

# Experimental Analysis of Pheromone-based Heuristic Column Generation using irace

Florence Massen<sup>1</sup>, Manuel López-Ibáñez<sup>2</sup>, Thomas Stützle<sup>2</sup>, and Yves Deville<sup>1</sup>

<sup>1</sup> ICTEAM, Université catholique de Louvain, Belgium  
{Florence.Massen, Yves.Deville}@uclouvain.be  
<sup>2</sup> IRIDIA, Université libre de Bruxelles, Belgium  
{manuel.lopez-ibanez, stuetzle}@ulb.ac.be

**Abstract.** Pheromone-based heuristic column generation (ACO-HCG) is a hybrid algorithm that combines ant colony optimization and a MIP solver to tackle vehicle routing problems (VRP) with black-box feasibility. Traditionally, the experimental analysis of such a complex algorithm has been carried out manually by trial and error. Moreover, a full-factorial statistical analysis is infeasible due to the large number of parameters and the time required for each algorithm run. In this paper, we first automatically configure the algorithm parameters by using an automatic algorithm configuration tool. Then, we perform a basic sensitivity analysis of the tuned configuration in order to understand the significance of each parameter setting. In this way, we avoid wasting effort analyzing parameter settings that do not lead to a high-performing algorithm. Finally, we show that the tuned parameter settings improve the performance of ACO-HCG on the multiple VRP and the three-dimensional loading capacitated VRP.

## 1 Introduction

An advantage of metaheuristics is that they can be adapted to different problem variants by adjusting their algorithmic components and parameter settings. The parameter configuration often has a crucial impact on the performance of an algorithm on a particular problem. However, complex algorithms typically have many parameters, resulting in a large number of possible configurations of the algorithm. Testing all possible configurations is typically intractable, particularly on problems where one run of the algorithm may take several hours. Due to this intractability, only a very limited subset of parameter configurations are tested when designing an algorithm, and such a subset is chosen by the designer based on her intuitions. This manual approach may easily miss the best-performing configurations for a metaheuristic.

Automatic algorithm configuration methods aim to identify high-performing configurations of an algorithm for a given problem [2, 8]. Typically, the method is given a description of the parametric space of the algorithm (types and domains of the parameters), a set of training instances representative of the problem, and a computational budget (e.g., a maximum number of algorithm runs). The goal of the method is to find a high-performing parameter configuration for unseen instances of the same problem. When the goal is not simply to obtain the best performance, but also to understand

the effect of parameters, the number of possible configurations that need to be evaluated and analyzed may be extremely large. This is the case even if most configurations are of little interest, since they produce poor results. In this paper, we propose to use automatic configuration as a first step before analyzing the effect of parameters.

We use this approach to improve the performance and analyze the parameters of pheromone-based heuristic column generation (ACO-HCG) for vehicle routing problems with black box feasibility (VRPBB) [11]. VRPBBs are an abstraction of rich vehicle routing problems. In the VRPBB, the problem structure remains unchanged with respect to a basic VRP, but a combinatorial side-problem needs to be solved to verify the feasibility of each route. Examples of such problems combine routing and loading [7] or routing and scheduling [12]. In the VRPBB, the combinatorial side-problem to be solved for every route is considered to be unknown, but a black-box function allowing to test the feasibility of a route is provided. An optimization procedure for the VRPBB should be independent of the particular side-problem. An example of such an optimization procedure is ACO-HCG, which combines ant colony optimization with an exact solver. At each iteration, ants generate a set of feasible routes probabilistically according to pheromone values. A solution is obtained by solving a relaxed set partitioning problem ( $RMP_{\text{relax}}$ ) over the set of feasible routes. This solution is used to update the pheromone information, biasing the construction of new feasible routes by the ants in subsequent iterations. In addition, feasible routes are further improved using local search. This procedure continues until a time limit is reached, and the final solution is obtained by solving the integer set partitioning problem over the set of all feasible routes ever generated.

In this paper, we extend ACO-HCG with new parameters formalizing algorithmic design choices. Then, we use *irace* [10] to automatically find a high-performing configuration that improves the results of the original ACO-HCG. In particular, we improve previous results of ACO-HCG in two VRPBB problems, the multi-pile VRP (MPVRP) and the three-dimensional loading capacitated VRP (3L-CVRP). The new ACO-HCG also improves the best-known results from the literature on the MPVRP and the 3L-CVRP on several instances. Finally, we perform a basic sensitivity analysis of the parameters of ACO-HCG with the improved configuration generated by *irace*. This analysis shows that the new components play a significant role in the effectiveness of the algorithm.

## 2 Vehicle Routing Problems with Black Box Feasibility

In this section, we present the capacitated vehicle routing problem (CVRP), extend it to VRPs with black-box feasibility and present two example applications of the resulting VRPBB.

### 2.1 The Capacitated Vehicle Routing Problem and Black-Box Feasibility

The CVRP [17] is defined on a complete undirected and weighted graph  $G = (V, E)$ . In the set of vertexes  $V = \{0, 1, \dots, n\}$ , 0 is the depot vertex and vertexes  $1, \dots, n$

correspond to the  $n$  customers that must be visited. The set of edges connecting every pair of vertexes in  $V$  is given in  $E$ . With each edge  $(i, j) \in E$  s.t.  $i, j \in V, i \neq j$  is associated a non-negative weight  $c_{ij}$  corresponding to the cost of traveling through edge  $(i, j)$ , i.e., from vertex  $i$  to  $j$ . A homogeneous fleet of  $K$  vehicles is given for visiting the customers. All vehicles have a limited capacity  $Q$ . Each customer  $i$  ( $i = 1, \dots, n$ ) has a given demand  $d_i$ . A solution to the CVRP corresponds to a set of routes. Each route is a sequence of vertexes, where the first and last vertex always correspond to the depot, while the remaining vertexes correspond to customers. No customer vertex may appear more than once in a route. Finally, the goal is to find a set of routes  $Sol$  such that (i) the number of routes in  $Sol$  does not exceed  $K$ ; (ii) each customer  $i \in V \setminus \{0\}$  is visited exactly once; (iii) the sum of the demands of the customers visited on a route does not exceed the vehicle capacity  $Q$ ; (iv) the total traveling cost obtained by summing the weights of the used edges is minimized.

The VRPBB is based on the CVRP and a feasible solution to the VRPBB must also be feasible for the CVRP. However, in addition to the CVRP constraints, routes in the VRPBB must satisfy a fixed set of unknown constraints  $F$ . A route  $r$  is feasible with respect to  $F$  if and only if it satisfies all the constraints in  $F$ . We suppose that a deterministic black-box function is given to verify the feasibility of a route  $r$  w.r.t.  $F$ . The function is considered to be computationally expensive in comparison to common VRP-feasibility functions. In the following, we call a route black-box (BB)-feasible if it is feasible w.r.t.  $F$ , VRP-feasible if it respects the CVRP constraints, and feasible if it is VRP- and BB-feasible.

## 2.2 Applications of the VRPBB

In this paper, we consider two applications of the VRPBB, the MPVRP and the 3L-CVRP, which combine a basic VRP with two different types of loading constraints. We refer to [9] for an overview of such problems.

**The Multi-Pile VRP.** The MPVRP [4] is based on the CVRP, but there are no restrictions on the capacity nor on the number of vehicles ( $Q = \infty, K = n$ ). However the loading space of the vehicles has a limited length  $L$ , width  $W$  and height  $H$ . Furthermore, the loading space is partitioned into  $p$  piles. Each customer  $i$  demands a set of  $m_i$  items  $I_i$ . Each item  $I_{ik}$  ( $i = 1, \dots, n; k = 1, \dots, m_i$ ) has a fixed width  $W$ , can take two possible lengths  $l_{ik} \in \{L/p, L\}$  and has a height  $h_{ik}$ . A solution to the MPVRP must respect the CVRP constraints and a feasible loading for the items of the visited customers must exist. That is, no items must overlap (*non-overlapping*) and the items must fit in the loading space (*containment*). When a customer is visited, all its items must be directly accessible, i.e., its items are on top of the piles in the back of the truck (*sequential loading*). Existing works on the MPVRP use metaheuristic and exact approaches. Doerner et al. [4] propose heuristic approaches based on the fact that the loading problem to be solved is a generalization of the  $P||C_{max}$  scheduling problem. Tricoire et al. [18] propose an exact approach to solve the loading problem, and a variable neighborhood search as well as a branch-and-cut method for the routing problem.

**The Three-dimensional Loading VRP.** In the 3L-CVRP [7], vehicles in a homogeneous fleet have a limited capacity  $Q$ . Additionally, the loading space of the vehicles has a limited length  $L$ , a limited width  $W$  and a limited height  $H$ . With each customer  $i$  is associated a demand  $d_i$  and a set  $I_i$  of  $m_i$  items. An item  $I_{ik}$  ( $i = 1, \dots, n; k = 1, \dots, m_i$ ) corresponds to a three-dimensional box that is either fragile or non-fragile and has width  $w_{ik}$ , height  $h_{ik}$  and length  $l_{ik}$ . A solution to the 3L-CVRP is feasible if it respects the CVRP constraints and a feasible loading for the items of customers on a route exists. In order to be feasible, no items in a loading may overlap (*non-overlapping*) and all items need to fit into the volume of the vehicle (*containment*). Also, each box must be supported by a surface corresponding to at least 75% of the box's bottom surface (*support*). Moreover, only fragile items may be placed on top of fragile items (*fragility*). Finally, when visiting a customer, the unloading of the items of this customer must not be hindered by items belonging to customers yet to be visited (*LIFO policy*). Most of the approaches proposed for the 3L-CVRP are metaheuristics [7, 16, 6, 3, 22, 15], except for an exact approach proposed recently [14]. In these approaches, the loading problem is typically solved using various packing heuristics.

### 3 Pheromone-based Heuristic Column Generation for the VRPBB

The pheromone-based heuristic column generation algorithm (ACO-HCG) analyzed in this paper was originally proposed by Massen et al. [11]. It is based on the idea of reformulating the VRPBB as a set partitioning problem (SPP). In this section, we briefly explain this reformulation, summarize the ACO-HCG algorithm, and propose several extensions.

#### 3.1 Reformulation as Set Partitioning Problem

The VRPBB may be reformulated as a SPP, where the goal is to choose from the set of all feasible routes ( $\mathcal{R}$ ) a subset of at most  $K$  routes such that each customer appears exactly once and such that the total traveling cost is minimized. However, generating the set of all feasible routes  $\mathcal{R}$  is, in general, intractable due to the exponential number of possible routes and the complexity of the black-box feasibility check. Therefore, a restricted version of the problem (*RMP*) is solved using a restricted set of feasible routes  $\mathcal{R}^*$ . This problem can be further relaxed, resulting in two variants: the linear relaxation of the *RMP* (called  $RMP_{\text{relax}}$ ) or the integer *RMP* (called  $RMP_{\text{int}}$ ). In both, customers may be visited more than once; thus,  $RMP_{\text{int}}$  is also a reduced form of the SPP to be solved. In ACO-HCG, collector ants generate new feasible routes that are iteratively added to  $\mathcal{R}^*$ .

#### 3.2 Pheromone-based Heuristic Column Generation (ACO-HCG)

The ACO-HCG algorithm iterates over three steps: (i) collector ants generate new feasible routes according to the pheromone information  $\tau$ ; (ii) the new routes are added to  $\mathcal{R}^*$  and the corresponding  $RMP_{\text{relax}}$  is solved, obtaining a solution *Sol*; and (iii) *Sol* is used to update the pheromone information  $\tau$ . After a time limit has been reached, the

SPP is solved over the set of feasible routes collected ( $\mathcal{R}^*$ ) to produce the final solution to the VRPBB.

**Collector Ants.** Collector ants are based on the savings-based ants algorithm [13]. Each ant iteratively constructs a set of feasible routes, starting from an initial state where each customer  $i$  ( $i = 1, \dots, n$ ) is visited in a route of its own ( $0-i-0$ ). At each step, the ant constructs a set  $\Omega$  of potential route merges, selects a merge from  $\Omega$  and executes the merge. A merge corresponds to the concatenation of two routes  $0-i_1-i_2-\dots-i_e-0$  and  $0-j_1-j_2-\dots-j_l-0$  producing route  $0-i_1-i_2-\dots-i_e-j_1-j_2-\dots-j_l-0$ . That is, edges  $(i_e, 0)$  and  $(0, j_1)$  are dropped and replaced by a new edge  $(i_e, j_1)$ . The gain in cost resulting from this merge is computed as  $\eta_{i_e j_1} = c_{i_e 0} + c_{0 j_1} - c_{i_e j_1}$ .

The set  $\Omega$  of merges considered by an ant is a subset of the set  $M$  of all merges resulting in VRP-feasible routes (VRP-feasible merges). The ant computes the attractiveness of each merge  $h \in M$  as

$$\text{attractiveness}(h) = \tau_{ij}^\alpha + \eta_{ij}^\beta \quad (1)$$

where  $(i, j)$  is the edge being introduced in merge  $h$ . The merges in  $M$  are then ordered by non-increasing attractiveness and considered one by one for inclusion in  $\Omega$ . Since the BB-feasibility check is computationally expensive the feasibility information for every checked route is stored in a feasibility pool. If an ant encounters a merge resulting in a route that might help to reduce the total traveling cost (see the original paper [11] for more information) or it is unknown to the feasibility pool, then the ant checks the BB-feasibility (using either the feasibility pool or the black-box function), and the merge is added to  $\Omega$  only if the route is BB-feasible. If the merge will not reduce the total traveling cost but the resulting route is known to be BB-infeasible, then the merge is nonetheless added to  $\Omega$ , since this merge may allow discovering further feasible merges that would remain undiscovered otherwise. The construction of  $\Omega$  stops once  $\pi$  feasible merges have been included, or all merges in  $M$  have been considered. Finally, a merge is selected from  $\Omega$  and executed using roulette-wheel selection based on the attractiveness values. The ant stops its process once no further VRP-feasible merge is possible.

During the construction of  $\Omega$ , the ants "collect" routes and add them to  $\mathcal{R}^*$  only if they are BB-feasible. In addition, these routes are post-optimized before being inserted into  $\mathcal{R}^*$  using a tabu search (TS) with infinite length tabu list using 2-opt and relocation moves.

**Pheromone Update.** After all ants finish collecting routes for  $\mathcal{R}^*$ ,  $RMP_{\text{relax}}$  is solved, and the pheromones are updated using the resulting solution  $Sol$  for every edge  $(i, j) \in E$  using the formula  $\tau_{ij} = \rho\tau_{ij} + \sigma_{ij}\Delta\tau$ . That is, for each edge  $(i, j) \in E$ , the current quantity of pheromones is evaporated ( $\rho$  is the trail persistence) and new pheromones are deposited. This quantity is relative to  $\sigma_{ij}$ , the number of times an edge  $(i, j)$  appears in  $Sol$  and a parameter  $\Delta\tau$ . Moreover,  $\tau_{ij}$  is not allowed to drop below  $\tau_{\min}$ .

**Extensions of ACO-HCG.** We propose several extensions that were not considered in the original paper. First, the original algorithm solves  $RMP_{\text{relax}}$  at each iteration.

**Table 1.** Parameters considered for automatic configuration

Parameter	Domain	Description
$\pi$	$[10, 50] \in \mathbb{N}$	# (VRP and BB)-feasible merges in $\Omega$
$m$	$[1, 10] \in \mathbb{N}$	# ants executed per iteration
$\alpha$	$[0, 20] \in \mathbb{N}$	exp. factor for $\tau$ in merge attractiveness (Eq. 1)
$\beta$	$[0, 20] \in \mathbb{N}$	exp. factor for $\eta$ in merge attractiveness (Eq. 1)
$\Delta\tau$	$[0.0, 1.0] \in \mathbb{R}$	pheromone update constant
$\rho$	$[0, 1] \in \mathbb{R}$	trail persistence
$\tau_{\min}$	$[0, 1] \in \mathbb{R}$	lower bound on pheromone level
$useint$	$\{never, always, \nu\}$	solve $RMP_{\text{int}}$ instead of $RMP_{\text{relax}}$
$\nu$	$[2, 10] \in \mathbb{N}$	if $useint == \nu$ , solve $RMP_{\text{int}}$ instead of $RMP_{\text{relax}}$ every $\nu$ iterations
$strictness$	$\{strict, liberal\}$	strict / liberal ants
$post-opt$	$\{ILS, TS\}$	ILS / TS for post-optimization
$op$	$\{+, \cdot\}$	use addition/multiplication operator in Eq. 1

Here, we consider solving  $RMP_{\text{int}}$  instead, in order to obtain a non-fractional solution at the expense of more computation time. This component is controlled by parameter  $useint = \{never, always, \nu\}$ . When  $useint = \nu$ ,  $RMP_{\text{int}}$  is only used for some iterations, concretely, every  $\nu$  iterations. Second, we implement iterated local search (ILS) using 2-opt, relocation and 4-opt double-bridge moves, as an alternative post-optimization method to the tabu search (TS) proposed in the original ACO-HCG. Third, we consider a stricter variant of the ants (“strict ants”) that only include merges in  $\Omega$  if they are both VRP- and BB-feasible. As explained above, the original ants, which we call “liberal”, included BB-infeasible merges in  $\Omega$  in some circumstances. Fourth, when computing the attractiveness (Eq. 1), the original algorithm sums the pheromone and heuristic values, which is fast to compute but if the ranges of the two values are very different then one will completely dominate the other. Here, we propose to use the product  $attractiveness(h) = \tau_{ij}^\alpha \cdot \eta_{ij}^\beta$ , which is potentially slower, but it is more robust if  $\tau^\alpha$  and  $\eta^\beta$  have different ranges. These two alternatives are controlled by parameter  $op$ . A summary of all the algorithmic parameters and their domains is given in Table 1.

## 4 Experimental Setup

We analyze the components of ACO-HCG in a novel way. First, we find a high-performing parameter configuration by means of `irace`, an automatic algorithm configuration tool. Second, we examine the effect of each parameter starting from this high-performing configuration.

ACO-HCG was implemented in C++, compiled using gcc 4.4.6 and uses CPLEX 12.4 as the MILP solver. The black box for the MPVRP was provided by Tricoire et al. [18]. We use their exact approach with a time limit of 5 seconds. The black box used for the 3L-CVRP is a reimplementation of the loading approach proposed by Bort-

feldt [3] using the same parameters. Experiments were run on a single core of an AMD Opteron 6272 CPU (2.1 GHz, 16 MB L2/L3 cache size) running under Cluster Rocks Linux version 6/CentOS 6.3, 64bits.

As for the automatic algorithm configuration tool, we use *irace* [10], a publicly available implementation of Iterated F-Race [1]. Iterated F-Race starts by sampling a number of parameter configurations of a given algorithm uniformly at random. Then, at each iteration, it selects a set of elite configurations using a racing procedure and the non-parametric Friedman test. This racing procedure runs the algorithm configurations iteratively on a sequence of (training) problem instances, and discards configurations as soon as there is enough statistical evidence that they perform worse than the best one. After the race, the elite configurations are used to bias a local sampling model. The next iteration starts by sampling new configurations from this model, and racing is applied to these configurations together with the previous elite configurations. This procedure is repeated until a given budget of runs is exhausted. In this work, we tuned ACO-HCG using a budget of 5 000 runs.

Ideally, an automatic configuration method should produce an algorithm configuration that performs well on unseen instances of the same problem. In other words, the method should generalize over the given set of instances, and not overtune the algorithm to those specific instances. To prevent such overtuning, we use *training* instances for the tuning process that are different from the *test* instances used for comparison and in the analysis of parameters. The *test* instances are benchmark instances from the literature and available online: see [20] for MPVRP, and [19] for 3L-CVRP.

The *training* instances were generated by perturbing the benchmark instances from the literature. Only slight perturbations of some customer properties were allowed in order to not destroy the underlying problem structure. Each customer property was perturbed with a probability of 95%. A perturbation replaces the value of the property by  $current\ value + r \cdot \max_{prop}$ , where  $r$  is a number selected uniformly at random in the interval  $[-\delta, \delta]$  and  $\max_{prop}$  is the maximal value for the considered property in the original instance. Different values for  $\delta$  were considered ( $\delta \in \{0.05, 0.1, 0.15\}$  for MPVRP, and  $\delta \in \{0.1, 0.15\}$  for 3L-CVRP).

For the MPVRP instances only predefined types of items are available [4], and, hence, the demand of a customer corresponds to the number of items demanded per type. The following customer properties were considered for perturbation:  $x$ -coordinate and demand of one randomly selected type of item. For the 3L-CVRP, the following customer properties were considered:  $x$ -coordinate, demand, randomly selected dimension of randomly selected item, fragility of randomly selected item. Five different combinations of these properties were considered.

With each generated instance is associated a time limit (on the route generation phase). This time limit corresponds to the limit associated with the original instance in [18] for the MPVRP (1800 seconds) and [7] (1800, 3600 and 7200 seconds based on the instance size) for the 3L-CVRP.

## 5 Experimental Results

In this section, we first compare the parameter configuration obtained automatically using `irace` with the manual configuration of ACO-HCG. Then, we analyze the parameters of ACO-HCG one by one, starting from the automatically obtained configuration. As mentioned above, the automatic configuration uses a set of training instances, generated by us, and the comparison and analysis uses a different set of test instances from the literature. The performance of the algorithm is measured by computing for each instance the relative percentage deviation (%-deviation) with respect to the best-known solution from the literature. We only consider best-known solutions obtained with the same loading algorithms that we use here as black-box functions ([18, 11] for the MPVRP and [3, 11] for the 3L-CVRP). We call these solutions *best-bb* in the remainder of this paper. The %-deviation is computed as  $100 \cdot \frac{z - z_{\text{best}}}{z_{\text{best}}}$ , where  $z$  is the solution cost obtained by a run of the algorithm and  $z_{\text{best}}$  is the best-bb solution cost for the same instance. We use the Wilcoxon signed-rank test with confidence level 95% to assess the statistical significance of the results.

### 5.1 Manual vs. automatic parameter configurations

We carry out the automatic configuration of ACO-HCG using `irace`, the parameter domains given in Table 1, and the set of training instances. A run of `irace` stops after 5 000 runs of ACO-HCG. We run `irace` two times, once for the MPVRP training instances and another time for the 3L-CVRP training instances. Thus, we obtain two "automatic" configurations of ACO-HCG. Table 2 shows these automatic configurations and the two "manual" configurations that were reported in the original paper [11]. However, two parameters of the original algorithm are different in this manual configuration. First, in the original paper a sum ( $op = \{+\}$ ) was used for computing attractiveness. This has been changed to a multiplication ( $op = \{\cdot\}$ ) in order to be coherent with the standard attractiveness formulation used in the ACO literature [5]. Second, in the original algorithm the parameter  $\pi$  was implicitly defined in terms of two other parameters and the instance size, while here  $\pi$  is a single parameter, which simplifies the analysis. The setting of  $\pi$  in the manual configuration is a close approximation to the value that would be obtained given the default values of the two parameters replaced and typical instance sizes. The manual configurations were based on preliminary experiments and standard ACO parameters, and they were found to be competitive with existing approaches.

Some notable differences between the settings of the automatic configurations and the manual ones are that the former have larger value of  $\beta$ , a larger number of ants ( $m$ ), a lower pheromone persistence ( $\rho$ ), they solve  $RMP_{\text{int}}$  in some iterations ( $\nu$ ), and the ants are *strict* instead of *liberal*. The last two parameter settings are actually extensions of the ACO-HCG that we propose in this paper, and selected by `irace` on its own. In fact, we also propose ILS as an alternative post-optimization method (*post-opt*), but `irace` did not select it. The parametric analysis in the next section indeed indicates that ILS does not bring any improvement over TS in any of the two benchmark problems.

The automatic and manual configurations of ACO-HCG are compared in Fig. 1 in terms of %-deviation from the best-bb cost on the test instances. Each point in the plot shows the mean %-deviation over 20 independent runs (with different random seeds)

**Table 2.** Parameter configurations of ACO-HCG.

<b>Problem</b>	<b>Config.</b>	$\pi$	$m$	$\alpha$	$\beta$	$\Delta\tau$	$\rho$	$\tau_{\min}$	useint ( $\nu$ )	<i>strict.</i>	<i>post-opt</i>	<i>op</i>
MPVRP	Manual	13	1	5	5	0.15	0.95	0.20	never	liberal	TS	mult
	Automatic	10	9	1	10	0.69	0.34	0.79	$\nu = 7$	strict	TS	mult
3L-CVRP	Manual	13	5	5	5	0.15	0.95	0.20	never	liberal	TS	mult
	Automatic	41	10	3	9	0.66	0.45	0.29	$\nu = 6$	strict	TS	mult

on the same test instance. The two configurations perform equally on the same instance if the point is on the diagonal, the automatic configuration performs better if the point is under the diagonal, and the manual configuration performs better if the point is above the diagonal. Moreover, the symbols denote whether the differences observed are statistically significant.

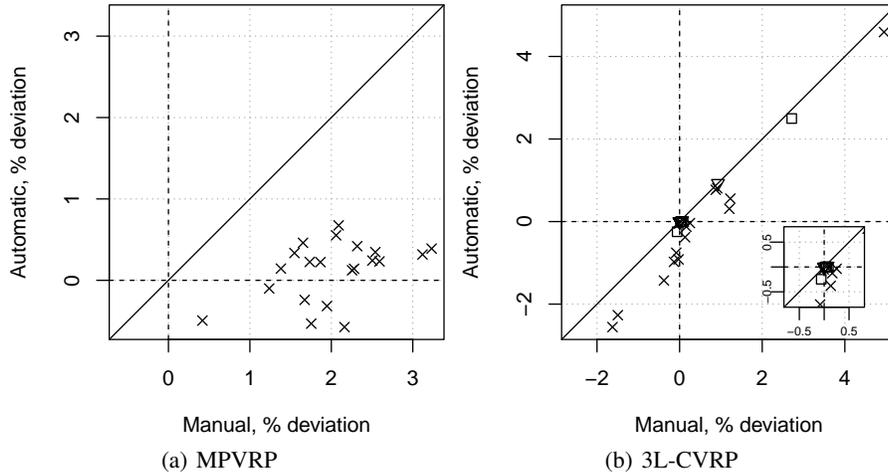
For the MPVRP (Fig. 1(a)), the improvement of the automatic configuration over the manual one is considerable. In particular, all the differences are statistically significant. Moreover, for many instances, the automatic configuration obtains an average result that is better than the best-bb solution. For the 3L-CVRP (Fig. 1(b)), the improvement is smaller. Nonetheless, the automatic configuration is never worse than the manual configuration and it is significantly better on a few instances. In Table 3 the manual and automatic configuration are compared to the best known solutions in literature using all kinds of loading algorithms (and thus possibly obtained using loading algorithms different from the ones used as black box functions in this work). The automatic configuration is able to find new best solutions for 16 out of the 21 MPVRP instances and 3 out of the 27 3L-CVRP instances. Complete tables with worst values and standard deviation are provided as supplementary material [21]

## 5.2 Experimental analysis of the ACO-HCG parameters

In this section, we systematically examine several algorithm parameters. In contrast to how such parametric analyses are carried out in the literature, we adopt a different approach that exploits the benefits of automatic configuration tools. In particular, we do not consider a fully-factorial experimental design, since the number of parameters and the computation time required by each run make such an approach intractable. Instead, we start from the high-performing parameter configuration automatically obtained in the previous section, and examine parameter settings that disable or replace one algorithmic component at a time.

**Pheromone information ( $\alpha$ ).** By setting  $\alpha = 0$ , we disable the influence of the pheromone information. The result is a noticeable deterioration in quality in most MPVRP instances (Fig 2(a)) and some 3L-CVRP instances (Fig 2(e)). This suggests that the pheromone information plays a positive role in the performance of the algorithm, probably helping to diversify the routes produced by the ants.

**Savings heuristic ( $\beta$ ).** By setting  $\beta = 0$ , we disable the use of the savings heuristic  $\eta$  in the attractiveness equation. The savings heuristic guides the ants to build cost-efficient routes, and, hence, disabling it leads to a substantial quality deterioration in



**Fig. 1.** Comparison between manual and automatic configuration. Each point gives the mean %-deviation from the best-bb solution over 20 runs with different random seed on the same test instance. The symbols denote whether there is a statistically significant difference (X) or not ( $\square$ ), or all the runs obtained the same cost ( $\nabla$ ).

both problems (MPVRP, Fig. 2(b), and 3L-CVRP, Fig. 2(f)). For small instances of the 3L-CVRP, however, the differences are typically minor. This is due to the high value for parameter  $\pi$ . In fact, for small instances with only few customers, setting  $\pi$ , that is, the number of feasible merges in  $\Omega$ , to a high value, results in most possible merges being included in  $\Omega$ . These are then checked for feasibility and added to  $\mathcal{R}^*$ . However, for large instances, the setting of parameter  $\pi$  excludes many interesting merges that would have a high heuristic value if  $\beta \neq 0$ . In summary, the savings heuristic remains essential for the generation of high-quality routes.

**Learning mechanism ( $\rho$ ).** By setting  $\rho = 0$ , the pheromones are reset at every iteration, and only the amount deposited in the current iteration has an effect. Hence, this setting disables the learning mechanism of ACO and forces the ants to focus on the solution found in the current iteration. Given that the results do not show a clear effect of the learning mechanism (Fig. 2(d) and 2(h)), but that completely disabling the pheromone information ( $\alpha = 0$ , as discussed above) does deteriorate quality, we conclude that the pheromone information provides a diversification mechanism rather than learning the best edges over time.

**Strict vs. liberal ants.** Whereas *strict* ants may only execute merges resulting in VRP- and BB-feasible routes, *liberal* ants may also execute merges resulting in only VRP-feasible routes. The rationale of *liberal* ants is that BB-infeasible routes might be merged in order to produce BB-feasible routes. This, of course, depends on the black box at hand. In the case of the MPVRP, such a situation cannot arise and, hence, *strict* ants produce much better results (Fig. 2(c)). While for the 3L-CVRP two infeasible

routes can theoretically be concatenated to produce a feasible route, such a situation does not seem to occur frequently in the benchmark instances available (Fig. 2(g)). Thus, the choice of *strict* ants rather than *liberal* ants in the automatic configuration seems to be justified.

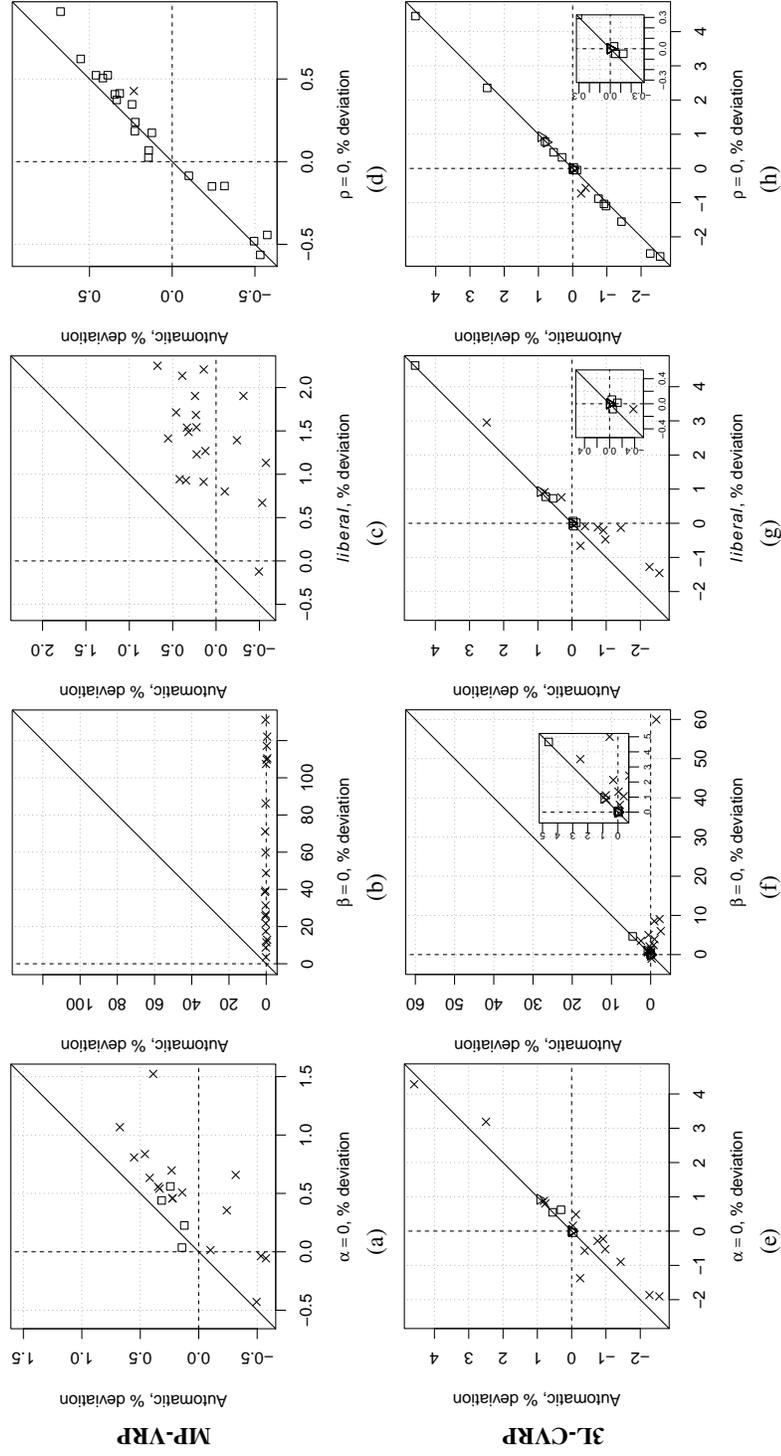
**Sum vs. multiplication in attractiveness equation.** The results clearly worsen when using a sum ( $op = +$ ) in the attractiveness equation. Since the values of the savings heuristic are much larger than the pheromone values, summing both neglects the effect of the pheromones. In fact, the plots (available as supplementary material [21]) are almost identical to those where the pheromone information is disabled ( $\alpha = 0$ , Fig 2(a) and 2(e)). Therefore, the use of multiplication is recommended.

We also analyzed other parameters of ACO-HCG, however, for the sake of conciseness, we only briefly summarize our findings here, and provide the full plots as supplementary material [21]. In particular, we analyzed the parameter *useint*, which controls whether the solution used to update the pheromone information is obtained by solving  $RMP_{int}$  or  $RMP_{relax}$ . For the MPVRP, while never using the integer solution does not have a significant effect on most instances, always using it slightly deteriorates the quality in some instances. For the 3L-CVRP, it does not matter whether we use  $RMP_{relax}$  or  $RMP_{int}$  to update the pheromones. When comparing ILS vs. TS as the post-optimization method, we observe that, in the MPVRP, solution quality does deteriorate in some instances when using ILS instead of TS, and in the 3L-CVRP, a slight deterioration can be observed on a couple of instances. Finally, if the number of ants is set to one ( $m = 1$ ), results improve slightly on a few instances and get slightly worse in others. In the case of the 3L-CVRP, setting  $m = 1$  slightly deteriorates quality in a majority of the instances.

## 6 Conclusion

In this paper, we carried out a parametric analysis of ACO-HCG for the VRPBB in a novel way. As a first step, we obtained a high-performing configuration of ACO-HCG by means of automatic configuration for two variants of the VRPBB, namely, MPVRP and 3L-CVRP. This automatic configuration significantly improves the results obtained by the default configuration of ACO-HCG. The default configuration was developed based on intuition and a few preliminary experiments, and it was found to be competitive with the state of the art. However, due to the long computation times required and the large number of parameters, further improving the parameter configuration by traditional methods was deemed intractable. Automatic configuration tools allowed us to overcome this difficulty and test new algorithmic components. In fact, the new configuration is able to improve the best-known solutions on many instances.

As a second step, we systematically analyzed the parameters of ACO-HCG, starting from the automatically-found configuration, and disabling or replacing one component at a time to observe its effect on quality. In this way, we identified four components that have a large effect on the results, namely, the pheromone information, the savings heuristic, the strictness of the ants when considering feasible routes, and the equation



**Fig. 2.** Each plot shows the effect of changing one parameter of the "automatic" configuration. Results for the MP-VRP are in the upper row; results for 3L-CVRP in the lower row. Each point gives the mean % deviation from the best-bb solution over 20 runs with different random seed on the same test instance. The symbols denote whether there is statistically significant difference (X) or not (square), or all the runs obtained the same cost ( $\nabla$ ).



used for computing the attractiveness. Moreover, although the use of pheromone information is necessary to complement the savings heuristic, we also determined that the learning mechanism of ACO does not have a strong effect. These insights should lead to future improvements in the algorithm.

The analysis methodology used here can easily be applied to other algorithms (using other automatic configuration tools besides *irace*). We expect that the larger improvements and best insights will be obtained when analyzing hybrid algorithms with many parameters that require long runs. In that case, the default configuration is probably far from optimal and previous studies may have missed insights that are only relevant for high-performing parameter configurations.

Future work should consider new algorithmic components that may improve ACO-HCG, other black-box functions found in the literature, and the effect of the parameters on the computation time required by the algorithm. We will also perform a more elaborate sensitivity analysis of the parameters, evaluating the influence of variations in their values.

**Acknowledgments.** This work was supported by the META-X project, an *Action de Recherche Concertée* funded by the Scientific Research Directorate of the French Community of Belgium. Manuel López-Ibáñez and Thomas Stützle acknowledge support from the Belgian F.R.S.-FNRS, of which they are a postdoctoral researcher and a research associate, respectively. The authors also acknowledge support from the FRFC project “*Méthodes de recherche hybrides pour la résolution de problèmes complexes*”. This research and its results have also received funding from the COMEX project within the Interuniversity Attraction Poles Programme of the Belgian Science Policy Office. Florence Massen is supported by the National Research Fund, Luxembourg.

## References

1. Balaprakash, P., Birattari, M., Stützle, T.: Improvement strategies for the F-race algorithm: Sampling design and iterative refinement. In: Bartz-Beielstein, T., et al. (eds.) *Hybrid Metaheuristics*, LNCS, vol. 4771, pp. 108–122. Springer (2007)
2. Birattari, M.: *Tuning Metaheuristics: A Machine Learning Perspective*, Studies in Computational Intelligence, vol. 197. Springer (2009)
3. Bortfeldt, A.: A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. *Comp. & Op. Res.* 39(9), 2248–2257 (2012)
4. Doerner, K.F., Fuellerer, G., Hartl, R.F., Gronalt, M., Iori, M.: Metaheuristics for the vehicle routing problem with loading constraints. *Networks* 49(4), 294–307 (2007)
5. Dorigo, M., Stützle, T.: *Ant Colony Optimization*. MIT Press (2004)
6. Fuellerer, G., Doerner, K.F., Hartl, R.F., Iori, M.: Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *EJOR* 201(3), 751–759 (2010)
7. Gendreau, M., Iori, M., Laporte, G., Martello, S.: A tabu search algorithm for a routing and container loading problem. *Trans. Sci.* 40(3), 342–350 (2006)
8. Hutter, F., Hoos, H.H., Leyton-Brown, K., Stützle, T.: ParamILS: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research* 36, 267–306 (Oct 2009)
9. Iori, M., Martello, S.: Routing problems with loading constraints. *TOP* 18, 4–27 (2010)

10. López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., Birattari, M.: The irace package, iterated race for automatic algorithm configuration. Tech. Rep. TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium (2011)
11. Massen, F., Deville, Y., Hentenryck, P.: Pheromone-based heuristic column generation for vehicle routing problems with black box feasibility. In: Beldiceanu, N., Jussien, N., Pinson, É. (eds.) *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, LNCS, vol. 7298, pp. 260–274. Springer Berlin Heidelberg (2012)
12. Prescott-Gagnon, E., Desaulniers, G., Drexler, M., Rousseau, L.M.: European driver rules in vehicle routing with time windows. *Trans. Sci.* 44(4), 455–473 (2010)
13. Reimann, M., Doerner, K., Hartl, R.F.: D-ants: Savings based ants divide and conquer the vehicle routing problem. *Comp. & Op. Res.* 31(4), 563–591 (2004)
14. Ren, J., Tian, Y., Sawaragi, T.: A relaxation method for the three-dimensional loading capacitated vehicle routing problem. In: *System Integration (SII), 2011 IEEE/SICE International Symposium on*. pp. 750–755. IEEE (2011)
15. Ruan, Q., Zhang, Z., Miao, L., Shen, H.: A hybrid approach for the vehicle routing problem with three-dimensional loading constraints. *Comp. & Op. Res.* (2011)
16. Tarantilis, C., Zachariadis, E., Kiranoudis, C.: A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem. *Intelligent Transportation Systems, IEEE Transactions on* 10(2), 255–271 (2009)
17. Toth, P., Vigo, D. (eds.): *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications (2002)
18. Tricoire, F., Doerner, K.F., Hartl, R.F., Iori, M.: Heuristic and exact algorithms for the multiple vehicle routing problem. *OR spectrum* 33(4), 931–959 (2011)
19. Benchmark instances for the 3L-CVRP, <http://www.or.deis.unibo.it/research.html>
20. Benchmark instances for the MPVRP, <http://prolog.univie.ac.at/research/VRPandBPP/>
21. Supp. material, <http://becool.info.ucl.ac.be/resources/ACO-HCG-IRACE>
22. Zhu, W., Qin, H., Lim, A., Wang, L.: A two-stage tabu search algorithm with enhanced packing heuristics for the 3L-CVRP and M3L-CVRP. *Comp. & Op. Res.* 39(9), 2178–2195 (2012)