

A Thompson Sampling Based Operator Selection in Adaptive Large Neighborhood Search for Facility Location with Customer Incompatibilities

Gabriel Souto[1(✉)], Diego Brandão[2], Luidi Simonetti[1], and Pedro Henrique González Silva[1]

[1] Federal University of Rio de Janeiro, Rio de Janeiro, Brazil
{soutog,luidi,pegonzalez}@cos.ufrj.br
[2] Federal Center for Technological Education of Rio de Janeiro, Rio de Janeiro, Brazil
diego.brandao@cefet-rj.br

Abstract. The Multi-Source Capacitated Facility Location Problem with Customer Incompatibilities (MS-CFLP-CI) arises in real-world decision-making contexts such as urban logistics, healthcare service planning, and hazardous material distribution, where operational constraints prevent certain customers from being served by the same facility. These incompatibility constraints significantly increase problem complexity and limit the applicability of exact optimization methods, especially in large-scale scenarios. In this work, we address this challenging problem by proposing an Adaptive Large Neighborhood Search (ALNS) framework enhanced with a learning-based operator selection mechanism. Specifically, we integrate contextual Thompson Sampling to dynamically balance exploration and exploitation based on the current search state. In addition, we introduce a novel conflict-aware destroy operator that explicitly targets regions of the solution space characterized by high incompatibility interactions. The proposed approach is designed to support large-scale decision-making, remaining scalable to instances involving thousands of facilities and customers. Computational experiments on benchmark datasets demonstrate that the method consistently improves both solution quality and robustness compared to state-of-the-art heuristics under comparable time limits. These results indicate that combining adaptive search with learning-based operator selection provides an effective and scalable strategy for solving complex facility location problems with real-world constraints.

Keywords: Large Neighborhood Search · Bandit Algorithms · Adaptive Operator Selection · Facility Location · Combinatorial Optimization

1 Introduction

Facility location problems play a central role in a wide range of real-world decision-making systems, including urban logistics, healthcare service planning,

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16769, pp. 378–395, 2027.
https://doi.org/10.1007/978-3-032-30494-0_24

waste management, and hazardous material distribution [10]. In these applications, planners must determine where to open facilities and how to allocate demand while respecting operational constraints, regulatory requirements, and cost efficiency. These decisions directly impact service quality, safety, and resource utilization, making facility location a key component of modern decision-support systems [3,7].

A challenging aspect in many of these settings is the presence of incompatibility constraints between customers. Such constraints arise when certain entities cannot be jointly served by the same facility due to safety regulations, risk mitigation policies, or operational conflicts. For example, hazardous and non-hazardous materials must often be separated, and competing clients may require exclusive service conditions. These incompatibilities introduce a highly combinatorial structure into the problem, significantly increasing its complexity and limiting the effectiveness of traditional exact optimization methods, especially in large-scale scenarios.

The Multi-Source Capacitated Facility Location Problem with Customer Incompatibilities (MS-CFLP-CI) extends MS-CFLP by prohibiting specific customer pairs from being served by the same facility [7]. This additional disjunctive structure makes even the allocation subproblem NP-hard and leads to large, challenging benchmark instances [3,4]. Recent studies report strong metaheuristics and neighborhood-based methods, but solution quality remains sensitive to operator choices and instance characteristics [3,4,7].

These characteristics make the MS-CFLP-CI relevant for decision-support systems, where planners must evaluate complex trade-offs between cost, capacity, and operational feasibility. In such contexts, scalable and adaptive optimization methods are essential to support high-quality decisions in large-scale and constraint-rich environments.

In this work, we address this challenging optimization problem by proposing an Adaptive Large Neighborhood Search (ALNS) framework for MS-CFLP-CI that combines a new conflict-aware destroy operator (CCF) with a learning-based operator selection mechanism based on Thompson Sampling [11,13]. The goal is to balance exploration and exploitation while adapting the search behavior to the evolving solution structure and instance characteristics, ultimately supporting more effective decision-making in large-scale facility location problems.

The remainder of this paper is organized as follows: Sect. 2 defines the problem and its formulation. Section 3 reviews related work on MS-CFLP-CI. Section 4 describes the proposed ALNS and the CCF operator with Thompson Sampling. Section 5 reports the computational experiments. Finally, Sect. 6 concludes the paper.

2 Problem Definition

For a mathematical definition of MS-CFLP-CI, we consider a set of customers $N = \{1, \ldots, n\}$ and a set of candidate facilities $M = \{1, \ldots, m\}$. Each customer

$i \in N$ has demand $d_i > 0$, each facility $j \in M$ has capacity $s_j > 0$ and fixed opening cost $f_j > 0$, and c_{ij} denotes the unit shipping cost from facility j to customer i. Let $\mathcal{I}$ be the set of incompatible customer pairs; for each $(i_1, i_2) \in \mathcal{I}$, customers i_1 and i_2 cannot be served by the same facility. In the multi-source setting, a customer may be supplied by multiple facilities.

We use continuous variables $x_{ij} \in [0, 1]$ to represent the fraction of customer i's demand supplied by facility j (the shipped quantity is $d_i x_{ij}$). Binary variables y_j indicate whether facility j is open, and binary variables w_{ij} indicate whether facility j supplies customer i with a positive amount. The MS-CFLP-CI can be modeled as the following Mixed Integer Programming [3,4,7]:

$$\min \sum_{i \in N} \sum_{j \in M} c_{ij} d_i x_{ij} + \sum_{j \in M} f_j y_j \tag{1}$$

$$\text{s.t.} \quad \sum_{j \in M} x_{ij} = 1 \qquad \forall i \in N \tag{2}$$

$$\sum_{i \in N} d_i x_{ij} \leq s_j y_j \qquad \forall j \in M \tag{3}$$

$$x_{ij} \leq w_{ij} \qquad \forall i \in N, \forall j \in M \tag{4}$$

$$w_{ij} \leq y_j \qquad \forall i \in N, \forall j \in M \tag{5}$$

$$w_{i_1 j} + w_{i_2 j} \leq 1 \qquad \forall (i_1, i_2) \in \mathcal{I}, \forall j \in M \tag{6}$$

$$x_{ij} \in [0, 1],\ w_{ij} \in \{0, 1\},\ y_j \in \{0, 1\} \qquad \forall i \in N, \forall j \in M \tag{7}$$

The objective (1) minimizes the total opening and transportation costs. Constraints (2) ensure that each customer's demand is satisfied, while Constraint (3) enforces facility capacities and implicitly prevents closed facilities from shipping. Constraints (4) and (5) link the continuous allocation variables to the binary variables, and (6) enforces customer incompatibilities for each facility. The linking Constraint (5) is redundant given Constraint (3), but it is commonly included to strengthen the formulation and improve solver performance [4].

3 Related Works

The MS-CFLP-CI was introduced by [7] along with the first public benchmark instances (wlp). The authors called the problem as a multi-source capacitated location model with incompatibility pairs and proposed a portfolio of independently developed metaheuristics. The portfolio includes MineReduce, a multistart ILS enriched with data mining, a GRASP method, a permutation-coded evolutionary algorithm, and a multistart greedy approach.

Their experiments used a common hardware platform and time limits tied to instance size to ensure fair comparisons. Results indicate that MineReduce is generally the best method across most instances, while the simpler greedy method can be competitive on the very large cases. Beyond algorithmic performance, the work is notable for providing instance generators, a solution validator, and open implementations, enabling direct reproducibility and comparisons.

Ceschia and Schaerf [3] proposed a multi-neighborhood simulated annealing (SA) approach that targets large instances through search space reductions. Their method starts from a greedy initial solution and explores neighborhoods that change suppliers, swap suppliers between customers, and close/open facilities. To control complexity, they restrict solutions to at most two suppliers per customer and limit candidate suppliers to a short list, which reduces the cost of feasibility checks.

The SA method is tuned with F-Race and tested on the wlp dataset and on a new benchmark (cflp-ci) released by the authors. The solver improves upon prior methods, achieving average improvements above 6% relative to MineReduce on most instances. The paper also provides models and lower bounds for smaller instances, strengthening the empirical baseline for MS-CFLP-CI.

Gjergji et al. [4] propose the first Large Neighborhood Search (LNS) designed for the MS-CFLP-CI. Their method introduces a set of problem-aware destroy operators that explicitly account for customer incompatibilities, coupled with a repair phase formulated as a restricted mixed-integer program. By embedding an exact solver within the heuristic loop, the approach effectively combines the intensification of mathematical programming with the flexibility of large-scale neighborhood exploration.

Computational experiments on benchmark datasets demonstrate that the proposed method consistently outperforms state-of-the-art approaches in terms of both solution quality and robustness under comparable time limits.

The observed variability in operator effectiveness motivates the use of adaptive operator selection mechanisms. In addition, we propose a new destroy operator, CCF, designed to exploit incompatible customers. This observation aligns with the objective of the present work, which extends the ALNS framework by integrating Thompson Sampling to dynamically balance exploration and exploitation, allowing the search process to adapt online to instance-specific characteristics.

These works highlight the importance of developing scalable optimization methods to address real-world facility location problems with complex operational constraints.

4 Proposed Method

This section presents the proposed methodology, which extends the Large Neighborhood Search (LNS) framework for the MS-CFLP-CI in two directions. First, we introduce a new conflict-aware destroy operator (CCF) that explicitly exploits the incompatibility structure of the problem. Second, we incorporate a learning-based operator selection mechanism based on contextual Thompson Sampling to dynamically guide the search process.

We begin by describing the set of destroy and repair operators (Sect. 4.1), followed by the Adaptive Large Neighborhood Search (ALNS) framework (Sect. 4.2), and finally the contextual Thompson Sampling mechanism used for operator selection (Sect. 4.3).

4.1 Destroy and Repair Operators

The proposed method relies on a set of destroy and repair operators that define large neighborhoods explored during the search. We adopt the operators introduced by [4] as a baseline and extend this set with a new conflict-aware operator.

Operators from Gjergji et al. We consider the three destroy operators proposed in [4], namely CW (Cheapest Facilities), CC (Cheapest Customers), and EC (Expensive Customers), together with an MIP-based repair operator.

The CW operator computes, for each open facility f', the average cost of serving the customers currently assigned to another facility f, and selects the ℓ facilities with the lowest values. The customers assigned to these facilities define the subproblem.

The CC operator selects the ℓ customers with the lowest unit shipping costs and includes these customers along with their associated facilities in the subproblem.

The EC operator builds a pool consisting of the most expensive customer of each open facility, selects one of them (either randomly or based on cost), and then includes the ℓ facilities with the lowest cost for that customer, along with their associated customers.

Given the subset of customers selected by a destroy operator, the repair phase consists of solving a restricted mixed-integer programming (MIP) model that preserves all original constraints while allowing customer reassignment and facility open/close decisions within the selected subset.

Conflict-Aware Customer Facility (CCF) Operator. While the operators of [4] are driven by cost-based criteria, they do not explicitly exploit the incompatibility structure of the MS-CFLP-CI. However, infeasibility and poor solution quality are often driven by strongly coupled, incompatible customers.

To address this limitation, we propose a conflict-aware destroy operator, named **CCF**, that targets regions of the solution space with high incompatibility density.

The operator assigns a score to each customer i defined as:

$$\text{score}(i) = \alpha \cdot \text{costShare}(i) + (1 - \alpha) \cdot \deg_I(i), \tag{8}$$

where $\text{costShare}(i)$ represents the fraction of total transportation cost associated with customer i, and $\deg_I(i)$ denotes the normalized degree of i in the incompatibility graph.

A seed customer is selected with probability proportional to this score. The subproblem is then expanded as:

$$U = \{i\} \cup N_I(i) \cup \text{coServed}(i), \tag{9}$$

where $N_I(i)$ represents the set of incompatible neighbors of i, and $\text{coServed}(i)$ denotes customers currently sharing facilities with i.

If the size of U exceeds a predefined limit ν, the set is truncated by prioritizing high-degree incompatible neighbors and selecting additional customers based on their score.

This construction allows the repair phase to focus on highly constrained regions of the solution, enabling more effective restructuring of assignments under incompatibility constraints.

4.2 Adaptive Large Neighborhood Search Framework

Given the set of destroy and repair operators described above, we embed them into an Adaptive Large Neighborhood Search (ALNS) framework [11].

At each iteration, the algorithm selects a destroy operator to define a subproblem, which is then reconstructed by the repair operator. This process allows the exploration of large and adaptive neighborhoods, balancing intensification and diversification.

Starting from an initial solution s_0, generated using the constructive heuristic of [4], the algorithm iteratively applies destroy and repair steps to produce candidate solutions.

Acceptance of candidate solutions follows a simulated annealing criterion, allowing occasional worsening moves with probability:

$$\exp\left(-\frac{f(s') - f(s)}{T}\right),$$

where $f(\cdot)$ denotes the objective value and T is the current temperature.

In contrast to classical ALNS, where operator selection is governed by segment-based weight adaptation and roulette-wheel selection, we replace this mechanism with a learning-based approach described in the next section.

4.3 Contextual Thompson Sampling for Operator Selection

The effectiveness of destroy operators may vary significantly across search phases and instance characteristics. The adaptive weight mechanism described in the ALNS framework of [11], in which operators are estimated from aggregated rewards over fixed-length segments, selects operators using a roulette wheel. While effective and widely adopted, adaptation occurs only at segment boundaries, and operator choice is independent of the current search state. As a result, the method can react slowly to changes in search dynamics and does not explicitly exploit context.

To address this, we replace the segment-based roulette selection with a learning-based mechanism using contextual Thompson Sampling [1,2,6,8,12,13]. In our implementation, each arm corresponds to a *pair* (destroy, repair); however, for the TS runs reported here, we fix the repair to MIP-only and use the (CW, CC, EC, CCF) destroy operators, so each arm corresponds to one destroy operator. We also model operator selection as a linear contextual bandit problem, in which the expected reward of each arm is assumed to be a linear function of the current search context.

Let $x \in \mathbb{R}^d$ denote the context vector, where d is the number of contextual features used to summarize the current search state.

$$x = [1,\ \text{progress},\ T/T_0,\ \text{no\_improve},\ \text{last\_acc},\ \text{last\_imp},\ \text{infeas},\ \text{open\_ratio},\ \text{conflict\_ratio}] \tag{10}$$

where progress is elapsed time over the time limit, T/T_0 is the normalized temperature, no_improve is the normalized iterations since last best, last_acc and last_imp capture the previous acceptance and relative improvement, infeas flags the last infeasible attempt, open_ratio is the fraction of open facilities, and conflict_ratio is a fraction of facility–customer pairs that are blocked by incompatibilities in the current allocation solution. For each arm a, the expected reward is modeled as

$$\mathbb{E}[r \mid x, a] = x^\top \theta_a,$$

with an independent Bayesian linear model per arm. At each ALNS iteration, we sample $\tilde{\theta}_a$ from the posterior of each arm and select the arm maximizing $x^\top \tilde{\theta}_a$.

The selected features capture aspects of the search dynamics, including progress in time, acceptance behavior, solution structure, and conflict intensity, enabling TS to condition operator choice on search phase and problem-specific difficulty.

The reward used in our TS runs is binary: if the candidate is accepted and yields either a new global best or an improvement over the current solution of at least a relative threshold τ, the reward is 1; otherwise, it is 0. If a candidate is rejected or the destroy/repair fails, the reward is 0. To cope with non-stationarity, posterior updates apply exponential forgetting with factor ρ. We adopt a binary reward for simplicity and robustness, so that operator credit reflects whether an iteration produced a meaningful accepted improvement.

It is important to emphasize that the proposed Thompson Sampling mechanism replaces only the operator selection and adaptation components of the baseline ALNS. All remaining components of the baseline algorithm, namely the MIP-based repair procedure, the simulated annealing acceptance criterion, and the overall iterative search loop, are left unchanged. This design choice ensures that the proposed method preserves the structure of ALNS while introducing a context-aware operator selection mechanism capable of adapting online to the evolving search dynamics.

Algorithm 1 presents the resulting ALNS-TS algorithm as implemented in our experiments. The initial solution s_0 is generated using the same constructive heuristic employed by the LNS, which opens facilities according to fixed costs and assigns demands greedily. Within the Thompson Sampling framework, each arm corresponds to a pair in the Cartesian product $D \times R$ of destroy and repair operators. Since the repair set R contains only the MIP-based repair in our implementation, each arm effectively represents a distinct destroy operator (CW, CC, EC, and CCF). At each iteration, operator selection is conditioned on a contextual feature vector x computed from the current search state, and the algorithm proceeds until the time limit t is reached.

Algorithm 1: ALNS-TS

Input: Initial solution s_0, destroy set D, repair set R (MIP), time limit t
Output: Best solution s^*
Build s_0 with the LNS constructive heuristic; initialize posterior parameters for each arm $a \in D \times R$; $s \leftarrow s_0$, $s^* \leftarrow s_0$;
while *time* $< t$ **do**
  Compute context vector x;
  Sample $\tilde{\theta}_a$ for each arm $a = (d, r) \in D \times R$ and select $a = \arg\max_a x^\top \tilde{\theta}_a$;
  Apply destroy operator d to s and build subproblem $(U, F^{open}, F^{closed})$;
  Apply repair operator r to obtain candidate s';
  Compute reward q from the outcome of s';
  Update posterior of arm a with reward q and forgetting factor ρ;
  accepted $\leftarrow$ simulated-annealing decision on s' given s;
  if *accepted* **then**
    if $f(s') < f(s^*)$ **then**
      $s^* \leftarrow s'$;
    $s \leftarrow s'$;
return s^*;

In this way, operator selection becomes adaptive at every iteration, allowing the search to react online to changes in solution quality, stagnation, and incompatibility structure.

5 Computational Experiments

This section summarizes the experimental setup and evaluation methodology, then reports results in three parts: experimental details and benchmark instances, an ablation analysis to isolate the contribution of Thompson Sampling, and a comparison with the state-of-the-art LNS, followed by a statistical significance analysis of the observed improvements.

5.1 Experimental Details and Instances

We follow the evaluation methodology of [4] and use the two public benchmark instance sets for MS-CFLP-CI. The `wlp` dataset from [7] contains 30 instances, ranging from small cases (50 facilities) to very large ones (up to 3000 facilities). The `cflp-ci` dataset from [3] includes 50 instances with sizes up to roughly 3000 facilities, providing a complementary and challenging test bed. Each instance is solved under two timeouts: a short timeout of $10\sqrt{m}$ seconds and a long timeout of m seconds, where m is the number of facilities.

For ALNS-TS, we use the destroy operators CW, CC, EC, and the conflict-aware CCF, with MIP-only repair. Parameter values were selected via a grid-search procedure over the main ALNS-TS hyperparameters. The repair subproblem is solved with a time limit of 20 s per call, a maximum of two extra facilities

opened $(o + 2)$, and a relative MIP gap of 10^{-4}, using a single CPLEX thread. The subproblem size follows the dynamic rule $\ell = \nu|F|/n$ with $\nu = 50$ (Eq. (12) in 4), and closed facilities are selected using the same probability $p_{\text{MC}} = 0.5$ as in the LNS operators. Simulated annealing acceptance is enabled, with a time-based schedule and final temperature ratio of 0.05; the initial temperature is set to accept a 1% worsening with probability 0.5.

For contextual Thompson Sampling, we use a binary reward with success threshold $\tau = 0.001$, a Gaussian prior with $\lambda = 1.0$, noise $\sigma = 0.2$, and forgetting factor $\rho = 0.995$, with the `basic` context feature set. All methods use the same constructive heuristic (open facilities by fixed cost and assign demands greedily) and are executed on a single CPU core for a fair comparison. All experiments were run on a single machine with 16 GB RAM and an Intel Core i7-11800H (8 cores/16 threads, 2.30 GHz), x86_64 architecture.

The computational overhead introduced by the contextual Thompson Sampling mechanism is negligible compared to the cost of the MIP repair phase. Per iteration, TS requires only the construction of a low-dimensional context vector $(d = 9)$, one posterior sample per arm, and an inner-product evaluation, which together take well under one millisecond. Since each MIP repair call is bounded by 20 s and typically dominates the iteration time, the relative overhead of the learning mechanism remains below 0.1% of total runtime in our experiments.

5.2 Ablation Analysis: Impact of Thompson Sampling

In order to investigate the contribution of the proposed Thompson Sampling–based operator selection mechanism, we conduct an ablation study. We compare two algorithmic variants: the Adaptive Large Neighborhood Search without learning-based control (ALNS), and the proposed method ALNS-TS, which replaces the classical weight adaptation and roulette-wheel selection by contextual Thompson Sampling.

The evaluation is conducted on a subset of 15 instances randomly selected from the full benchmark set of 80 instances. Each instance is solved independently five times using a long time limit of m seconds, where m denotes the number of facilities in the instance. For each method, we report both the best solution obtained across runs (*best*) and the average solution quality (*avg*), allowing us to assess not only peak performance but also robustness.

Table 1 reports the best solution values obtained by ALNS, and ALNS-TS across the 15 tested instances. Overall, ALNS-TS attains the best solution quality on the majority of instances, outperforming the baseline ALNS.

In particular, ALNS-TS is the method capable of reaching the best known solution for instance `wlp01`, highlighting its enhanced ability to identify high-quality regions of the search space. For instance `wlp21`, ALNS-TS and ALNS tie by both reaching the optimal solution, indicating that the learning-based operator selection does not degrade performance when the baseline ALNS is already effective.

On a small number of instances (e.g., `cflp_ci_19`, `cflp_ci_48`, and `wlp15`), ALNS achieves marginally better best solutions than ALNS-TS. However, the

Table 1. Best and average solution values: ALNS and ALNS-TS.

Instance	Best		Average	
	ALNS	ALNS-TS	ALNS	ALNS-TS
cflp_ci_06	152.710	**152.650**	152.894	**152.828**
cflp_ci_10	600.146	**600.060**	600.458	**600.391**
cflp_ci_11	**30.739**	**30.739**	30.758	**30.783**
cflp_ci_19	**149.900**	149.993	**150.096**	150.097
cflp_ci_28	463.140	**463.111**	463.411	**463.302**
cflp_ci_39	189.904	**189.892**	189.982	**189.976**
cflp_ci_48	**765.460**	765.774	**765.604**	766.018
wlp01	28.730	**28.716**	28.742	**28.728**
wlp02	53.120	**52.975**	53.181	**53.146**
wlp03	64.415	**64.395**	64.508	**64.607**
wlp12	297.542	**297.349**	297.661	**297.535**
wlp15	**457.249**	457.251	**457.477**	457.535
wlp17	562.406	**562.386**	562.656	**562.633**
wlp21	**38.067**	**38.067**	38.150	**38.112**
wlp28	455.931	**455.826**	456.201	**456.032**

absolute differences are small when compared to the scale of the objective values, remaining within a small range even for large-cost instances. This indicates that the integration of Thompson Sampling does not introduce bias and remains competitive even in unfavorable cases.

The average solution values reported in Table 1 reveal an advantage of ALNS-TS over the ALNS. Across the ablation set, ALNS-TS improves the average solution in 12 out of 15 instances and never degrades performance by more than 0.05%.

Notably, ALNS-TS consistently improves upon ALNS, even when differences in best solution quality are marginal. This behavior suggests that the learning-based operator selection mechanism leads to a more stable search process, mitigating performance variability induced by random decisions and initial conditions. The gains in average performance are particularly relevant under the considered long time limit of m seconds, as they reflect sustained efficiency rather than isolated successful runs.

On instances where ALNS attains slightly better average values (e.g., `cflp_ci_19`, `cflp_ci_48`, and `wlp15`), the observed gaps remain small in absolute terms and do not outweigh the overall robustness gains achieved by ALNS-TS across the test set. Overall, the results indicate that contextual Thompson Sampling enables a better balance between intensification and diversification by adapting operator selection at every iteration based on the current search context, rather than relying on delayed, segment-based updates.

Overall, the ablation study demonstrates that the performance gains observed in ALNS-TS stem directly from the integration of Thompson Sampling into the operator selection mechanism. The method consistently improves average performance and frequently improves best-known solutions, without introducing instability or significant performance regressions. These results confirm that learning-based, context-aware operator selection constitutes an enhancement over ALNS schemes in MS-CFLP-CI.

5.3 Comparison with State-of-the-Art Methods

Based upon the ablation analysis, we now evaluate the ALNS-TS method against the current state of the art for MS-CFLP-CI, the LNS of [4]. Under the same two timeouts ($10\sqrt{m}$ and m seconds). Tables 2 and 3 report the best (min) and average (avg) solution values over the ten runs for each instance. The results show a clear and consistent advantage for ALNS-TS in both datasets, in terms of both best and average costs.

For `wlp` (Table 2), ALNS-TS improves the best solution in 26/30 instances under the short timeout and 28/30 under the long timeout; for average solutions, ALNS-TS is better in 29/30 instances under both budgets. On average, the relative improvement of ALNS-TS over LNS is about 0.79% (best, short), 0.88% (avg, short), 0.99% (best, long), and 1.09% (avg, long).

For `cflp-ci` (Table 3), the dominance is even stronger: ALNS-TS improves the best solution in 49/50 instances under the short timeout and 50/50 under the long timeout, and improves the average solution in 49/50 instances under the short timeout and 50/50 under the long timeout. The mean relative gains are approximately 0.76% (best, short), 0.94% (avg, short), 0.20% (best, long), and 0.24% (avg, long).

To compare methods on a common scale, we compute the percentage gap to the best-known solutions (BKS) as in [4]:

$$\text{GAP}(\%) = \frac{Z_{\text{sol}} - \text{BKS}}{\text{BKS}} \cdot 100, \tag{11}$$

where Z_{sol} is the solution value (min or avg) for a given method and timeout. We compare LNS and ALNS-TS using the same BKS values per instance, as shown in Table 2 and 3.

Figure 1 shows the GAP values for all 80 instances (30 `wlp` + 50 `cflp-ci`) under the m seconds timeout, demonstrating both best and average solutions for LNS and ALNS-TS. The figure highlights two patterns: (i) ALNS-TS shifts the GAP curves downward for both min and avg, indicating improved solution quality relative to LNS, and (ii) the separation between the average curves is stable across instances, evidencing reduced variance and improved robustness.

These improvements suggest that the proposed approach is suitable for supporting large-scale decision-making in real-world facility location systems with complex operational constraints.

Table 2. Results for dataset `wlp`, both timeouts.

Inst	$10\sqrt{m}$ seconds timeout				m seconds timeout			
	LNS		ALNS-TS		LNS		ALNS-TS	
	min	avg	min	avg	min	avg	min	avg
wlp01	28768.00	28805.30	**28716.00**	**28720.67**	28768.00	28805.30	**28716.00**	**28716.00**
wlp02	53133.50	53247.90	**52987.00**	**53011.33**	53133.50	53247.90	**52990.00**	**53017.75**
wlp03	**64482.50**	**64641.80**	64594.00	64754.25	**64482.50**	**64641.80**	64504.00	64751.00
wlp04	85039.50	85159.10	**84831.00**	**84881.42**	85039.50	85159.10	**84714.00**	**84723.58**
wlp05	104413.50	104639.80	**104000.00**	**104023.92**	104413.50	104639.80	**104095.00**	**104312.67**
wlp06	112169.50	112349.10	**111808.00**	**111915.25**	112169.50	112349.10	**112202.00**	**112326.45**
wlp07	163356.50	163489.60	**162760.00**	**162799.42**	163356.50	163489.60	**163125.00**	**163185.08**
wlp08	188851.50	189195.50	**188574.00**	**188582.00**	188851.50	189195.50	**188316.00**	**188461.67**
wlp09	219567.50	219879.00	**218849.00**	**219108.75**	219567.50	219879.00	**218663.00**	**218707.83**
wlp10	246586.50	246892.50	**245739.00**	**245836.67**	246586.50	246892.50	**245491.00**	**245664.75**
wlp11	290420.50	290620.40	**289364.00**	**289409.25**	290420.50	290620.40	**289066.00**	**289193.42**
wlp12	299146.00	299360.00	**297940.00**	**298086.25**	299146.00	299360.00	**297502.00**	**297637.33**
wlp13	316751.00	316879.20	**315340.00**	**315417.42**	316751.00	316879.20	**315225.00**	**315330.50**
wlp14	397169.00	397500.70	**395373.00**	**395618.00**	397169.00	397500.70	**395134.00**	**395531.67**
wlp15	460259.00	460909.00	**457843.00**	**458026.25**	460259.00	460909.00	**457438.00**	**457535.50**
wlp16	534789.50	535111.30	**531344.00**	**531795.75**	534789.50	535111.30	**530771.00**	**530895.25**
wlp17	566845.50	567518.70	**563344.00**	**563815.58**	566845.50	567518.70	**562509.00**	**562929.50**
wlp18	633046.50	633938.90	**628752.00**	**628847.25**	633046.50	633938.90	**627139.00**	**627389.83**
wlp19	752678.00	754195.30	**743934.00**	**744049.42**	752678.00	754195.30	**741398.00**	**741921.08**
wlp20	1000880.00	1004303.50	**976396.00**	**976783.92**	1000880.00	1004303.50	**971615.00**	**971997.58**
wlp21	**38128.00**	38237.80	38176.00	**38176.00**	38128.00	38237.80	**38067.00**	**38067.00**
wlp22	**74781.00**	75008.40	74899.00	**74912.50**	74781.00	75008.40	**74634.00**	**74710.67**
wlp23	**119622.50**	119806.90	119686.00	**119703.00**	119622.50	119806.90	**119412.00**	**119518.33**
wlp24	169886.00	170019.20	**169471.00**	**169648.50**	169886.00	170019.20	**169206.00**	**169234.33**
wlp25	231324.00	231536.90	**230477.00**	**230581.00**	231324.00	231536.90	**230453.00**	**230616.33**
wlp26	291883.50	292234.30	**291022.00**	**291088.67**	291883.50	292234.30	**290360.00**	**290467.83**
wlp27	391918.00	392156.80	**389970.00**	**390367.00**	391918.00	392156.80	**389331.00**	**389471.33**
wlp28	458944.50	459378.70	**456689.00**	**456805.58**	458944.50	459378.70	**455730.00**	**455877.75**
wlp29	597152.00	597880.80	**594279.00**	**594484.58**	597152.00	597880.80	**592512.00**	**592688.83**
wlp30	883828.00	884985.50	**871747.00**	**872360.58**	883828.00	884985.50	**868392.00**	**868714.33**
Avg	325860.70	326329.40	323296.80	323453.67	325860.70	326329.40	322623.67	322786.51

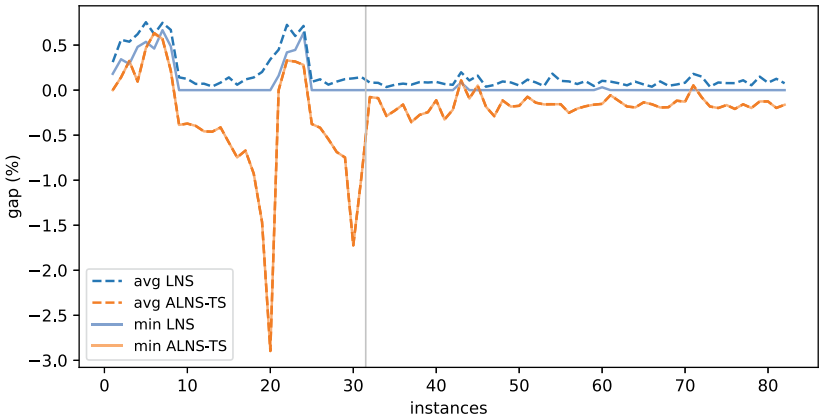

Fig. 1. GAP (%) per instance for LNS and ALNS-TS (long timeout). Negative values indicate solutions better than the BKS.

Table 3. Results for dataset `cflp-ci`, both timeouts.

Inst	$10\sqrt{m}$ seconds timeout				m seconds timeout			
	LNS		ALNS-TS		LNS		ALNS-TS	
	min	avg	min	avg	min	avg	min	avg
cflp-ci-00	418220.00	418769.90	**417946.00**	**418025.75**	417459.00	417808.80	**417026.00**	**417124.45**
cflp-ci-01	572832.00	575196.50	**558771.00**	**559844.50**	558300.00	558751.80	**556994.00**	**557481.91**
cflp-ci-02	445260.00	448563.50	**434599.00**	**434630.33**	434732.00	434882.70	**433483.00**	**433592.64**
cflp-ci-03	1507550.00	1513753.00	**1484520.00**	**1484716.75**	1477770.00	1478651.00	**1474278.00**	**1474904.55**
cflp-ci-04	1206190.00	1208104.00	**1204026.00**	**1204741.67**	1202900.00	1203768.00	**1200752.00**	**1201062.00**
cflp-ci-05	362702.00	363627.80	**357721.00**	**358103.92**	358155.00	358373.10	**356849.00**	**356992.82**
cflp-ci-06	153306.00	153663.10	**153096.00**	**153187.67**	153212.00	153349.10	**152779.00**	**152817.82**
cflp-ci-07	444237.00	445113.40	**439247.00**	**439818.00**	439747.00	440118.70	**438441.00**	**438663.82**
cflp-ci-08	695499.00	696752.50	**688982.00**	**689023.17**	688789.00	689417.00	**687619.00**	**688058.18**
cflp-ci-09	328371.00	329313.00	**325840.00**	**325944.25**	326288.00	326509.40	**325078.00**	**325239.73**
cflp-ci-10	604208.00	605182.70	**600782.00**	**601044.25**	601848.00	602204.70	**600272.00**	**600441.09**
cflp-ci-11	30746.00	30821.80	**30779.00**	**30779.00**	30772.00	30806.80	**30779.00**	**30779.00**
cflp-ci-12	454795.00	455467.90	**454375.00**	**454561.58**	454615.00	455101.90	**454193.00**	**454287.45**
cflp-ci-13	382532.00	383371.60	**382094.00**	**382268.67**	382412.00	383026.10	**381176.00**	**382154.36**
cflp-ci-14	1018520.00	1019821.00	**1015368.00**	**1015995.92**	1015270.00	1015657.00	**1012966.00**	**1013298.18**
cflp-ci-15	555115.00	556381.90	**550397.00**	**550855.42**	550969.00	551284.40	**549376.00**	**549747.09**
cflp-ci-16	878886.00	879700.80	**877093.00**	**877301.50**	876911.00	877754.10	**875826.00**	**876002.45**
cflp-ci-17	1218110.00	1219417.00	**1214177.00**	**1215260.58**	1212530.00	1213562.00	**1210262.00**	**1210754.45**
cflp-ci-18	893188.00	894583.40	**891207.00**	**891658.58**	891134.00	891594.60	**889582.00**	**890156.36**
cflp-ci-19	150444.00	150730.60	**150036.00**	**150163.33**	150326.00	150501.10	**150173.00**	**150195.82**
cflp-ci-20	1266230.00	1268089.00	**1260923.00**	**1261473.67**	1257690.00	1258765.00	**1255345.00**	**1255639.27**
cflp-ci-21	454159.00	454484.90	**452655.00**	**453088.25**	452760.00	452964.90	**451844.00**	**452034.73**
cflp-ci-22	**309532.00**	310162.10	309645.00	**309702.83**	309350.00	309914.20	**308590.00**	**308682.91**
cflp-ci-23	408511.00	408954.40	**406688.00**	**406854.58**	407059.00	407477.40	**406314.00**	**406405.82**
cflp-ci-24	465496.00	467897.30	**460252.00**	**460433.25**	460253.00	460698.20	**459030.00**	**459214.09**

(*continued*)

Table 3. (*continued*)

Inst	$10\sqrt{m}$ seconds timeout				m seconds timeout			
	LNS		ALNS-TS		LNS		ALNS-TS	
	min	avg	min	avg	min	avg	min	avg
cflp-ci-25	800345.00	803679.60	**780863.00**	**782003.42**	780002.00	780531.20	**778314.00**	**778484.00**
cflp-ci-26	689491.00	690836.70	**684720.00**	**685073.42**	683412.00	684091.60	**681708.00**	**682126.64**
cflp-ci-27	1430360.00	1436020.00	**1410422.00**	**1411582.58**	1405340.00	1405995.00	**1402442.00**	**1402920.27**
cflp-ci-28	463857.00	464817.80	**463336.00**	**463439.58**	464003.00	464335.60	**463063.00**	**463163.36**
cflp-ci-29	1370480.00	1374636.00	**1362198.00**	**1362425.92**	1356660.00	1357946.00	**1354323.00**	**1355330.27**
cflp-ci-30	573256.00	573932.80	**570948.00**	**571205.67**	571667.00	572104.40	**570262.00**	**570499.91**
cflp-ci-31	1202910.00	1207278.00	**1193317.00**	**1194103.33**	1191740.00	1192378.00	**1188694.00**	**1189288.36**
cflp-ci-32	482193.00	482863.10	**479186.00**	**479293.42**	478800.00	479249.60	**477774.00**	**477943.18**
cflp-ci-33	1071180.00	1073044.00	**1064393.00**	**1064523.33**	1060900.00	1061572.00	**1059286.00**	**1059571.82**
cflp-ci-34	1143900.00	1146552.00	**1138734.00**	**1139118.08**	1137760.00	1138195.00	**1135649.00**	**1135922.18**
cflp-ci-35	546574.00	549328.20	**530529.00**	**530698.33**	529424.00	529936.30	**528198.00**	**528557.00**
cflp-ci-36	548831.00	549771.90	**546476.00**	**546576.58**	546761.00	547024.40	**545255.00**	**545733.36**
cflp-ci-37	1275540.00	1276290.00	**1272054.00**	**1272499.58**	1269810.00	1270609.00	**1268299.00**	**1268602.09**
cflp-ci-38	1044810.00	1045944.00	**1042490.00**	**1043008.25**	1042340.00	1043200.00	**1040646.00**	**1041029.09**
cflp-ci-39	190422.00	190673.10	**189721.00**	**190121.92**	190241.00	190584.20	**190124.00**	**190324.45**
cflp-ci-40	630056.00	630750.30	**627629.00**	**628032.50**	627341.00	628284.50	**626632.00**	**626912.73**
cflp-ci-41	1308560.00	1313895.00	**1284233.00**	**1284616.00**	1279750.00	1280235.00	**1277021.00**	**1277684.82**
cflp-ci-42	417079.00	417824.10	**416317.00**	**416397.75**	416525.00	416879.00	**415698.00**	**415913.64**
cflp-ci-43	501463.00	503412.80	**497322.00**	**497380.67**	496873.00	497260.80	**495732.00**	**495922.00**
cflp-ci-44	759045.00	763009.50	**748200.00**	**749311.42**	748310.00	748891.20	**746744.00**	**746887.55**
cflp-ci-45	714632.00	715128.10	**711231.00**	**711814.42**	711314.00	712092.30	**709883.00**	**710339.45**
cflp-ci-46	626501.00	628046.60	**622111.00**	**622520.25**	621462.00	621798.80	**620142.00**	**620312.27**
cflp-ci-47	405474.00	406156.90	**405102.00**	**405343.58**	405218.00	405824.70	**404364.00**	**404707.82**
cflp-ci-48	768781.00	769566.70	**767034.00**	**767180.00**	766996.00	767625.10	**765513.00**	**765822.73**
cflp-ci-49	300901.00	301208.90	**299875.00**	**300108.67**	300328.00	300705.40	**299735.00**	**299811.91**
Avg	689825.60	691451.78	684593.20	684957.12	683884.56	684405.82	682501.36	682790.84

5.4 Statistical Significance Analysis

While average and best-case results provide important evidence of algorithmic performance, they are insufficient to determine whether observed improvements are due to stochastic variability. To assess whether the gains achieved by ALNS-TS over the baseline LNS are statistically significant, we conduct a hypothesis testing analysis based on paired non-parametric tests.

Each algorithm was executed independently ten times per instance, yielding stochastic solution distributions with non-normal characteristics. These characteristics motivate the use of non-parametric paired tests, which do not rely on distributional assumptions and are recommended in the evaluation of metaheuristic algorithms.

We employ the Wilcoxon signed-rank test [5,9] to compare LNS and ALNS-TS under the short timeout ($10\sqrt{m}$ seconds). For each instance, the average solution cost over the ten runs is computed for both algorithms, and the test is applied to the paired per-instance differences. The null and alternative hypotheses are defined as

$$H_0 : \text{median}(C_{\text{LNS}} - C_{\text{ALNS-TS}}) = 0, \quad H_1 : \text{median}(C_{\text{LNS}} - C_{\text{ALNS-TS}}) > 0,$$

where C denotes the solution cost. The alternative hypothesis reflects the expected directional improvement of ALNS-TS, which aims to achieve lower costs through adaptive operator selection. All tests are conducted at a significance level of $\alpha = 0.05$.

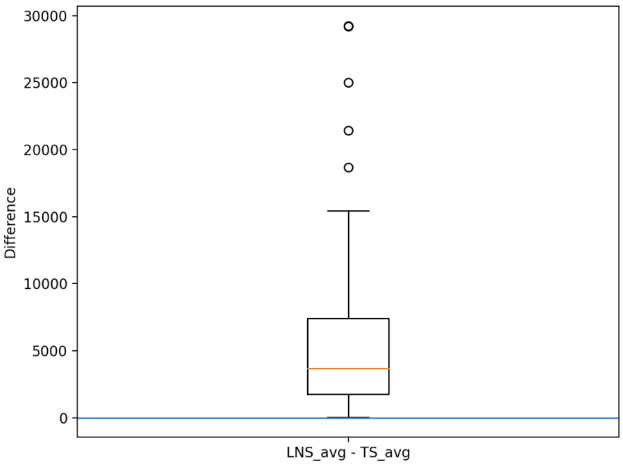

Fig. 2. Paired differences in per-instance average costs for `cflp-ci` ($C_{\text{LNS}} - C_{\text{ALNS-TS}}$). Positive values indicate ALNS-TS is better.

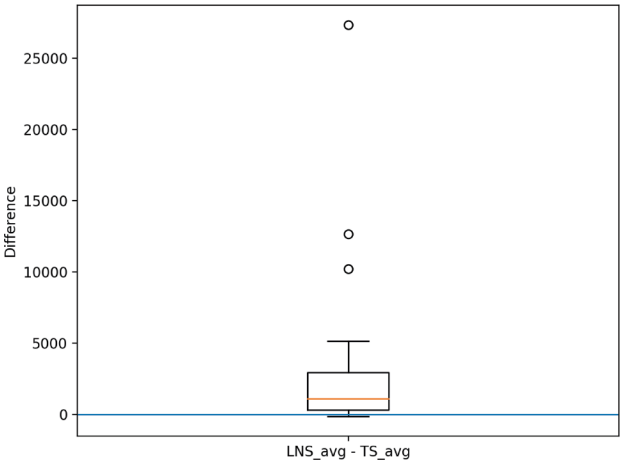

Fig. 3. Paired differences in per-instance average costs for `wlp` ($C_{\text{LNS}} - C_{\text{ALNS-TS}}$). Positive values indicate ALNS-TS is better.

Figures 2 and 3 summarize the paired differences in per-instance average costs, $C_{\text{LNS}} - C_{\text{ALNS-TS}}$. Positive values indicate instances for which ALNS-TS outperforms LNS.

Across both instance sets, the distributions of paired differences are predominantly positive, with median values clearly above zero. This indicates that, for most instances, ALNS-TS consistently achieves lower average costs than LNS. Although a small number of instances exhibit near-zero differences, these cases are rare and of limited magnitude when compared to the overall cost scale.

The Wilcoxon signed-rank test confirms that the observed improvements of ALNS-TS over LNS are statistically significant for both instance sets. In particular, the *p-value* the test rejects the null hypothesis at the 5% significance level, indicating that the median per-instance average cost achieved by ALNS-TS is lower than that of LNS.

For the `wlp` instances, the p-values are $p = 9.64 \times 10^{-8}$ for the minimum-cost comparison and $p = 4.66 \times 10^{-9}$ for the average-cost comparison. For the larger `cflp-ci` instances, the corresponding p-values are even smaller, with $p = 3.26 \times 10^{-10}$ for minimum costs and $p = 2.57 \times 10^{-10}$ for average costs.

These small p-values provide statistical evidence that the median per-instance solution cost achieved by ALNS-TS is lower than that of LNS, ruling out stochastic variability as an explanation for the observed performance differences.

The results reflect a reliable improvement induced by the learning-based operator selection mechanism. By adapting operator choices online and in a context-aware manner, Thompson Sampling enables a more effective balance between intensification and diversification, resulting in both lower average costs and improved stability across independent runs.

Overall, the statistical analysis provides evidence that integrating Thompson Sampling into the ALNS framework yields a statistically significant enhancement over the LNS approach.

6 Conclusions

This paper introduced ALNS-TS, an Adaptive Large Neighborhood Search framework augmented with a contextual Thompson Sampling mechanism for operator selection, targeting the Multi-Source Capacitated Facility Location Problem with Customer Incompatibilities (MS-CFLP-CI). The proposed approach preserves the core structure of the LNS algorithm, proposing a new conflict-aware operator and replacing traditional segment-based operator adaptation with an online, context-aware selection strategy.

Computational results on the `wlp` and `cflp-ci` benchmark sets show that ALNS-TS improves both best-case and average solution quality over the current state-of-the-art LNS, with statistically significant gains across both benchmarks. These gains are observed across different instance sizes and time budgets, indicating that the proposed learning mechanism is effective in the search process.

The methodological contribution of this work has two components with different scopes. The contextual Thompson Sampling mechanism for operator selection is problem-agnostic: it relies only on a generic search-state feature vector

(progress, temperature, stagnation, acceptance signals, and structural ratios) and on a Bayesian linear model per arm, which can be transferred to any ALNS-based solver with multiple destroy/repair operators. In particular, vehicle routing, scheduling, and network design problems for which ALNS is a standard approach could benefit directly from the same selection scheme, possibly augmented with a few problem-specific context features. In contrast, the conflict-aware CCF operator is specifically tailored to MS-CFLP-CI, as it exploits the incompatibility graph structure of the problem; analogous problem-aware operators would need to be designed for other domains, but the integration pattern with contextual Thompson Sampling remains the same.

Several directions for future research emerge from this work. First, richer context representations and alternative reward definitions could be explored to further improve credit assignment and responsiveness to search dynamics. Second, the proposed framework extends to settings with multiple repair operators, including hybrid exact–heuristic repair strategies.

Overall, this paper demonstrates that embedding contextual Thompson Sampling into the ALNS framework yields a robust and statistically significant improvement over classical LNS approaches, highlighting the potential of learning-based operator selection as a general paradigm for complex combinatorial optimization problems.

References

1. Abeille, M., Lazaric, A.: Linear Thompson sampling revisited. Artif. Intel. Stat. (2017)
2. Agrawal, S., Goyal, N.: Thompson sampling for contextual bandits with linear payoffs. In: Proceedings of the 30th International Conference on Machine Learning, pp. 127–135. ICML (2013)
3. Ceschia, S., Schaerf, A.: Multi-neighborhood simulated annealing for the capacitated facility location problem with customer incompatibilities. Comput. Ind. Eng. **188**, 109858 (2024). https://doi.org/10.1016/j.cie.2023.109858
4. Gjergji, I., Kletzander, L., Musliu, N., Schaerf, A.: Large neighborhood search for capacitated facility location with customer incompatibilities. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '25 (2025). https://doi.org/10.1145/3712256.3726355
5. Hettmansperger, T.P., McKean, J.W.: Robust Nonparametric Statistical Methods. CRC Press (2010)
6. Lattimore, T., Szepesvári, C.: Bandit Algorithms. Cambridge University Press (2020)
7. Maia, M.R.H., et al.: Metaheuristic techniques for the capacitated facility location problem with customer incompatibilities. Soft. Comput. **27**, 4685–4698 (2023). https://doi.org/10.1007/s00500-022-07600-z
8. Osband, I., Russo, D., Van Roy, B.: More efficient reinforcement learning via posterior sampling. In: Advances in Neural Information Processing Systems (2013)
9. Pratt, J.W.: Remarks on zeros and ties in the Wilcoxon signed rank procedures. J. Am. Stat. Assoc. **54**(287), 655–667 (1959)
10. Revelle, C.S., Laporte, G.: The plant location problem: new models and research prospects. Oper. Res. **44**(6), 864–874 (1996)

11. Røpke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. **40**(4), 455–472 (2006). https://doi.org/10.1287/trsc.1050.0135
12. Russo, D., Van Roy, B., Kazerouni, A., Osband, I., Wen, Z.: A tutorial on Thompson sampling. Found. Trends Mach. Learn. **11**(1), 1–96 (2018)
13. Thompson, W.R.: On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika **25**(3–4), 285–294 (1933)