Appendix A: Proofs

Proof of Proposition 1. Let π be the policy associated to sequence $C=(v_0,v_1,...,v_n)$. We consider state $x_k=(v_k,0)$ with charging station v_k not being available and $v_k\neq v_n$. Thus, for $i<n$, $\prod_{j=i}^{i}(1-p_{v_j})=1$. For $i=n$, $\prod_{j=i}^{i}(1-p_{v_j})=1-p_{v_n}$.

We then introduce F as follows

$$\begin{aligned} F^{\pi}(x_k)=&\prod_{i=k}^{n}(1-\tilde{p}_{v_i}(C_{[k:i-1]}))\beta_{v_n}+\sum_{i=k}^{n-1}\left[t_{v_i,v_{i+1}}\prod_{j=k}^{i}(1-\tilde{p}_{v_j}(C_{[k:j-1]}))\right]\\ &+\sum_{i=k}^{n}\left[\gamma_{v_i}\tilde{p}_{v_i}(C_{[k:i-1]})\prod_{j=k}^{i-1}(1-\tilde{p}_{v_j}(C_{[k:i-1]}))\right]. \end{aligned} \tag{A.1}$$

We notice that $F^{\pi}(x_n)=(1-p_{v_n})\beta_{v_n}+p_{v_n}\gamma_{v_n}=V^{\pi}(x_n)$.

We now show that F fulfills the recursive definition of the policy specific cost function and by recursion that $F=V$. From state $x_k=(C_{[0,k]},0)$, we let the cost for being in the next state $x_{k+1}=(C_{[0,k+1]},0)$ and seek to express $F^{\pi}(x_k)$ as a function of $F^{\pi}(x_{k+1})$ to fulfill the recursive Definition 3.5.

$$\begin{aligned} F^{\pi}(x_{k+1})=&\prod_{j=k+1}^{n}(1-p_{v_j})(\beta_{v_n})+\sum_{i=k+1}^{n-1}[t_{v_i,v_{i+1}}\prod_{j=k+1}^{i}(1-p_{v_j})]\\ &+\sum_{i=k+1}^{n}[\gamma_{v_i}p_{v_i}\prod_{j=k+1}^{i-1}(1-p_{v_j})] \end{aligned} \tag{A.2}$$

$$\begin{aligned} F^{\pi}(x_k)=&\ (1-p_{v_k})\prod_{j=k+1}^{n}(1-p_{v_j})(\beta_{v_n})+t_{v_k,v_{k+1}}+(1-p_k)\sum_{i=k+1}^{n-1}[t_{v_i,v_{i+1}}\prod_{j=k+1}^{i}(1-p_{v_j})]\\ &+p_{v_k}\gamma_{v_k}+(1-p_{v_k})\sum_{i=k}^{n}[\gamma_{v_i}p_{v_i}\prod_{j=k}^{i-1}(1-p_{v_j})]\\ F^{\pi}(x_k)=&\ t_{v_k,v_{k+1}}+(1-p_{v_k})F^{\pi}(x_{k+1})+p_{v_k}\gamma_{v_k} \end{aligned} \tag{A.3}$$

Accordingly, F fulfills the recursive Definition 3.5 for $w=0$, which concludes the proof. □

Proof of Proposition 2. We consider two simple search sequences (v) and (v,v') extended with the same visit sequence $C=(v_0,...,v_n)$. Let $C'=(v)\circ C$, resp. $C''=(v,v')\circ C$, associated to policies π', resp. π''. Let $t_{v,v_0}=t_{v,v'}+t_{v',v_0}$ and let v' be a direct neighbor of v (i.e., there is no station v'' such that $t_{v,v'}=t_{v,v''}+t_{v'',v'}$) and let v_0 not be a direct neighbor.

We now show that from the considered state $x_0=((v),0)$, visiting v' before v_0 is always better than straightforwardly visiting v_0. We get

$$V^{\pi'}((v),0)=t_{v,v_0}+(1-\tilde{p}_{v_0})V^{\pi'}((v,v_0),0),$$

$$V^{\pi''}((v),0)=t_{v,v'}+(1-\tilde{p}_{v'})t_{v',v_0}+(1-\tilde{p}_{v'})(1-\tilde{p}_{v_0})V^{\pi''}((v,v',v_0),0)$$

and distinguish two cases:

Case 1 ($v'\notin C$): In this case, the valuation of any unexplored station after v_0 does not depend on preceding visits in the respective sequence, i.e., $V^{\pi''}((v,v_0),0)=V^{\pi''}((v,v',v_0),0)$. Given that $(1-\tilde{p}_v)\leq 1$, we straightforwardly obtain $V^{\pi''}((v),0)\leq V^{\pi'}((v),0)$.

Case 2 ($v' \in C$): In this case, the valuation of any unexplored station after v_0 depends on preceding visits in the respective sequence and we obtain the following cost expansions, visiting v' at position k:

For path C', v' is visited for the first time such that $\tilde{p}_{v_k} = p_{v'}$ and we get

$$V^{\pi'}(C',0) = \prod_{j=0}^{n}(1-\tilde{p}_{v_j})(\overline{\beta}) + \sum_{i=0}^{k-1} t_{v_i,v_{i+1}} \prod_{j=0}^{i}(1-\tilde{p}_{v_j}) + \sum_{i=k}^{n-1}[t_{v_i,v_{i+1}}(1-\tilde{p}_{v'}) \prod_{j=0,j\neq k}^{i}(1-\tilde{p}_{v_j}) \quad \text{(A.4)}$$

For path C'', v' is visited for the second time such that $\tilde{p}_{v_k} = 0$ and we get

$$(1-\tilde{p}_{v'})V^{\pi''}(C'',0) = (1-\tilde{p}_{v'}) \prod_{j=0}^{n}(1-p_{v_j})(\overline{\beta}) + (1-\tilde{p}_{v'}) \sum_{i=0}^{k-1}[t_{v_i,v_{i+1}} \prod_{j=0}^{i}(1-p_{v_j}) + \sum_{i=k}^{n-1}[t_{v_i,v_{i+1}}(1-\tilde{p}_{v'}) \prod_{j=0,j\neq k}^{i}(1-p_{v_j}) \quad \text{(A.5)}$$

Since $(1-\tilde{p}_{v'}) \leq 1$, we have

$$(1-\tilde{p}_{v'})V^{\pi''}((v,v',v_0),0) \leq V^{\pi'}((v,v_0),0)$$

and consequently

$$V^{\pi''}(x_0) \leq V^{\pi'}(x_0).$$

In both cases, π'' is preferred over π' (thus C'' over C'), such that candidate stations can be restricted to neighbor stations only, which concludes the proof.

□

Appendix B: Problem Complexity

Proposition 3. *The SCPS problem is NP-hard, even with metric travel times.*

Proof. We show hardness through reduction from the traveling salesman problem (TSP) with metric and integer travel times. The decision problem variant of the TSP can be defined as follows: we consider a set $\mathcal{V}$ of n sites and travel times $t_{v,v'} \in \mathbb{N}$ between these. Travel times are bounded $1 \leq t_{v,v'} \leq \Delta$ for all $v, v' \in \mathcal{V}$. We are asked for a tour (i.e., a Hamiltonian path) $v_0, \ldots, v_n = v_0$ whose length satisfies $\sum_{i=0}^{n-1} t_{v_i,v_{i+1}} \leq \theta$ for a given θ. Given that travel times are integer, we can assume w.l.o.g. that $\theta \in \mathbb{N}$. Further, we assume that triangle inequality holds, i.e., $t_{v,v'} + t_{v',v''} \geq t_{v,v''}$ for all $v, v', v'' \in \mathcal{V}$. We note that hardness for this restricted metric case implies hardness for the generic case as well.

Step 1: We construct an instance for the station search from the TSP instance as follows: We select an arbitrary vertex $s \in \mathcal{V}$ and designate it as start vertex $v_0 := s$ for the search. We then create a duplicate s' in the same location ($t_{s,s'} = 0$), which serves as the termination vertex. Let q be an arbitrary value satisfying

$$\left(1 - \frac{1}{\Delta(n+1)}\right)^{\frac{1}{n-1}} \leq q < 1\,. \quad \text{(B.1)}$$

We parameterize the search as follows: All vertices c have an availability of $p_i := 1-q$ without recovery. There is no penalty for successful charging ($\gamma_i := 0\ \forall i$). For unsuccessfully terminating at s', the penalty is

$$\beta_{s'} := \frac{2\Delta}{q(1-q)} + 1\,. \quad \text{(B.2)}$$

For all other vertices $c \neq s'$ the penalty is

$$\beta_v := \frac{1}{q^{n+1}}(\beta_{s'} + n\Delta) + 1\,. \tag{B.3}$$

Now for any search path $C = (v_0, \ldots, v_k)$ that does not visit any vertex multiple times, it holds that

$$\begin{aligned}\alpha(C) &= \left(\prod_{i=0}^{k}(1-p_{v_i})\right)\beta_{v_k} + \sum_{i=0}^{k-1} t_{v_i,v_{i+1}} \prod_{j=0}^{i}(1-p_{v_j}) + \sum_{i=0}^{k} \gamma_{v_i} p_{v_i} \prod_{j=0}^{i-1}(1-p_{v_j}) \\ &= q^{k+1}\beta_{v_k} + \sum_{i=0}^{k-1} q^{i+1} t_{v_i,v_{i+1}}\,.\end{aligned} \tag{B.4}$$

Step 2: We now claim that the TSP instance possesses a solution with cost at most $\theta \in \mathbb{N}$ if, and only if, the station search admits a search path C with $\alpha(C) \leq q\theta + q^{n+1}\beta_{s'}$. This is done by transforming solutions between the two problems and carefully mapping their objective values.

For the first direction, we assume that a TSP tour is given. We convert it to a search path $C = (v_0, \ldots, v_n)$ by cutting at s such that $v_0 = s$ and $v_n = s'$. Then

$$\alpha(C) = q^{n+1}\beta_{s'} + \sum_{i=0}^{n-1} q^{i+1} t_{v_i,v_{i+1}} \leq q^{n+1}\beta_{s'} + q\sum_{i=0}^{n-1} t_{v_i,v_{i+1}} \leq q^{n+1}\beta_{s'} + q\theta\,.$$

Vice versa, we assume that we are given a search path P with $\alpha(P) \leq q\theta + q^{n+1}\beta_{s'}$. Then for an *optimal* search path $C = (v_0, \ldots, v_k)$ it holds that $\alpha(P) \leq \alpha(C) \leq q\theta + q^{n+1}\beta_{s'}$. Given metric travel times and no recovery, we can assume that C does not visit any vertex more than once. We construct a tour through the following observations:

1. C visits s': Assume it does not. Let C' be C extended by ending at s'. Then

$$\begin{aligned}\alpha(C') &= \alpha(C) - q^{k+1}\beta_{v_k} + q^{k+2}\beta_{s'} + q^{k+1}t_{v_k,s'} \\ &= \alpha(C) - q^{k+1}(\beta_{v_k} - q\beta_{s'} - t_{v_k,s'}) \overset{(*)}{<} \alpha(C)\end{aligned} \tag{B.5}$$

   using in (*) that $\beta_{v_k} > q\beta_{s'} + \Delta$ by (B.3). This contradicts the optimality of C.

2. C visits s' last: Assume it does not. We obtain C' from C by moving s' to the end. Then it holds that

$$\begin{aligned}&\alpha(C) \geq q^{k+1}\beta_{v_k} \geq q^{n+1}\beta_{v_k} \text{ and} \\ &\alpha(C') \leq q^{k+1}\beta_{s'} + \sum_{i=0}^{k-1} q^{i+1}\Delta \leq \beta_{s'} + n\Delta\,.\end{aligned} \tag{B.6}$$

   Given that $q^{n+1}\beta_{v_k} > \beta_{s'} + n\Delta$ (B.3), it holds that $\alpha(C) > \alpha(C')$, contradicting optimality.

3. C visits every vertex: Assume $C = (v_0, \ldots, v_{k-1}, s')$ omits some vertex v'. Let $C' = (v_0, \ldots, v_{k-1}, v', s')$. Then

$$\begin{aligned}\alpha(C) - \alpha(C') &= \left(q^k t_{v_{k-1},s} + q^{k+1}\beta_{s'}\right) - \left(q^k t_{v_{k-1},v'} + q^{k+1}t_{v',s'} + q^{k+2}\beta_{s'}\right) \\ &= q^k(t_{v_{k-1},s} - t_{v_{k-1},v'} - qt_{v',s'} + (q-q^2)\beta_{s'}) \\ &\geq q^k(-2\Delta + q(1-q)\beta_{s'}) \overset{\text{(B.2)}}{>} 0\,,\end{aligned} \tag{B.7}$$

   again contradicting optimality.

4. It is clear now that C corresponds to a TSP tour, by identifying s with s'. It holds that

$$\alpha(C) = \sum_{i=0}^{n-1} q^{i+1} t_{v_i,v_{i+1}} + q^{n+1}\beta_{s'} \le q\theta + q^{n+1}\beta_{s'} \,. \tag{B.8}$$

Assume the tour would violate the threshold, i.e., $\sum_{i=0}^{n-1} t_{v_i,v_{i+1}} > \theta$. Given integrality, the length then is at least $\theta+1$. It follows that

$$\begin{aligned}\sum_{i=0}^{n-1} q^{i+1} t_{v_i,v_{i+1}} &\ge \sum_{i=0}^{n-1} q^{n} t_{v_i,v_{i+1}} = q\sum_{i=0}^{n-1} q^{n-1} t_{v_i,v_{i+1}} \\ &\overset{\text{(B.1)}}{\ge} q\sum_{i=0}^{n-1}(1-\frac{1}{\Delta(n+1)})t_{v_i,v_{i+1}} = q\left[\sum_{i=0}^{n-1} t_{v_i,v_{i+1}} - \sum_{i=0}^{n-1}\frac{t_{v_i,v_{i+1}}}{\Delta(n+1)}\right] \\ &\ge q\left[\theta+1-\sum_{i=0}^{n-1}\frac{1}{n+1}\right] > q\theta \,.\end{aligned} \tag{B.9}$$

This contradicts (B.8), thereby proving that $\sum_{i=0}^{n-1} t_{v_i,v_{i+1}} \le \theta$.

□

Appendix C: Variant-specific methods

We now present the required methodological changes for the problem variants $\neg W/\neg C$, $W/\neg C$ and W/C. Besides describing the cost structure variants (Section C.1), we detail the necessary modifications for the labeling and the rollout algorithms (Section C.2).

C.1. Cost structure variants

For $\neg W/\neg C$, $W/\neg C$ and W/C problem variant as introduced in Section 2.1.1, the cost functions $V^\pi(C,0)$, $V^\pi(C,1)$ and $V^\pi(x_n)$ for a termination state $x_n = (C,a)$, can be expressed as follows, with C' denoting the extension of C by station v_{k+1} for non-termination states.

No waiting, without charging cost ($\neg W/\neg C$)

$$\begin{aligned} V^\pi(C,0) &= t_{v_k,v_{k+1}} + (1-\tilde{p}_{v_{k+1}})V^\pi(C',0)\,, \\ V^\pi(C,1) &= 0\,, \\ V^\pi(x_n) &= (1-\tilde{p}_{v_n})\overline{\beta}\,. \end{aligned} \tag{C.1}$$

Waiting permitted, without charging cost ($W/\neg C$)

$$\begin{aligned} V^\pi(C,0) &= wW_{v_k} + (1-w)[t_{v_k,v_{k+1}} + (1-\tilde{p}_{v_{k+1}})V^\pi(C',0)]\,, \\ V^\pi(C,1) &= 0\,, \\ V^\pi(x_n) &= (1-\tilde{p}_{v_n})W_{v_n}\,. \end{aligned} \tag{C.2}$$

Waiting permitted, with heterogeneous charging costs (W/C)

$$\begin{aligned} V^\pi(C,0) &= w(Y_{v_k} + W_{v_k}) + (1-w)[t_{v_k,v_{k+1}} + (1-\tilde{p}_{v_{k+1}})V^\pi(C',0) + \tilde{p}_{v_{k+1}}L_{v_{k+1}}]\,, \\ V^\pi(C,1) &= Y_{v_k}\,, \\ V^\pi(x_n) &= (1-\tilde{p}_{v_n})W_{v_n} + Y_{v_n}\,. \end{aligned} \tag{C.3}$$

Figure 11 Dynamic programming based labeling algorithm.

1: $\mathcal{L}^a \leftarrow \{L_0\}$, $L^* \leftarrow L_0$
2: **while** $\mathcal{L}^a \neq \emptyset$ **do**
3:     $L \leftarrow$ `costMinimumLabel(`$\mathcal{L}^a$`)`
4:     $\mathcal{L}^a \leftarrow \mathcal{L}^a \setminus \{L\}$
5:     **for** $(v, v') \in \delta^+(L)$ **do**
6:         $L' \leftarrow \mathcal{F}_{ij}(L)$
7:         **if** `isNotDominated(`$L', \mathcal{L}^a$`)` **then**
8:             `dominanceCheck(`$\mathcal{L}^a, L'$`)`
9:             $\mathcal{L}^a \leftarrow \mathcal{L}^a \cup \{L'\}$
10:             **if** $\alpha(L') < \alpha(L^*)$ **then**
11:                 $L^* \leftarrow L'$
12: **return** L^*

C.2. Algorithm variants

The main methodology section discusses the algorithm implementation for problem variant $\neg W/C$, which can be applied to the problem variant $\neg W/\neg C$ without any further changes, because both variants ignore the waiting decision. In the following, we specify the required changes to account for waiting times and waiting decisions that can be applied to both problem variants $W/\neg C$ and W/C.

C.2.1. Labeling algorithm Similar to the changes required to account for the time-dependent recovery functions detailed in Section 3.4, we now introduce the resource R_v in the label definition (cf. Equation 3.27), and define a label as follows: $L_v = (t_v, A_v, \rho_v, \alpha_v, S_v, R_v)$.

For problem variants $W/\neg C$ and W/C, we recall that a search can terminate before the whole time budget is spent and that a station terminating the search might have been visited earlier. Accordingly we drop Condition 3.21 and use Condition 3.28 (defined in Section 3.4) in the dominance criterion that checks whether all visited vertices reachable by π_2 can be reached from π_1 as well. With L_1 and L_2 being the associated labels to π_1 and π_2, we then say that $L_1 \succ L_2$ if Conditions (3.17)–(3.20) are true and Condition (3.28) holds.

Here, each newly created label can also serve as a termination label, as the driver can decide to wait to terminate the search. Figure 11 shows the adapted pseudo-code (l.10), where we relax the termination condition stating that no feasible successors should be left.

C.2.2. Rollout algorithm For problem variants $W/\neg C$ and W/C, we use an additional procedure `refinePolicy(C)` that bases on the resulting visits sequence C to introduce the waiting decision at the best stage $1 \leq k \leq n$, with n being the total amount of station visits in C. Figure 12 shows the adapted pseudo-code, where we account for the waiting decision.

We first calculate the no-waiting case and compute policy π with the associated ordered sequence of charging stations $C = (v_0, ..., v_n)$. We denote π as an intermediate policy and introduce π_S representing the final search policy. Then, `refinePolicy(C)` calculates π_S using the intermediate policy π while permitting wait-actions (l.12). For each intermediary charging station v_k at the k^{th} decision epoch, $v_k \in (v_0, ..., v_n), k \neq n$, π provides a sub sequence of charging stations to visit until the end of the search $(v_{k+1}, ..., v_n)$ and thus the policy-specific cost value $V^\pi(x_k)$, associated to policy π and state $x_k = ((v_0, ..., v_k), 0)$. We aim to quantify for each station v_k whether the termination cost β_{v_k} is actually lower than the expected cost for continuing

Figure 12 **Forward programming based algorithm.**

$v_k \leftarrow v_0$, $C \leftarrow (v_0)$, $x_k \leftarrow (C,0)$, $t \leftarrow 0$
while $t \leq \overline{T}$ **do**
  $v^* \leftarrow 0$, $x^* \leftarrow 0$, $C^* \leftarrow 0$, $Q \leftarrow \infty$
  for $(v,0) \in \mathcal{U}(x_k)$ **do**
    $x_{k+1} \leftarrow (C',0)$
    $V \leftarrow \texttt{greedyCost}(x_{k+1})$
    $Q(x_k, v, x_{k+1}) \leftarrow t_{v_k,v} + (1-\tilde{p}_v)V + \tilde{p}_v\gamma_v$
    if $Q(x_k, v, x_{k+1}) < Q$ **then**
      $Q \leftarrow Q(x_k, v, x_{k+1})$
      $v^* \leftarrow v$, $x^* \leftarrow x$, $C^* \leftarrow C'$
  $C \leftarrow C^*$, $x_k \leftarrow x^*$, $t \leftarrow t + t_{v_k,v^*}$, $v_k \leftarrow v^*$
$C \leftarrow \texttt{refinePolicy}(C)$
return C

the search $V^\pi(x_k)$. If this is the case, the optimal decision is to wait and we refine π into π_S with $\pi_S(x_k) = (v_k, w=1)$ and $V^{\pi_S}(x_l) \leq V^\pi(x_l)\ \forall l \in [0,n]$.

We define π_S as

$$\pi_S(x_k) = \underset{\pi(x_k),(v_k,w=1)}{\arg\min}\ w\beta_{v_k} + (1-w)[t_{v_k,v_{k+1}} + (1-\tilde{p}_{v_{k+1}})\min(V^\pi(x_{k+1}), V^{\pi_S}(x_{k+1})) + \tilde{p}_{v_{k+1}}\gamma_{v_{k+1}}], \tag{C.4}$$

where $\pi(x_k) = (v_{k+1}, w=0)$. If there exists an index k such that $0 \leq k < n$, $x_k = ((v_0,...,v_k),0)$ and $\pi_S(x_k) = (v_k,1)$, then state x_k terminates the search, as the driver will wait at v_k if v_k is not immediately available. In this case π_S encodes the solution $(v_0,...,v_k)$.

C.2.3. Integrating search related energy consumption Charge times become path-sensitive for problem variant W/C, similar to problem variant $\neg W/C$, whereas $\neg W/\neg C$ and $W/\neg C$ do not account for path-dependent charging times based on their definitions.

Appendix D: Reduced action spaces results

Table 11 compares the computational times, and the percentage share of instances that can be computed in less than 15.000 seconds by each heuristic. Further, for the instances that can be solved to optimality within 15.000 seconds, the table compares for both heuristics the averaged optimality gap and computational time gap.

Preliminary results show that using the *complete* action space for problem variants $\neg W/\neg C$ and $W/\neg C$ with LH is computationally too heavy to be of any practical use, such that we restrict results to the other restricted action spaces. As can be seen, while *direct* only slightly helps saving computational times on average for $\neg W/\neg C$ and LH, it allows to solve 6% more instances within the allocated time for $W/\neg C$. For RO, results show a 96% decrease of the computational time for $W/\neg C$. For $\neg W/C$, restricting the next station visits from the current location to the ones accessible in less than five minutes allows to save 36% of the computational times compared to *complete*. Accordingly, Table 12 shows the most appropriate action space for each heuristic and problem variant.

Appendix E: Problem variant-specific results

In the following, we discuss the results for problem variants $\neg W/\neg C$, $W/\neg C$ and W/C of the applicability study (Section E.1), the computational tractability study (Section E.2) and the extended analysis (Section E.3).

E.1. Applicability

We use the same metrics to evaluate the performances of our algorithms for problem variant $\neg W/\neg C$. Since the search always ends successfully in waiting variants $W/\neg C$ and W/C, we analyze the algorithms based on their maximum realized search cost. Similarly to the simulated search cost evaluation, we use the maximum realized search cost deviation, that results to $\Delta\widehat{\alpha}^{\max} = \widehat{\alpha}^{\max}/\widehat{\alpha}^{\max^*}$ with $\widehat{\alpha}^{\max} = \max \widehat{\alpha}_i$ being the maximum cost out of all simulation runs for a single algorithm and with $\widehat{\alpha}^{\max^*}$ being the maximum cost over all simulation runs and all algorithms. Note that the realized search cost corresponds to the realized driving time needed to find a station for variants $\neg W/\neg C$ and $W/\neg C$, and to complete charging for W/C. For W variants, $\hat{\alpha}$ includes the waiting time whenever the last visited station is occupied.

Table 13 details the average realized search cost deviations between all algorithms for problem variants $\neg W/\neg C$, $W/\neg C$ and W/C while Table 14 details the realized success rate of each algorithm for $\neg W/\neg C$.

Table 11 Aggregated computational results over all tested instances for each problem variant and used action space

Graph setup		LH				RO			
Problem variant	Action space	$\hat{\Delta}(t)$	$\hat{\Delta}(\alpha)$	$\hat{t}$	$\hat{n}$	$\hat{\Delta}(t)$	$\hat{\Delta}(\alpha)$	$\hat{t}$	$\hat{n}$
$\neg W/\neg C$	*direct*	-0.79	0.01	197	0.73	-0.65	0.27	0.08	1.00
	direct/restricted	-0.84	0.04	235	0.82	–	–	–	–
	T^r*-restricted*	-0.83	0.00	221	0.74	–	–	–	–
	complete	–	–	–	–	-0.77	0.13	0.58	1.00
$W/\neg C$	*direct*	-0.80	0.00	72.7	0.83	-0.80	0.03	0.03	1.00
	direct/restricted	-0.84	0.02	16.7	1.00	–	–	–	–
	T^r*-restricted*	-0.84	0.01	250	0.78	–	–	–	–
	complete	–	–	–	–	-0.64	0.04	0.76	1.00
$\neg W/C$	T^r*-restricted*	-0.83	0.01	311	0.84	-0.82	0.12	0.31	1.00
	complete	-0.84	0.01	486	0.84	-0.81	0.12	0.38	1.00
W/C	T^r*-restricted*	-0.74	0.00	120	0.83	-0.69	0.00	0.88	1.00
	complete	-0.80	0.00	100	0.83	-0.66	0.00	1.05	1.00

Abbreviations hold as follows: $\hat{\Delta}(t)$ - averaged computational time gap with $\Delta(t) = \frac{t^{\text{heur}} - t^{\text{opt}}}{t^{\text{opt}}}$ [%], $\hat{\Delta}(\alpha)$ - averaged optimality gap over all tested instances with $\Delta(\alpha) = \frac{\alpha^{\text{heur}} - \alpha^{\text{opt}}}{\alpha^{\text{opt}}}$ [%], $\hat{t}$ - averaged computational time [s], $\hat{n}$ - rate of instances that can be computed in less than 15.000 seconds.

Table 12 Best action space /heuristic combination per problem variant

Problem variant	LH	RO
$\neg W/\neg C$	*direct*	*complete*
$W/\neg C$	*direct*	*direct*
$\neg W/C$	T^r*-restricted*	*complete*
W/C	*complete*	*complete*

For all problem variants, the table shows for each heuristic (LH, RO) the graph setting to that provides the best trade-off between computational times and solution quality.

Table 13 Average search cost deviations for $\neg W/\neg C$, $W/\neg C$ and W/C

		low-15%				*avg-60%*				*high-90%*			
		N	G	RO	LH	N	G	RO	LH	N	G	RO	LH
$\neg W/\neg C$	*SF-1*/1	1.38	0.78	**0.00**	0.77	2.63	0.41	**0.00**	**0.00**	1.33	0.35	0.19	**0.00**
	SF-1/2	1.13	0.59	**0.09**	0.14	2.66	2.55	**0.00**	**0.00**	2.52	1.08	**0.00**	**0.00**
	BER-1/1	0.08	**0.02**	**0.02**	0.08	0.39	0.39	**0.25**	0.27	0.28	0.98	**0.00**	**0.00**
	BER-1/2	0.18	0.08	**0.01**	**0.01**	3.38	0.16	**0.03**	**0.03**	4.14	0.24	**0.00**	**0.00**
	SF-2/1	4.74	4.72	0.57	**0.28**	5.98	5.98	**0.01**	**0.01**	6.51	3.18	**0.01**	**0.01**
	SF-2/2	8.31	6.52	0.25	**0.16**	10.7	6.16	0.17	**0.10**	6.43	6.43	**0.01**	**0.01**
$W/\neg C$	*SF-1*/1	5.67	6.61	6.68	**0.60**	2.71	1.72	**0.01**	**0.01**	1.08	0.12	**0.00**	**0.00**
	SF-1/2	2.30	6.19	0.67	**0.41**	2.71	1.69	**0.01**	**0.01**	2.53	0.11	**0.00**	**0.00**
	BER-1/1	0.41	1.32	**0.20**	**0.20**	0.15	0.36	**0.04**	**0.04**	0.25	**0.00**	**0.00**	**0.00**
	BER-1/2	4.39	1.35	**0.20**	**0.20**	4.36	0.62	**0.24**	**0.24**	4.12	**0.00**	**0.00**	**0.00**
	SF-2/1	58.1	4.55	**0.42**	**0.42**	6.41	0.55	**0.01**	**0.01**	6.52	0.04	**0.01**	**0.01**
	SF-2/2	33.1	4.72	**0.19**	**0.19**	10.8	0.73	0.17	**0.09**	6.40	0.05	**0.01**	0.05
W/C	*SF-1*/1	2.37	1.17	**0.00**	**0.00**	1.14	0.04	**0.00**	**0.00**	0.98	**0.00**	**0.00**	**0.00**
	SF-1/2	2.11	4.24	**0.01**	**0.01**	2.83	0.04	**0.00**	**0.00**	0.06	**0.00**	**0.00**	**0.00**
	BER-1/1	0.74	0.01	**0.00**	**0.00**	0.12	0.02	**0.00**	**0.00**	2.79	0.02	**0.00**	**0.00**
	BER-1/2	1.37	**0.01**	**0.01**	**0.01**	0.94	0.02	**0.00**	**0.00**	2.92	**0.00**	**0.00**	**0.00**
	SF-2/1	3.48	0.03	**0.01**	**0.01**	0.17	0.01	**0.00**	**0.00**	1.02	**0.00**	**0.00**	**0.00**
	SF-2/2	0.89	0.04	**0.02**	**0.02**	0.98	0.01	**0.00**	**0.00**	1.02	**0.00**	**0.00**	**0.00**

The table compares the average search cost deviation $\Delta\widehat{\alpha}$ for LH, RO, G, and N for each instance of problem variants $\neg W/\neg C$, $W/\neg C$, and W/C.

Table 14 Success rate for $\neg W/\neg C$

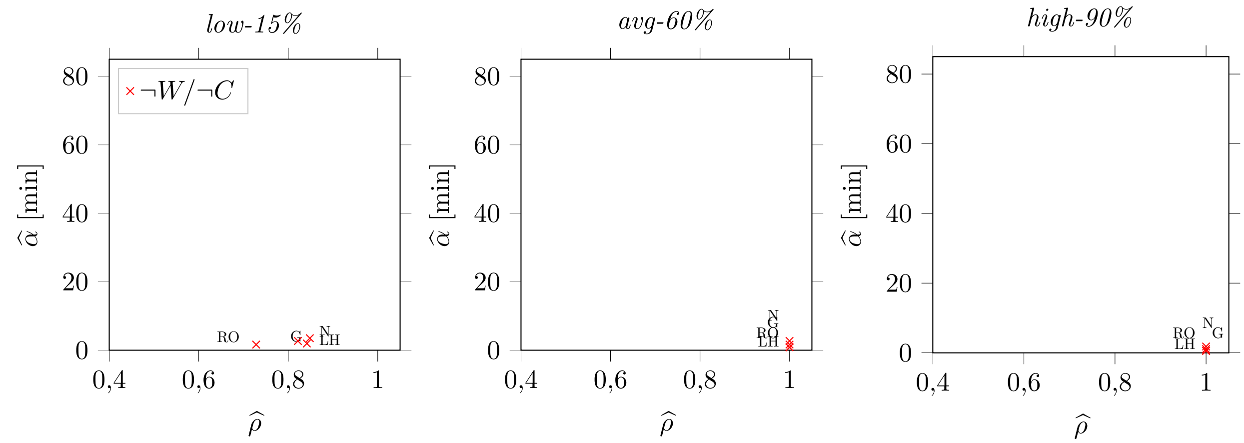
		low-15%				*avg-60%*				*high-90%*			
		N	G	RO	LH	N	G	RO	LH	N	G	RO	LH
$\neg W/\neg C$	*SF-1*/1	0.81	0.85	0.36	0.86	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	SF-1/2	0.95	0.85	0.81	0.85	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	BER-1/1	0.78	0.77	0.77	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	BER-1/2	0.95	0.80	0.78	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	SF-2/1	0.70	0.80	0.78	0.86	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	SF-2/2	0.90	0.86	0.87	0.88	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

The table compares the success rate $\widehat{\rho}$ for LH, RO, G, and N for each instance of the $\neg W/\neg C$ problem variant.

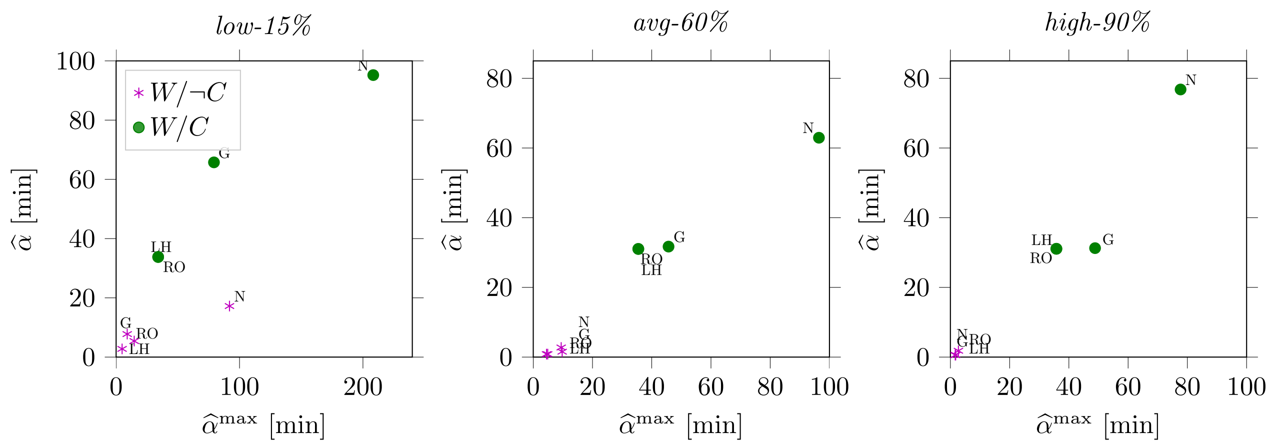
Table 15 Maximum realized search cost deviation for $W/\neg C$ and W/C

		low-15%				*avg-60%*				*high-90%*			
		N	G	RO	LH	N	G	RO	LH	N	G	RO	LH
$W/\neg C$	*SF-1*/1	22.6	2.29	10.8	**0.00**	1.42	**0.00**	0.10	0.24	2.11	**0.00**	0.13	0.13
	SF-1/2	4.45	2.29	**0.00**	**0.00**	2.01	0.11	**0.00**	**0.00**	0.69	**0.00**	0.13	0.13
	BER-1/1	1.12	**0.00**	0.32	0.32	0.28	2.08	**0.00**	**0.00**	0.27	0.34	**0.00**	**0.00**
	BER-1/2	33.7	**0.00**	0.32	0.32	0.60	2.08	**0.00**	**0.00**	0.71	0.34	**0.00**	**0.00**
	SF-2/1	29.0	0.27	**0.00**	**0.00**	0.89	1.05	**0.00**	**0.00**	0.74	**0.00**	0.70	0.70
	SF-2/2	30.2	0.27	**0.00**	**0.00**	3.60	2.04	**0.00**	1.07	0.02	0.55	**0.00**	0.55
W/C	*SF-1*/1	6.07	1.22	**0.00**	**0.00**	2.60	**0.00**	0.01	0.01	0.85	**0.00**	**0.00**	**0.00**
	SF-1/2	2.78	6.24	**0.00**	**0.00**	2.61	0.01	**0.00**	**0.00**	**0.00**	0.03	0.03	0.03
	BER-1/1	2.69	**0.00**	**0.00**	**0.00**	0.73	0.36	**0.00**	**0.00**	1.63	1.70	**0.00**	**0.00**
	BER-1/2	6.41	**0.00**	**0.00**	**0.00**	0.96	1.38	**0.00**	**0.00**	2.39	0.01	**0.00**	**0.00**
	SF-2/1	6.34	0.06	**0.00**	**0.00**	2.57	**0.00**	**0.00**	**0.00**	0.99	**0.00**	0.01	0.01
	SF-2/2	6.49	0.20	**0.00**	**0.00**	0.85	**0.00**	0.02	0.01	0.95	**0.00**	0.01	**0.00**

The table compares the average maximal search cost deviation $\Delta\widehat{\alpha}^{\max}$ for LH, RO, G, and N for each instance of problem variants $W/\neg C$ and W/C.



(a) Averaged search cost vs. success rate for problem variant $\neg W/\neg C$



(b) Averaged search cost vs. worst search cost for problem variants $W/\neg C$ and W/C

Figure 13 Trade off between the average search cost and the success rate, resp. the worst search cost

Figure 13 shows the trade-off between the average realized search costs and (i) the realized success rate, aggregated over all scenarios for problem variant $\neg W/\neg C$ (Figure 13a) or (ii) the maximum realized search cost, aggregated over all scenarios for W problem variants (Figure 13b).

As can be seen in Table 13, larger search cost deviations occur for the W variants, especially for $W/\neg C$, due to the heterogeneous penalty costs that result from waiting at an occupied station. Particularly, on the *SF-1* instances at low-availability, RO performs significantly worse compared to LH with respect to the average search time. In this case, a single varying station visit between two search strategies can cause such differences due to the limited amount of candidate stations and the large amplitude between penalty costs. For $\neg W/\neg C$, the LH algorithm shows higher success rates compared to the RO algorithm (cf. Table 14 and Figure 13a), which highlights the superiority of the LH algorithm compared to the RO algorithm as both algorithms show comparable performances with respect to the average search cost.

Table 15 shows the deviation between the maximum search cost of all algorithms for W problem variants, which are guaranteed to be feasible at the price of high waiting cost. Here, we observe a trade-off between the average search cost and the maximum search cost (cf. Figure 13b). As one can see, the LH and the RO algorithm show a similar performance for scenarios with medium and high charging station availability, but the LH algorithm performs better in the SF-1/1 scenario with low charging station availability. Across all scenarios the greedy algorithm sometimes yields the least maximum cost. However, this improvement stems from significantly increased search costs (cf. Table 13).

For W problem variants, we observe that the advanced algorithms outperform the myopic algorithms significantly, independent of the charging station availability. This performance difference is higher for problem variant $W/\neg C$. For problem variant W/C, the greedy algorithm performs close to the advanced algorithms for scenarios with medium or high charging station availability.

E.2. Computational tractability

Table 16 compares the performance of the LH and RO algorithm against the exact labeling algorithm for the remaining $\neg W/\neg C$, $W/\neg C$ and W/C problem variants. As can be seen, the observed differences are sensitive to the problem variant. Significant differences occur for the $\neg W/\neg C$ problem variant, similar to $\neg W/C$, with low charging station availability and varying search radii, while the algorithms perform similarly on the waiting problem variants. These high deviations result from penalty costs for unsuccessful searches, which can only result for $\neg W$ problem variants and are more likely to occur at low charging station availability.

Figures 14-16 show the extensive heuristic comparisons for the remaining problem variants $W/\neg C$, $\neg W/C$, W/C. Similarly to $\neg W/C$, LH significantly outperforms RO for both non charging variants in low availability scenarios, especially in low density areas, whereas there exists no significant difference for problem variant W/C.

Table 16 Aggregated computational results over all tested instances per scenario for $\neg W/\neg C$, $W/\neg C$ and W/C

			L-H				RO				L-E	
			$\hat{\Delta}(\alpha)$	$\hat{\Delta}(\bar{\alpha})$	$\hat{t}$	$\hat{n}$	$\hat{\Delta}(\alpha)$	$\hat{\Delta}(\bar{\alpha})$	$\hat{t}$	$\hat{n}$	$\hat{t}$	$\hat{n}$
$\neg W/\neg C$	*low-15%*	*SF-1*	0.01	0.01	1.78	100	0.32	0.38	0.09	100	164	64
		BER-1	0.04	0.04	0.49	100	0.43	0.46	0.28	100	1386	46
		SF-2	0.03	0.01	444	66	0.22	0.18	1.38	100	8646	4
	avg-60%	*SF-1*	**0.00**	0.00	266	96	0.04	0.04	0.11	100	7.02	64
		BER-1	0.00	-0.02	84.5	100	0.01	-0.02	0.23	100	473	59
		SF-2	**0.00**	0.00	803	34	0.11	0.10	1.76	100	4816	5
	high-90%	*SF-1*	**0.00**	0.00	107	66	0.03	0.03	0.10	100	16.4	64
		BER-1	**0.00**	0.00	66.7	66	**0.00**	0.00	0.35	100	517	57
		SF-2	**0.00**	0.00	1.70	25	**0.00**	0.00	0.95	100	4156	4
$W/\neg C$	*low-15%*	*SF-1*	0.04	0.05	0.05	100	0.09	0.10	0.01	100	482	68
		BER-1	**0.00**	0.00	1.60	100	**0.00**	0.00	0.04	100	283	64
		SF-2	**0.00**	0.00	160	64	0.03	0.04	0.10	100	2111	25
	avg-60%	*SF-1*	0.00	0.00	0.05	100	0.03	0.03	0.01	100	2.33	64
		BER-1	**0.00**	0.01	0.72	100	**0.00**	0.00	0.01	100	1046	64
		SF-2	**0.00**	0.00	491	54	0.09	0.09	0.03	100	4219	25
	high-90%	*SF-1*	**0.00**	0.00	0.06	100	0.00	0.00	0.01	100	1.68	64
		BER-1	**0.00**	0.00	0.38	100	**0.00**	0.00	0.01	100	475	64
		SF-2	**0.00**	0.00	0.43	25	**0.00**	0.00	0.05	100	2024	25
W/C	*low-15%*	*SF-1*	**0.00**	0.00	1.90	100	**0.00**	0.00	0.12	100	493	68
		BER-1	**0.00**	0.00	5.98	100	**0.00**	0.00	0.77	100	803	66
		SF-2	**0.00**	0.00	358	79	**0.00**	0.00	2.95	100	2565	25
	avg-60%	*SF-1*	**0.00**	0.00	0.25	100	**0.00**	0.00	0.12	100	472	68
		BER-1	**0.00**	0.00	0.83	100	0.00	0.00	0.57	100	495	64
		SF-2	**0.00**	0.00	527	64	**0.00**	0.00	2.59	100	2031	25
	high-90%	*SF-1*	**0.00**	0.00	1.99	89	**0.00**	0.00	0.08	100	352	68
		BER-1	**0.00**	0.00	0.05	89	**0.00**	0.00	0.34	100	629	66
		SF-2	**0.00**	0.00	4.27	29	**0.00**	0.00	1.88	100	2661	25

Abbreviations hold as follows: $\hat{\Delta}(\alpha)$ - averaged optimality gap [%], $\hat{\Delta}(\bar{\alpha})$ - averaged simulated estimate deviation [%], $\hat{t}$ - averaged computational time [s], $\hat{n}$ - rate of instances that can be computed in less than 15.000 seconds. We note that an average $\hat{\Delta}(\alpha)$ of 0.00 indicates that an algorithm (almost) always finds the optimal solution. If it always finds the optimal solution we highlight the respective $\hat{\Delta}(\alpha)$ in bold font, whereas we leave it in normal font if some solutions remain heuristic but are not reflected in the value of $\hat{\Delta}(\alpha)$ due to rounding.

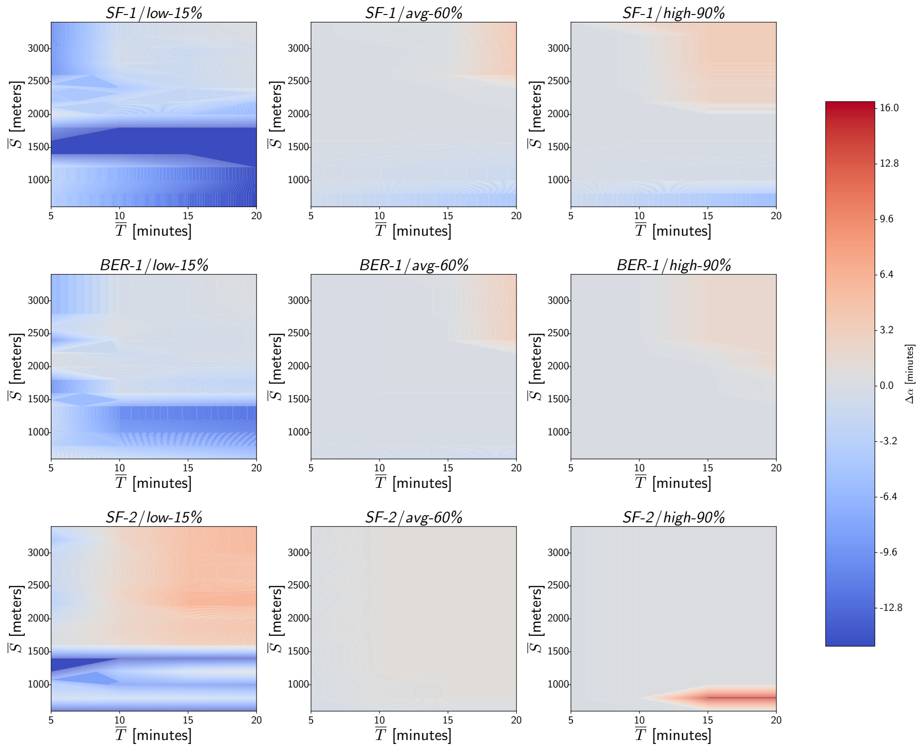


Figure 14 Extensive comparison of the LH and RO heuristics for problem variant $\neg W/\neg C$

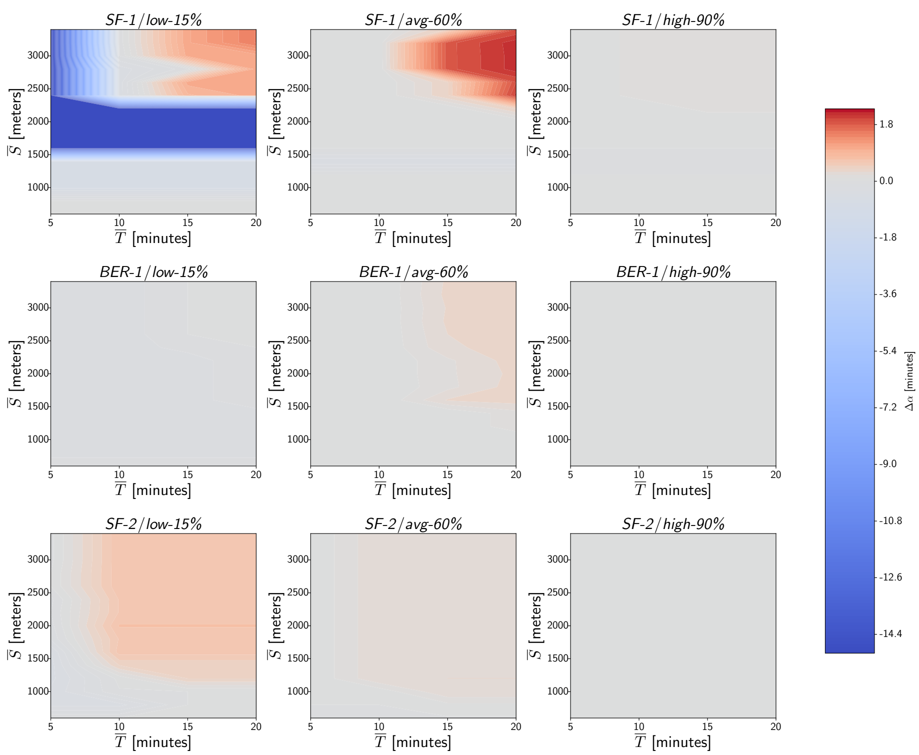


Figure 15 Extensive comparison of the LH and RO heuristics for problem variant $W/\neg C$

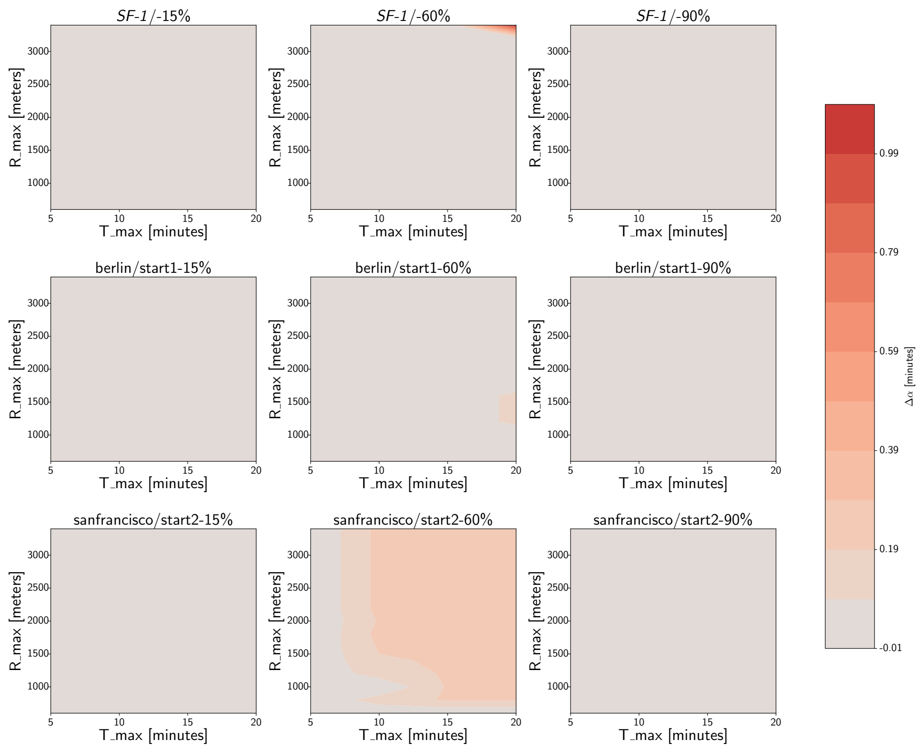


Figure 16 Extensive comparison of the LH and RO heuristics for problem variant W/C

E.3. Extended analysis

In the following, we discuss the algorithm's sensitivity toward $\bar{\beta}$ for problem variant $\neg W/\neg C$, the heuristic dominance criterion for problem variants $\neg W/\neg C$, $W/\neg C$ and W/C and the impact of time-dependent recovery functions.

Termination penalty

For problem variant $\neg W/\neg C$, t^s corresponds to the expected search time, due to the usage cost homogeneity ($\gamma_v = 0$, $\forall v \in \mathcal{V}$). As can be seen in Figure 17, a similar goal conflict as for problem variant $\neg W/C$, between expected search time and success rate, exists. Note that a lower $\bar{\beta}$ value is needed to obtain best success rate solutions (120 minutes).

Time-dependent recovery function

Results show no significant impact of the time-dependent recovery functions for the problem variants $W/\neg C$ and W/C, such that we only report values for problem variant $\neg W/\neg C$ in Table 17. As can be seen, results yield similar insights as for problem variant $\neg W/C$: time-dependent recovery functions mainly impact results in scenarios with smaller search areas, large time budget, and low charging station availability.

Relaxed dominance criteria

Figure 18 shows the trade-off between the optimality gap and the computational times for all dominance criteria for problem variants $\neg W/\neg C$, $W/\neg C$ and W/C. Similarly to $\neg W/C$, the (heuristic) dominance criterion as chosen in Section 3 – (1,1,0,0,0) – yields the lowest computational times possible to achieve

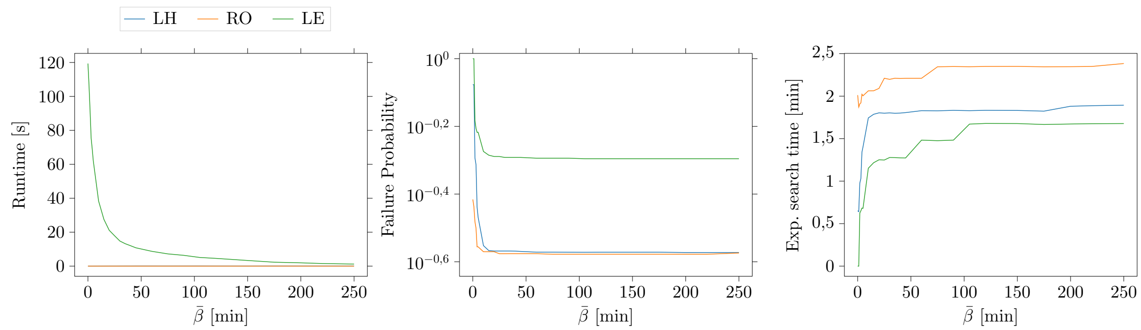


Figure 17 Impact of $\bar{\beta}$ on averaged computational time, expected search time and failure rate for the low-15% instances for $\neg W/\neg C$

Note. For LE, data are averaged over 6 instances corresponding to two instances ($\bar{S} = 800/\bar{T} = 5$ and $\bar{S} = 1.000,\bar{T} = 10$) per spatial scenario (*SF-1*, *BER-1* and *SF-2*) and for LH and RO over 9 instances corresponding to three instances ($\bar{S} = 1.000/\bar{T} = 5$, $\bar{S} = 1.500,\bar{T} = 10$ and $\bar{S} = 2.000,\bar{T} = 15$) per spatial scenario.

Table 17 Potential solution improvement for the time-dependent probability recovery function for problem variant $\neg W/\neg C$

		low-15%				*avg-60%*			
		LH		RO		LH		RO	
$\overline{T}$	$\overline{S}$	α^{ref}	α^{new}	α^{ref}	α^{new}	α^{ref}	α^{new}	α^{ref}	α^{new}
5	800	25.8	25.7	28.6	33.1	1.41	1.53	1.55	1.81
5	2000	13.8	13.85	13.9	13.8	1.32	1.32	1.34	1.34
5	3400	9.63	9.62	16.3	16.27	1.32	1.32	1.34	1.34
10	800	**23.9**	**21.1**	28.6	25.9	1.35	1.33	1.4	1.6
10	2000	6.84	6.84	7.32	11.8	1.23	1.23	1.23	1.23
10	3400	3.26	3.26	3.44	3.44	1.23	1.23	1.23	1.23
15	800	**23.9**	**18.1**	**28.6**	**24.4**	1.35	1.31	1.4	1.59
15	2000	4.25	4.28	5.05	6.14	1.23	1.23	1.23	1.23
15	3400	2.45	2.58	2.53	2.53	1.23	3.34	1.23	1.23
20	800	**23.9**	**16.3**	**28.6**	**23.4**	1.35	1.3	1.4	1.58
20	2000	3.51	3.51	4.46	4.15	1.23	1.33	1.23	1.23
20	3400	2.41	3.13	2.50	2.50	2.02	3.37	1.23	1.23

The table compares for *BER-1* combined with *low-15%* and *avg-60%* the objective value obtained in the updated setting (α^{new}) and the initial setting (α^{ref}). The table excludes *high-90%* results as these do not show any deviations. Significant differences are shown in bold characters.

the best possible solution quality among all heuristic dominance criteria for problem variants $\neg W/\neg C$ and $W/\neg C$. For problem variant W/C, (1,1,0,0,1) yields the best trade-off by slightly decreasing computational times obtained with (1,1,0,0,0), but selecting (1,1,0,0,0) allows the best possible generic implementation for LH.

Appendix F: Objective value decomposition

In this section, we show by proving Proposition 4 that the objective value $\alpha(\pi)$ as derived in Section 2 can be expressed based on $t(\pi)$, $\rho(\pi)$ and an additional quantity $t^s(\pi)$ representing the expected time to find and use a station assuming at least one station is available among C, the visits sequence associated with π. We derive the quantity $t^s(\pi)$ based on the work of Arndt et al. (2016) that describes $t^s(\pi)$ as a sum over all stations of the accumulated driving time to a station and its utilization cost weighted by the probability the driver charges exactly at this station.

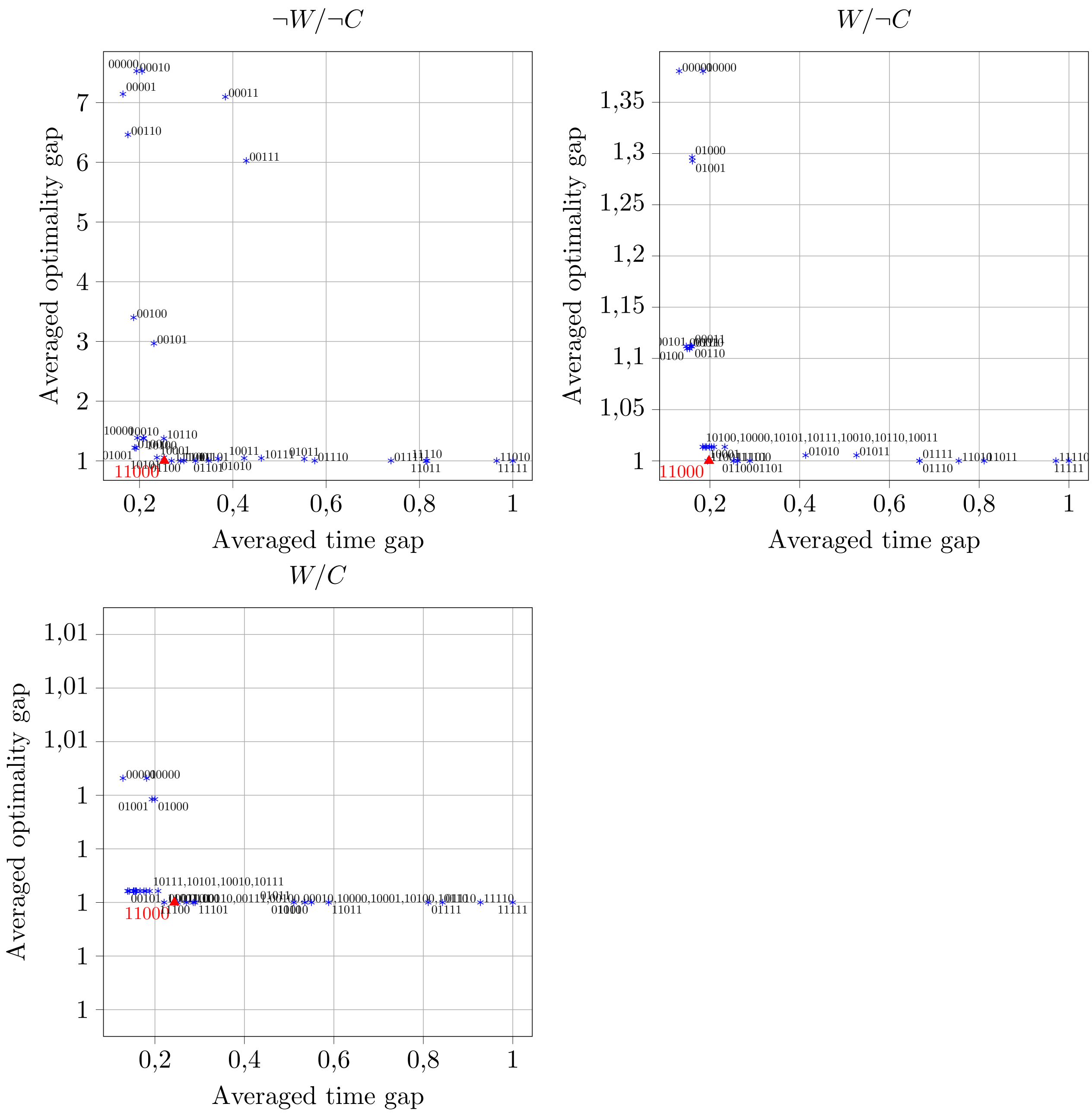


Figure 18 Comparison of heuristic dominance criteria for problem variants $\neg W/\neg C$, $W/\neg C$ and W/C

Note. The Figure shows the averaged optimality gap $g_\alpha = \sum_i {}^{\alpha_i}/_{\alpha_i^{\text{opt}}}$ as a function of the averaged computational time gap $g_t = \sum_i {}^{t_i}/_{t_i^{\text{opt}}}$ for each possible heuristic criterion for $\neg W/C$. For each variant, both values are averaged over 16 instances corresponding to *BER-1* and *SF-1* combined resp. with *low-15%*, *avg-60%* and *high-90%* for $\overline{S} \in [1200, 1400, 1600, 1800]$ and fixed $\overline{T} = 10$. Red triangles show our selected dominance criteria.

Proposition 4. *Cost α can be decomposed to exhibit $t^s(\pi)$ as follows,*

$$\alpha(\pi) = t^s(\pi) \cdot \rho(\pi) + \bar{\rho}(\pi) \cdot (t(\pi) + \beta_{v_n}) \tag{F.1}$$

.

Proof. We recall that $\alpha(\pi) = A(\pi) + \bar{\rho}(\pi) \cdot \beta_{v_n}$ (cf. Equation 3.9). For the sake of conciseness, we simplify the notation for the remainder of this proof as follows: $C = (0, 1, ..., n)$ such that $t_{k,k+1} = t_{v_k, v_{k+1}}$, $\bar{\rho}_k = \prod_{i=0}^{k} \bar{p}_k$ We let $\rho_n = \rho(\pi)$, $A_n = A(\pi)$, $t_n = t(\pi)$, $t_n^s = t^s(\pi)$.

We then define t^s based on Arndt et al. (2016) as

$$t_n^s = \frac{\sum_{k=0}^{n-1} \bar{\rho}_{k-1} p_k (t_k + \gamma_k)}{\rho_n}.$$

We now note that $\bar{\rho}_k p_k$ represents the probability of station k being available when visited, given that no station in $i \in (0, ..., k-1)$ was available.

We then introduce the quantity $B_n = \sum_{k=0}^{n-1} \bar{\rho}_{k-1} p_k (t_k + \gamma_k)$ such that $t_n^s \cdot \rho_n = B_n$ and note that to prove F.1, it is sufficient to show

$$A_n = B_n + \bar{\rho}_n \cdot t_n, \tag{F.2}$$

which follows by recursion:

Step 1: For $n = 0$, F.1 holds : $A_0 = t_{0,1}$ and $B_0 = p_1 t_{0,1} + \bar{p}_1 t_{0,1} = A_0$.

Step 2: We assume that F.1 holds and show that $A_{n+1} = B_{n+1} + \bar{\rho}_{n+1} \cdot t_{n+1}$ holds, too:

$$A_{n+1} = A_n + \bar{\rho}_n \cdot (t_{n,n+1} + p_{n+1}\gamma_{n+1})$$

.

$$B_{n+1} = B_n + \bar{\rho}_n p_{n+1} \cdot (t_{n+1} + \gamma_{n+1}) \tag{F.3}$$

Given F.1,

$$\begin{aligned} &A_{n+1} = B_n + \bar{\rho}_n \cdot t_n + \bar{\rho}_n (t_{n,n+1} + p_{n+1}\gamma_{n+1}) \\ \Leftrightarrow\ &A_{n+1} = B_n + \bar{\rho}_n \cdot t_n + \bar{\rho}_n (t_{n,n+1} + p_{n+1}\gamma_{n+1}) + \bar{\rho}_n p_{n+1} \cdot (t_{n+1} + \gamma_{n+1}) - \bar{\rho}_n p_{n+1} \cdot (t_{n+1} + \gamma_{n+1}) \end{aligned} \tag{F.4}$$

From F.3 and F.4, we get :

$$\begin{aligned} &A_{n+1} = B_{n+1} + \bar{\rho}_n \cdot (t_n) + \bar{\rho}_n (t_{n,n+1} + p_{n+1}\gamma_{n+1}) - \bar{\rho}_n p_{n+1} \cdot (t_{n+1} + \gamma_{n+1}) \\ \Leftrightarrow\ &A_{n+1} = B_{n+1} + \bar{\rho}_n \cdot t_{n+1} - \bar{\rho}_n (p_{n+1}) t_{n+1} \\ \Leftrightarrow\ &A_{n+1} = B_{n+1} + \bar{\rho}_{n+1} \cdot t_{n+1} \end{aligned} \tag{F.5}$$

This concludes the proof. □