguyen, Nghia Nguyen Duc, and Phuong Khanh Nguyen. An adaptive large neighborhood search for multi-trip multi-traffic pickup and delivery problem with time windows and synchronization. In *Proceedings of the Eighth International Symposium on Information and Communication Technology*, pages 439–446. ACM, 2017.

[92] Pamela J Palomo-Martínez, M Angélica Salazar-Aguilar, and Gilbert Laporte. Planning a selective delivery schedule through adaptive large neighborhood search. *Computers & Industrial Engineering*, 112:368–378, 2017.

[93] Sophie N Parragh and Jean-François Cordeau. Branch-and-price and adaptive large neighborhood search for the truck and trailer routing problem with time windows. *Computers & Operations Research*, 83:28–44, 2017.

[94] Sophie N Parragh and Verena Schmid. Hybrid column generation and large neighborhood search for the dial-a-ride problem. *Computers & Operations Research*, 40(1): 490–497, 2013.

[95] Paola Pellegrini, Thomas Stützle, and Mauro Birattari. A critical analysis of parameter adaptation in ant colony optimization. *Swarm Intelligence*, 6(1):23–48, 2012.

[96] Paola Pellegrini, Franco Mascia, Thomas Stützle, and Mauro Birattari. On the sensitivity of reactive tabu search to its meta-parameters. *Soft Computing*, 18(11):2177–2190, 2014.

[97] David Pisinger and Stefan Røpke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007.

[98] Rapeepan Pitakaso and Kanchana Sethanan. Adaptive large neighborhood search for scheduling sugarcane inbound logistics equipment and machinery under a sharing infield resource system. *Computers and Electronics in Agriculture*, 158:313–325, 2019.

[99] Robin L Plackett. Studies in the history of probability and statistics: Vii. the principle of the arithmetic mean. *Biometrika*, 45(1-2):130–135, 1958.

[100] Yuan Qu and Jonathan F Bard. A grasp with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers & Operations Research*, 39(10):2439–2456, 2012.

[101] Glaydston Mattos Ribeiro and Gilbert Laporte. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 39(3):728–735, 2012.

[102] Achmad P Rifai, Huu-Tho Nguyen, and Siti Zawiah Md Dawal. Multi-objective adaptive large neighborhood search for distributed reentrant permutation flow shop scheduling. *Applied Soft Computing*, 40:42–57, 2016.

[103] Juan-Pablo Riquelme-Rodríguez, André Langevin, and Michel Gamache. Adaptive large neighborhood search for the periodic capacitated arc routing problem with inventory constraints. *Networks*, 64(2):125–139, 2014.

[104] Stefan Røpke and David Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40 (4):455–472, 2006.

[105] David Sacramento, David Pisinger, and Stefan Røpke. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies*, 102:289–315, 2019.

[106] M Angélica Salazar-Aguilar, André Langevin, and Gilbert Laporte. An adaptive large neighborhood search heuristic for a snow plowing problem with synchronized routes. In *International Conference on Network Optimization*, pages 406–411. Springer, 2011.

[107] Alberto Santini. An adaptive large neighbourhood search algorithm for the orienteering problem. *Expert Systems with Applications*, 123:154–167, 2019.

[108] Alberto Santini, Stefan Røpke, and Lars Magnus Hvattum. A comparison of acceptance criteria for the adaptive large neighbourhood search metaheuristic. *Journal of Heuristics*, 24(5):783–815, 2018.

[109] Vinicius Gandra Martins Santos and Marco Antonio Moreira de Carvalho. Adaptive large neighborhood search applied to the design of electronic circuits. *Applied Soft Computing*, 73:14–23, 2018.

[110] Vinícius Gandra Martins Santos and Marco Antonio Moreira de Carvalho. Tailored heuristics in adaptive large neighborhood search applied to the cutwidth minimization problem. *European Journal of Operational Research*, 2019.

[111] Maximilian Schiffer and Grit Walther. An adaptive large neighborhood search for the location-routing problem with intra-route facilities. *Transportation Science*, 52 (2):331–352, 2017.

[112] Johannes Scholz and Norbert Bartelme. Spatial-enabled adaptive large neighborhood search for optimizing the wood supply chain. In *Proceedings of the Sixth international conference on Geographic Information Science*, 2010.

[113] Kristian Schopka and Herbert Kopfer. An adaptive large neighborhood search for the reverse open vehicle routing problem with time windows. In *Logistics Management*, pages 243–257. Springer, 2016.

[114] Gerhard Schrimpf, Johannes Schneider, Hermann Stamm-Wilbrandt, and Gunter Dueck. Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, 159(2):139–171, 2000.

[115] Paul Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *International conference on principles and practice of constraint programming*, pages 417–431. Springer, 1998.

[116] Viacheslav A Shirokikh and Victor V Zakharov. Dynamic adaptive large neighborhood search for inventory routing problem. In *Modelling, Computation and Optimization in Information Systems and Management Sciences*, pages 231–241. Springer, 2015.

[117] L.M. Simons. Adaptive large neighborhood search for rich and real-world vehicle routing problems. Master's thesis, Delft University of Technology, 2017.

[118] RJS Simpson and Karl Pearson. Report on certain enteric fever inoculation statistics. *The British Medical Journal*, pages 1243–1246, 1904.

[119] Matias Sørensen, Simon Kristiansen, and Thomas Riis Stidsen. International timetabling competition 2011: An adaptive large neighborhood search algorithm. In *9th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2012)*, 2012.

[120] Jonathan AC Sterne and Matthias Egger. Funnel plots for detecting bias in meta-analysis: guidelines on choice of axis. *Journal of clinical epidemiology*, 54(10):1046–1055, 2001.

[121] Donna F Stroup, Jesse A Berlin, Sally C Morton, Ingram Olkin, G David Williamson, Drummond Rennie, David Moher, Betsy J Becker, Theresa Ann Sipe, Stephen B Thacker, et al. Meta-analysis of observational studies in epidemiology: a proposal for reporting. *Jama*, 283(15):2008–2012, 2000.

[122] Doan Thanh Tan, Nathalie Bostel, and Ha Minh Hoang. An adaptive large neighborhood search for the hierarchical vehicle routing problem.

[123] Oscar Tellez, Samuel Vercraene, Fabien Lehuédé, Olivier Péton, and Thibaud Monteiro. The fleet size and mix dial-a-ride problem with reconfigurable vehicle capacity. *Transportation Research Part C: Emerging Technologies*, 91:99–123, 2018.

[124] Charles Thomas and Pierre Schaus. Revisiting the self-adaptive large neighborhood search. In *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 557–566. Springer, 2018.

[125] Hamza Ben Ticha, Nabil Absi, Dominique Feillet, and Alain Quilliot. Multigraph modeling and adaptive large neighborhood search for the vehicle routing problem with time windows. *Computers & Operations Research*, 104:113–126, 2019.

[126] Liana van der Hagen, TR Visser, and R Spliet. The pickup and delivery problem with time windows: an adaptive large neighborhood search heuristic. 2017.

[127] Nicholas Alexander Vergeylen. *A novel approach to city bicycle repositioning*. PhD thesis, University of Antwerp, 2018.

[128] Esteban Walker, Adrian V Hernandez, and Michael W Kattan. Meta-analysis: Its strengths and limitations. *Cleveland Clinic Journal of Medicine*, 75(6):431, 2008.

[129] Pei Wang, Gerhard Reinelt, and Yuejin Tan. Self-adaptive large neighborhood search algorithm for parallel machine scheduling problems. *Journal of Systems Engineering and Electronics*, 23(2):208–215, 2012.

[130] Min Wen, Esben Linde, Stefan Røpke, P Mirchandani, and Allan Larsen. An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem. *Computers & Operations Research*, 76:73–83, 2016.

[131] Yan Wu, Wang Yang, Guochao He, and Shennan Zhao. An improved adaptive large neighborhood search algorithm for the heterogeneous fixed fleet vehicle routing problem. In *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 657–663. IEEE, 2017.

[132] Zijie Yan and Adam Montjoy. Variants of the adaptive large neighborhood search for the inventory slack routing problem. Technical report, 2011.

[133] Chuhang Yu, Dong Zhang, and Henry YK Lau. An adaptive large neighborhood search heuristic for solving a robust gate assignment problem. *Expert Systems with Applications*, 84:143–154, 2017.