smart cities

MDPI

Article

Greedy-VoI Time-Mesh Design for Rolling-Horizon EMS: Optimizing Block-Variable Granularity and Horizon Under Compute Budgets

Gregorio Fernández [1,*], J. F. Sanz Osorio [2], Adrián Alarcón [1], Miguel Torres [1] and Alfonso Calavia [1]

[1] Department of Electrical Systems, CIRCE Technology Centre, 50018 Zaragoza, Spain; aealarcon@fcirce.es (A.A.); mtorres@fcirce.es (M.T.); acalavia@fcirce.es (A.C.)
[2] Instituto de Investigación Mixto de la Energía y la Eficiencia de los Recursos de Aragón, ENERGAIA, Universidad de Zaragoza-Fundación CIRCE, 50018 Zaragoza, Spain; jfsanz@unizar.es
* Correspondence: gfernandez@fcirce.es

Highlights

What are the main findings?

- Rolling-horizon EMS/MPC exhibits a substantial trade-off between temporal granularity and real-time computational feasibility, with diminishing economic gains beyond a specific time resolution.
- Multiple time-mesh and horizon configurations deliver near-identical optimal costs, while solver execution time varies across designs.

What are the implications of the main findings?

- Treating temporal granularity and optimization horizon as explicit design variables enables improved EMS/MPC performance under strict compute budgets.
- The proposed compute-aware and automatable mesh–horizon selection framework supports scalable and robust EMS/MPC deployment on resource-constrained platforms.

Abstract

Rolling-horizon energy management systems (EMSs) and model predictive control (MPC) for microgrids in smart cities face a fundamental trade-off: finer temporal discretization improves operational performance but rapidly increases the size of the optimization problem and execution time, jeopardizing real-time feasibility. Furthermore, in short-horizon operation, only the first control actions are implemented, while long-horizon decisions primarily guide feasibility and constraints. This paper proposes a computation-aware temporal mesh design layer that jointly selects a variable granularity of blocks and an optimization horizon, explicitly bounded by market-aligned settlement steps and per-cycle computation budgets. Candidate configurations are represented as pairs ⟨B, H⟩, where B is a constant-step block programme, and H is the optimization horizon, and they are uniquely tracked through an auditable mesh signature. The method first evaluates a predefined, market-consistent set of solutions ⟨B, H⟩ to establish reproducible cost and execution-time benchmarks, then applies a greedy value-of-information (Greedy-VoI) search that generates valid neighbouring meshes through local refinement, thickening, and resolution reallocation without violating the basic requirements that every solution must meet. All candidates are evaluated using the same microgrid use case and the same comparative KPIs, enabling the systematic identification of near-optimal mesh–horizon designs for practical EMS implementation.

Academic Editors: Pierluigi Siano and Luis M. Fernández-Ramírez

Received: 23 December 2025
Revised: 28 January 2026
Accepted: 2 February 2026
Published: 10 February 2026

Keywords: energy management system (EMS); model predictive control (MPC); rolling horizon; variable temporal granularity; greedy algorithm; value of information (VoI); non-uniform time mesh; microgrid scheduling; compute-aware optimization

1. Introduction

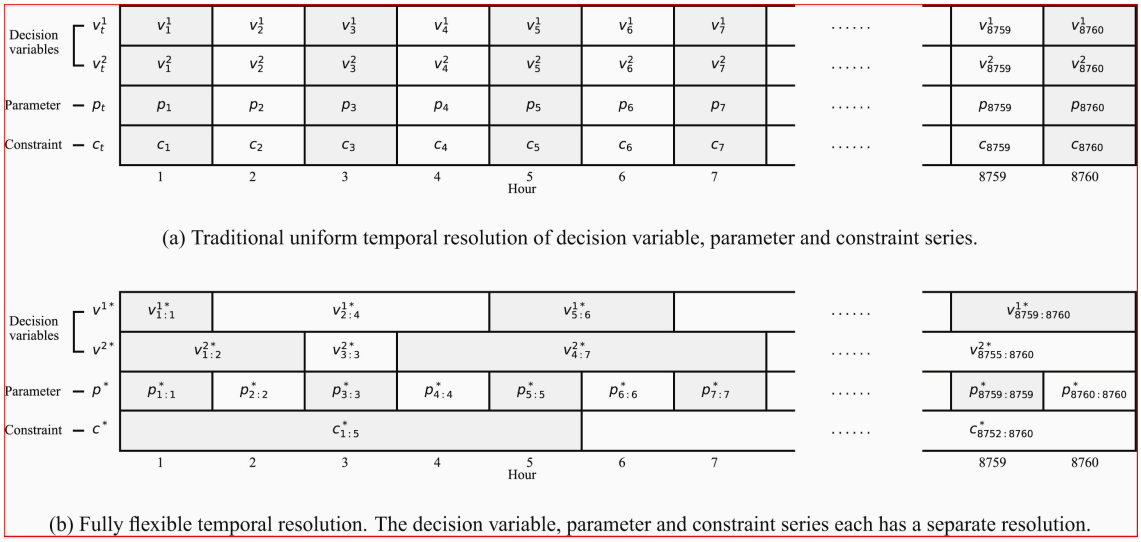
Electric power systems are being reshaped into the smart grid paradigm, characterized by bidirectional power and information flows and the rapid expansion of variable renewable generation and distributed energy resources (DERs), enabled by advanced metering, communications, and automation technologies [1]. This transition supports decarbonisation by displacing fossil generation and electrifying end-uses, including space and water heating [2]. At the same time, the proliferation of behind-the-meter PV, storage, EV charging, and flexible loads increases operational complexity in distribution grids, raising concerns such as reverse power flows, voltage deviations/unbalance, protection challenges, and local congestion [3,4]. Consequently, system "flexibility"—the ability to shift or modulate generation, consumption, and storage in response to external signals—has become a central enabling resource for integrating variable renewables while maintaining security and network compliance [5,6]. In particular, demand-side management can convert electrified loads into controllable resources that follow price, carbon-intensity, or grid operator signals, ranging from tariff-driven load shaping to direct, real-time control of smart loads and DER portfolios [7]. Energy management systems (EMS) operationalise flexibility by acquiring field data, forecasting, and optimally scheduling heterogeneous DERs and controllable loads to meet objectives (e.g., cost, self-consumption, emissions) under user and grid/market constraints [8,9].

To take advantage of the flexibility and controllability of distributed energy resources presupposes a high degree of automation in monitoring, optimization and actuation, because evidence from residential demand–response programmes shows that engagement is hindered by perceived complexity and effort, as well as concerns about risk and loss of control; accordingly, automation becomes an enabling condition when accompanied by transparency and user final control [10]. In line with these insights, the stakeholder-centric surveys conducted in the REEFLEX project report that only 39.8% of respondents currently use any energy management/monitoring tool and only 19% use a dedicated energy/flexibility management tool, while simultaneously revealing strong preferences—sometimes unanimous—for fully automated solutions with user override capabilities and for trust-building features such as long-term support and automatic security updates (with higher expectations of automatic control for critical loads) [11]. An EMS can therefore be defined as a supervisory hardware–software layer that acquires data, forecasts disturbances, and solves an optimization problem to schedule and dispatch controllable resources under technical and contractual constraints [7,8]. Such concepts span home, building, industrial, and microgrid/community EMSs. Across these domains, model predictive control (MPC) is widely adopted as a core algorithmic paradigm because its receding-horizon formulation explicitly embeds forecasts and operational constraints and has been extensively validated for both building HVAC energy management and microgrid operation optimization [12,13].

In EMSs/MPC based on deterministic optimization techniques, the dominant sources of performance degradation are driven mainly by imperfect forecasts of exogenous inputs (e.g., weather, demand, renewable generation, and prices) and by discretization of continuous dynamics and device operating modes into granular decisions on a fixed time grid [13–16]. Beyond deterministic EMS/MPC formulations, learning-based energy

management has gained traction in building and home applications, particularly to cope with complex, time-varying environments and continuous control actions. For example, a price-aware HEMS based on an actor–critic (DDPG) agent augmented with an LSTM for forecasting jointly controls HVAC power and an energy storage system to balance energy costs and thermal comfort, and is benchmarked against rule-based and DRL baselines [17]. The receding- or rolling-horizon principle partly mitigates forecast uncertainty by repeatedly updating the optimization as new measurements and forecasts become available, thereby correcting operation schedules before prediction errors accumulate [14,15]; likewise, increasing the temporal granularity of the EMS (e.g., 15 min intervals instead of hourly steps) can reduce discretization artefacts and better capture intra-hour variability, ramping behaviour, and constraint activation—although at the cost of a larger optimization problem with higher computation needs [12,18]. However, finer granularity increases the number of decision variables and constraints roughly proportionally to the number of time steps. It can cause solve times to rise sharply with problem size, particularly in mixed-integer EMS formulations, where combinatorial growth in binary variables is a significant factor [19]. In complex, large-scale deployments (e.g., microgrids or aggregated DER portfolios), this computational burden can make real-time operation impractical, leading to the paradox that the computation time exceeds the control interval so that the "optimal" solution becomes outdated before it can be applied [19].

To address the increasing computational demands posed by complex EMS/MPC formulations, particularly when discretized at finer time steps, the literature has proposed several strategies. A proposed solution is to decompose long-horizon (multi-day) mixed-integer scheduling problems into a sequence of overlapping subproblems (a few hours) solved in a rolling-horizon manner, which can deliver substantial runtime reductions for large energy-hub Mixed Integer Linear Programming (MILP) formulations while preserving the quality of the resulting operating policies [20]. Beyond fixed periodic re-optimization, the rolling-horizon execution itself can also be made adaptive: ref. [21] proposes a dynamic scheduling tool that selects informative start times for the rolling-horizon solves under uncertainty—rather than uniformly triggering them—thereby improving operational performance in a microgrid energy management case study. In parallel, temporal aggregation of input time series—via clustering into typical time steps or typical periods (e.g., typical days/weeks)—reduces the number of time steps and can be tuned to the mathematical structure of the optimization model (notably the presence of time-linking constraints such as storage); recent comparative studies formalize the accuracy–runtime trade-off as a Pareto frontier and provide methods to identify near-optimal aggregation configurations instead of relying on ad hoc choices [22,23]. Another alternative is multi-rate discretization, in which different decision variables and constraints are represented at distinct (and potentially time-varying) temporal resolutions according to their characteristic dynamics; the fully flexible temporal resolution formulation proposed by [24] explicitly enables per-variable resolutions and demonstrates computational efficiency gains in large-scale co-optimization of investment and operational decisions (see Figure 1). Finally, the computational budget is also shaped by EMS design choices, such as the length of the optimization horizon. Multi-objective receding-horizon scheduling studies suggest that longer optimization horizons can enhance renewable utilization and operational outcomes, albeit at a higher computational cost, thereby reinforcing the motivation for adaptive update policies and multi-resolution modelling in practical EMS deployments [25]. More generally, variable-granularity schemes—where temporal resolution is adjusted during execution—are emerging as another approach to address the computational demands of complex EMS/MPC formulations.



(a) Traditional uniform temporal resolution of decision variable, parameter and constraint series.

(b) Fully flexible temporal resolution. The decision variable, parameter and constraint series each has a separate resolution.

Figure 1. Examples of uniform and fully flexible temporal resolutions of time-dependent decision variables, parameters, and constraints. (**a**) Traditional uniform temporal resolution, where all series share the same time index and constant time-step duration over the whole horizon. (**b**) Fully flexible temporal resolution, where decision variables, parameters, and constraints are defined over independent time blocks of varying duration. The superscript "*" denotes a series defined using fully flexible temporal resolution. Time blocks are indexed by their starting and ending unit periods, separated by a colon (e.g., v2:4∗v^*_{2:4}v2:4∗ spans from unit period 2 to 4). The ellipsis ("...") indicates omitted intermediate time blocks for compact representation. Adapted from [24].

These computational strategies and the underlying multi-timescale rationale are not exclusive to deterministic EMS/MPC implementations. Similar tractability constraints arise in other energy decision-making layers, including market-operation models, where multiple temporal layers are naturally imposed by heterogeneous physical time constants (e.g., electricity, gas, and heat networks) and by settlement processes. Recent work on regional multi-energy markets explicitly models such multi-timescale feasibility constraints. It addresses their complexity via hierarchical reinforcement learning, thereby demonstrating that coarse-to-fine temporal structures can be instrumental for preserving feasibility while keeping computation manageable [26].

Variable temporal granularity (also referred to as flexible temporal resolution or multi-timescale discretization) can be defined as the deliberate use of non-uniform time steps along the optimization horizon—and, in some formulations, different time steps for different decision variables—so that fast dynamics and near-term decisions are represented with acceptable resolution. In contrast, slower trends and far-future decisions are represented more coarsely to reduce problem size. In EMS/MPC settings, this concept is especially natural because the controller is usually executed in a receding- or rolling-horizon structure, where forecasts and states are updated frequently and only the first portion of the computed optimal trajectory or operation set-points are implemented before re-optimizing; consequently, modelling fidelity is most valuable near the current time, whereas distant decisions mainly provide feasibility and boundary guidance. A basic justification for variable granularity is also offered by the exponential decay of sensitivity (EDS). Under suitable regularity conditions, the sensitivity of the optimal decision at stage i to perturbations at stage j can be bounded to decay exponentially with the temporal distance |i-j| [27]. In rolling-horizon EMS/MPC—where near-term actions are executed, and the horizon is repeatedly shifted—EDS implies that aggregating or coarsening far-future time steps may have limited impact on the current control action, while still capturing long-term constraints and dominant trends. This rationale is directly exploited in diffusing-horizon MPC,

which constructs a discretization grid whose spacing becomes exponentially sparser as the horizon is extended, achieving substantial computational savings with limited closed-loop performance degradation [28].

Multi-timescale rolling operation schemes for building and integrated energy systems exemplify this idea by combining day-ahead schedules with progressively finer intraday and real-time refinements (e.g., hourly $\rightarrow$ 15 min $\rightarrow$ 5 min) [29], and by embedding horizons with increasing step size directly into MPC to balance economic performance, robustness, and runtime [30]. Closely related multi-horizon formulations also seek long look-ahead horizons without an explosion in variables, e.g., distributed MPC for networks of energy hubs [31] and multi-resolution approaches for storage control that explicitly trade temporal/state resolution for computation, allowing re-optimization more frequently [32]. Beyond EMS, flexible temporal resolution has recently become an explicit computational lever in large-scale MILP scheduling, such as network-constrained unit commitment with adaptive time-period aggregation [33] and cost-oriented time-adaptive UC formulations [34], reinforcing the broader observation—quantified in comparative studies of temporal resolution selection—that higher temporal fidelity improves representation accuracy but can quickly render optimization impractical at scale [35].

Although temporal aggregation and resolution selection have been studied in planning-oriented energy system models, the methodological development for operational microgrid EMS/MPC—where computational latency becomes a hard real-time constraint—remains comparatively limited [22]. Recent contributions nevertheless illustrate the potential of non-uniform temporal designs: in reference [36] it is introduced an "optimal rolling-horizon strategy" (ORoHS) that explicitly partitions prediction/execution horizons to trade off dispatch cost and forecast accuracy under uncertainty. At the same time, diffusing-horizon MPC provides a constructive coarsening mechanism in which sampling points become progressively sparser (e.g., exponentially) as they move farther into the horizon, leveraging the fact that far-future perturbations have a weaker influence on near-term control actions [21]. In parallel, time-adaptive discretization is also emerging in large-scale mixed-integer scheduling problems (e.g., unit commitment), where cost-oriented temporal resolution seeks to retain fine granularity only where it is economically consequential [22]. However, most works either evaluate a limited set of discretizations or adopt formulation-specific coarsening rules, and rarely provide a general procedure for computing a variable-granularity design for a given microgrid and context. Moreover, the coupled problem of jointly selecting a variable mesh and an optimization horizon length remains largely unaddressed in EMS practice. Building on the above gaps, this work aims to take a step further by proposing a compute-budget-aware procedure that evaluates admissible optimization-horizon lengths and block-structured variable granularities tailored to the target market setting. Then, it employs a greedy value-of-information (Greedy-VoI) metaheuristic to generate and screen new mesh–horizon combinations. In doing so, the proposed methodology targets microgrid-specific near-optimal temporal designs that explicitly balance closed-loop economic performance, forecast uncertainty, and real-time solvability.

The remainder of this paper is organized as follows. Section 2 presents the proposed methodology, including the problem formulation, the design choices adopted for variable temporal granularity and optimization horizon selection, and the evaluation procedure. Section 3 reports the main results obtained across the considered case study. Section 4 discusses the findings, highlighting the practical implications and limitations observed. Finally, Section 5 summarizes the main conclusions and outlines directions for future research.

2. Materials and Methods

This section introduces the proposed methodology for selecting an optimal block-variable temporal discretization and an optimization horizon for rolling-horizon EMS/MPC operation under explicit computational constraints. First, the microgrid modelling layer (objective function and constraints) and the market layer are outlined, which define the admissible settlement-aligned time steps and candidate horizon lengths. Next, the workflow is described in Figure 2, where a pool of market-consistent "baseline" discretizations is evaluated to build a comparable KPI benchmark, which is then used to guide a metaheuristic search. Finally, the high-level logic of the Greedy-VoI procedure is presented, which is used to generate and assess new discretization–horizon candidates, along with the criteria for selecting the final recommended configuration.

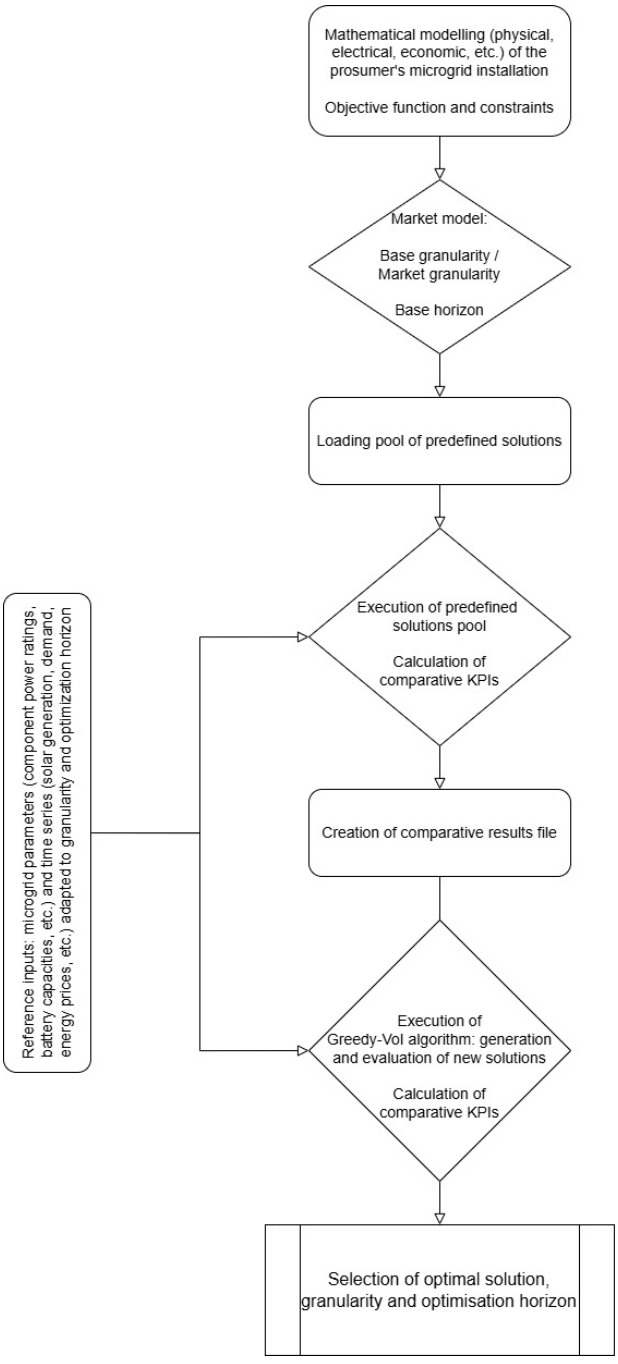


Figure 2. Overview of the proposed compute-aware workflow to select an EMS/MPC optimization horizon and a block-variable temporal granularity.

At a high level, the proposed procedure can be interpreted as a compute-aware time-mesh design layer placed on top of a conventional EMS optimizer. The starting point is a mathematical model of the prosumer microgrid that includes electrical/physical constraints, economic terms, and a set of reference inputs (e.g., demand, PV generation, electricity prices, and device limits) representative of the operating period of interest. The market layer then provides the practical boundaries of the decision space: (i) a base/market granularity consistent with market clearing and settlement, and (ii) a base horizon that reflects how far ahead the EMS is expected to plan. These elements define the admissible combinations of "how often decisions are made" (temporal resolution) and "how far ahead the EMS looks" (horizon). At the same time, a compute budget constrains what is feasible to solve repeatedly in real time.

The workflow in Figure 2 proceeds in two stages. In the first stage, the method loads and executes a pool of predefined solutions, i.e., a curated set of block schedules and horizons that are directly compatible with the target market structure. Each candidate is run through the same evaluation pipeline: a rolling-horizon simulation (or repeated replanning procedure) is executed, only the first control action is applied at each step, and the process is repeated across the study horizon. For each candidate configuration, the framework computes comparative KPIs that capture both operational quality and computational practicality. The output of this stage is a consolidated results dataset that provides a transparent baseline and a cost–runtime reference for subsequent search.

In the second stage, the Greedy-VoI algorithm iteratively generates and evaluates new candidate discretizations beyond the predefined pool. Conceptually, Greedy-VoI explores local modifications to the time mesh—such as refining near-term blocks (where actions are executed) and coarsening far-term blocks (where decisions primarily provide boundary guidance)—and may also test alternative horizon lengths to ensure market alignment and to address computational limits at all times. After each candidate is generated, it is evaluated through the same execution and KPIs, and the best-performing configurations are retained according to the selected decision rule. The procedure finishes when the compute budget and/or stopping criteria are met, returning the selected optimal solution together with its recommended block schedule (granularity profile) and optimization horizon, ready to be deployed as the EMS planning configuration for the specific microgrid and market framework.

2.1. Mathematical Definition of Granularity, Blocks, and Optimization Horizon

This section provides a mathematical definition of the blocks that form the EMS/MPC optimization horizon, the parameters that characterize them, and the equations they must satisfy. The pair defines each solution that makes up an optimization horizon:

$$\langle \mathrm{B},\mathrm{H} \rangle \tag{1}$$

where B is a schedule of time blocks, and H is the optimization horizon. A time-block schedule is an ordered list of blocks:

$$\mathrm{B} = \{(n_1 \times \Delta t_1),\ldots,(n_\mathrm{B} \times \Delta t_\mathrm{B})\} \tag{2}$$

where block i comprises $n_i \in \mathbb{Z}_{\geq 1}$ consecutive time steps (granularities) of identical duration $\Delta t_i > 0$ (min).

For auditability, the mesh signature is defined as

$$\mathrm{sig}(\mathrm{B},\mathrm{H}) = [(n_1 \times \Delta t_1),\ldots,(n_\mathrm{B} \times \Delta t_\mathrm{B});\mathrm{H}] \tag{3}$$

2.1.1. Optimization Horizon

The optimization horizon (minutes) is

$$\mathrm{H} = \sum_{i=1}^{\mathrm{B}} n_i * \Delta t_i \tag{4}$$

And the effective size of the discrete problem is

$$N = \sum_{i=1}^{\mathrm{B}} n_i \tag{5}$$

2.1.2. Conditions That Must Be Met by the Combinations of Values That Define the Optimization Horizons

Unless extended for a specific application (e.g., allowing 30′ in 15′ markets), there is a limited group or set of admissible time steps (granularities):

- {5, 15, 60} minutes for 15 min settlement markets;
- {15, 60, 120} minutes for 1 h settlement markets;
- Integer blocks and closure of the horizon:

$$\begin{gathered} \mathrm{H} = \sum_{i=1}^{\mathrm{B}} n_i * \Delta t_i \\ \mathrm{With B} \in Z_{\geq 1}, n_i \in Z_{\geq 1}, \Delta t_i \in \Delta \end{gathered} \tag{6}$$

This constraint ensures that the blocks tile the horizon exactly.

- Horizon domain:

$$\mathrm{H} \in \mathrm{H},\ \mathrm{H} = \{2880, 4320\}\ \text{(2 or 3 days in minutes)} \tag{7}$$

Project predefined horizon lengths are 2 or 3 days ($\mathrm{H} \in [48, 72]$ h). Other works could propose other optimization horizons.

- Model-size bound:

$$N = \sum_{i} n_i \leq N_{\max} \tag{8}$$

$N_{\max}$ is chosen based on solver and hardware limits (e.g., 220–300 periods for a MILP with mixed-integer devices). It is the first control against intractable instances.

- Real-time calculations:

A per-cycle compute budget $T_{\max}$ (seconds) is enforced, with

$$T_{\max} \leq \min \Delta t_i \tag{9}$$

In a rolling-horizon operation, the optimization is executed periodically, and only the first set of set-points is applied before the next re-optimization. Therefore, the solve time must be shorter than the control interval (which, in this work, is aligned with the minimum time step of the adopted time mesh). Otherwise, the controller may not deliver an updated action before it is due, leading to missed deadlines, overlapping optimization cycles (a new run starting before the previous one finishes), and the risk that the computed "optimal" schedule becomes outdated before implementation.

- Non-decreasing resolution:

$$\Delta t_1 \leq \Delta t_2 \leq \cdots \leq \Delta t_{\mathrm{B}} \tag{10}$$

This avoids repeated switching of fine → coarse → fine, which tends to generate unstable set-points and redundant solver work.

2.1.3. Predefined Pool of Solutions

As previously indicated, some solutions adapted to the electric market environment have been defined and tested before applying the Greedy-VoI algorithm to ensure they are evaluated. Table 1 presents this predefined market of solutions.

Table 1. Mathematical definition of a predefined pool of solutions.

Market Step	Solution	H [Horizon]	*i* [Blocks]	n_1	Δt_1	n_2	Δt_2	n_3	Δt_3	$[(n_1 x \Delta t) \ldots (n_B x \Delta t_B); H]$
1 h	0/reference	3 days 72 h 4320 min	1	72	60					[(72x60); 4320]
	1	3 days 72 h 4320 min	2	4	15	71	60			[(4x15), (71x60); 4320]
	2	3 days 72 h 4320 min	3	4	15	23	60	24	120	[(4x15), (23x60), (24x120); 4320]
	3	2 days 48 h 2880 min	2	4	15	47	60			[(4x15), (47x60); 2880]
	4	2 days 48 h 2880 min	3	4	15	23	60	12	120	[(4x15), (23x60), (12x120); 2880]
15 min	0/reference	3 days 72 h 4320 min	1	288	15					[(288x15); 4320]
	1	3 days 72 h 4320 min	2	3	5	287	15			[(3x5), (287x15); 4320]
	2	3 days 72 h 4320 min	3	3	5	95	15	48	60	[(3x5), (95x15), (48,60); 4320]
	3	2 days 48 h 2880 min	2	3	5	191	15			[(3x5), (191x15); 2880]
	4	2 days 48 h 2880 min	3	3	5	95	15	24	60	[(3x5), (95x15), (24x60); 2880]

2.2. Greedy-VoI Algorithm

Greedy-VoI frames time-mesh and horizon selection as a constrained discrete design problem over candidate signatures $s = \mathrm{sig}(\mathrm{B}, \mathrm{H})$, where $\mathrm{B} = \left\{ (n_1, \Delta t_1), \ldots, \left(n_{|B|}, \Delta t_{|B|} \right) \right\}$ is a schedule of blocks (each block containing n_i consecutive steps of duration Δt_i) and H is the optimization horizon. The induced discrete size is $N(\mathrm{B}) = \sum_i n_i$ and $\mathrm{H} = \sum_i n_i \Delta t_i$, subject to market-admissible step sizes, exact tiling of the horizon, monotone coarsening along the horizon $\left(\Delta t_1 \leq \cdots \leq \Delta t_{|B|} \right)$, real-time feasibility constraints (e.g., $N(\mathrm{B}) \leq N_{\max}$ and per-cycle solver time $\tau(s) \leq \tau_{\max}$). Over this feasible set, each candidate s is evaluated through a rolling-horizon execution to obtain an operational KPI (e.g., total cost or penalty-augmented cost) $J(s)$ and a computational metric $\tau(s)$. Greedy-VoI then

applies a greedy, value-of-information principle by prioritizing modifications that provide the largest marginal improvement per unit computational "expense", e.g.,

$$Vol(s \to s') = \frac{J(s) - J(s')}{\tau(s') - \tau(s) - \varepsilon} \tag{11}$$

where s' is obtained from s via an elementary mesh operator (refine, coarsen, or reallocate resolution across blocks, optionally combined with a horizon change) and $\varepsilon > 0$ avoids division by zero. A greedy choice based on marginal gains is theoretically motivated in settings that exhibit diminishing returns, where selecting the best incremental move at each iteration yields provable near-optimality for submodular objectives—a property that, while not guaranteed in EMS/MPC mesh design, provides a principled rationale for using marginal-gain criteria to guide a metaheuristic search.

Operationally, the algorithm starts from a small pool of market-consistent baseline meshes and selects an incumbent $s^{(0)}$ that is feasible under $(N_{\max}, \tau_{\max})$. At iteration k, a neighbourhood $\mathcal{N}\left(s^{(k)}\right)$ is generated by applying admissible mesh transformations (e.g., refining the near-term blocks where set-points are executed, and compensating by coarsening far-term blocks to preserve feasibility), producing a set of candidates $\{s'\}$. Each s' is either (i) screened using lightweight proxies (e.g., expected change in N or predicted τ) and/or (ii) fully evaluated by running the rolling-horizon EMS to obtain $J(s')$ and $\tau(s')$. The next iterate is chosen greedily as $s^{(k+1)} = \arg\max_{s' \in \mathcal{N}(s^{(k)})} \mathrm{VoI}\left(s^{(k)}, \to, s'\right)$ among candidates that respect the compute constraints; the incumbent is updated when J improves (or when a selected dominance rule improves a cost–time trade-off frontier). A key structural justification for focusing refinement near the beginning of the horizon is that, in rolling-horizon control, only the first decisions are implemented, and the problem is then re-solved; moreover, sensitivity to far-future discretization typically decreases with temporal distance. This intuition is consistent with the diffusing-horizon time-coarsening rationale—where the grid becomes progressively sparser farther into the horizon (often exponentially)—which has been shown to reduce computational effort while preserving near-term control quality in MPC.

The algorithm in Table 2 summarizes the workflow for searching for computationally feasible mesh-horizon designs. The method starts from an initial pool of evaluated, market-consistent predefined candidates and selects an incumbent configuration that satisfies the per-cycle compute budget. It then iteratively generates a neighbourhood of feasible meshes by applying local operators that reallocate temporal resolution while preserving admissible step sizes and the optimization horizon. Each neighbour is evaluated through the same EMS pipeline to obtain both an operational KPI (objective value) and a computational KPI (solver execution time). A greedy value-of-information criterion is then applied to select the next incumbent from among candidates that meet the computational budget, prioritizing those that deliver the best performance improvement per computational effort. The process continues until a stopping condition is met (e.g., a maximum number of evaluations or lack of improvement), returning a near-optimal mesh–horizon design together with an auditable mesh signature. It should be highlighted that s denotes the current incumbent mesh–horizon configuration, s' a candidate neighbour generated from s, and s^* the best-so-far configuration maintained throughout the search (and returned at termination).

For reproducibility, the Greedy-VoI implementation adopts the following main stopping and neighbourhood rules:

- Stopping: evaluate up to "*n_new*" new candidates (in batches of up to 5); stop when the budget is exhausted or when no new valid neighbours remain (all infeasible/visited).
- Neighbours: each candidate is generated by applying one operator

Table 2. Greedy-VoI search algorithm. The superscript '$*$' denotes the current incumbent (best-so-far) candidate selected by the Greedy-VoI procedure (not multiplication).

Greedy-VoI Search Algorithm Pseudocode

Inputs:
• Initial evaluated pool of candidates $\mathcal{S}0$ with results (objective function value J, execution time T)
• Baseline time series data and an evaluation pipeline (EMS optimiser model and inputs)
• Admissible time step set and mesh construction rules
• Neighbourhood operators (e.g., refine, coarsen, shift/reallocate resolution)
• Per-cycle compute budget $T_{\max}$ and stopping criteria (K evaluations/no-improvement)
Output:
• Best candidate $s^* = \langle B^*, H^* \rangle$ and results (objective function value $J(s^*)$, execution time $T(s^*)$)
 	
// Initialisation
Select an initial incumbent s^* from $\mathcal{S}0$ (best objective among candidates satisfying $T \leq T_{\max}$).
Mark all previously evaluated mesh signatures as "visited".
 	
// Greedy exploration loop
While the stopping criteria are not met, do:
Generate a set of neighbouring meshes $\mathcal{N}(s^*)$ by applying admissible operators to B^* and H^*.
Discard neighbours that violate feasibility rules (market-aligned steps, optimization horizon H, admissible sizes).
For each feasible neighbour s in $\mathcal{N}(s^*)$ not yet visited:
Adapt the required inputs to the candidate mesh (e.g., aggregate/interpolate time series).
Evaluate s using the same EMS pipeline to obtain ($J(s)$, $T(s)$).
Store (sig(s), $J(s)$, $T(s)$) in the results log and mark sig(s) as visited.
Select the next incumbent s_next among evaluated neighbours that satisfy $T(s) \leq T_{\max}$, prioritizing the best improvement in performance under the compute budget (Greedy-VoI rule).
If s_next improves upon s^* according to the decision rule, set $s^* \leftarrow s$_next; otherwise update finalization counters.
End while
Return s^* and its signature sig(B^*,H^*)

2.3. Tool Testing

The tool has been developed, implemented, and tested in the PYTHON environment, using these main libraries:

- PYTHON, version 3.9
- PANDAS, version 2.3.3
- NUMPY, version 2.0.2
- ORTOOLS, version 9.14.6206 (solver interface; SCIP backend)
  - o Per-run time limit: 300,000 ms.

The running platform's main characteristics are

- 11th Gen Intel(R) Core (TM) i7-1165G7 @ 2.80 GHz (2.80 GHz);
- 16 GB RAM.

EMS Simulation Scenario

It is essential to emphasize that the procedure or algorithm developed and proposed in this paper is agnostic about the microgrid in which the EMS is to be implemented, meaning that this methodology can be applied to all types of real-world installations.

A detailed microgrid model has been developed (see it in Figure 3), which includes solar photovoltaic systems for energy generation (in conjunction with alternative renewable energy sources such as wind and hydroelectric power), battery energy storage systems (BESSs), electric vehicle (EV) charging infrastructures, hydrogen production facilities, and various energy consumption nodes, which can be classified as manageable and non-manageable. This detailed microgrid model is customized to meet the specific installation requirements through the application of a configuration file that specifies the existing devices, along with the operational mode of the grid—whether interconnected or isolated—the incorporation of direct current (DC) or alternating current (AC) components, and additional operational factors, such as the ability to inject energy into the grid.

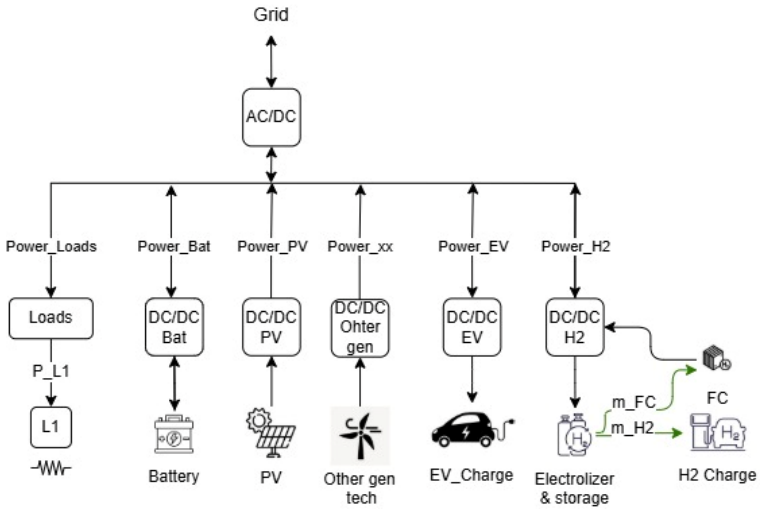


Figure 3. Example of a microgrid general model. The green lines indicate hydrogen flow.

The fundamental operational parameters of the microgrid components (power output, storage capacity, efficiency, among others) are introduced through an auxiliary configuration file. Forecasts (such as solar generation, energy pricing, and energy demand) are introduced via a forecasts file, which has a considerable impact on the system's results. This file further incorporates assessments of equipment availability, thereby facilitating the recalculation and establishment of optimal operational set-points in alignment with the microgrid's actual operational state.

The objective function minimizes the total operating cost, which is the sum of component-wise terms associated with the battery, EV charging/V2G, hydrogen system, and grid exchanges. Battery and EV costs include an energy/usage term and a smoothing penalty that discourages highly irregular charging/discharging profiles. The hydrogen term accounts for electrolyser and fuel cell operation, optional auxiliary truck refilling, and hydrogen revenues from vehicle refuelling. In contrast, the grid term captures contracted power and time-dependent buy/sell transactions. Feasible operation is enforced through a DC-bus power balance (Kirchhoff-based) that includes converter efficiencies, PV utilization limits bounded by forecasts and installed capacity, and asset-specific dynamics and limits. Storage devices (BESS, EV, and hydrogen tank) are constrained by initial state, minimum/maximum capacity, mutually exclusive charge/discharge modes, and inter-temporal

state update equations; additional auxiliary variables model step-to-step power variations to smooth the process. Finally, grid import/export is bounded by contracted power limits.

The general model presented in the previous paragraphs has been adapted to the characteristics of the real microgrid examined in this paper. The main characteristics of the microgrid of the use case are summarized in Table 3, and the microgrid diagram is illustrated in Figure 4. The system operates independently of the electrical grid. It encompasses a photovoltaic solar generation system, a controllable electric vehicle charging station, a hydrogen refuelling facility, and an energy storage system utilizing batteries. In this scenario, hydrogen is delivered to the system via external transport vehicles, as there is no electrolyzer, and a fuel cell converts the stored hydrogen into electrical energy. The optimization model is also formulated to determine the appropriate timing for hydrogen imports.

Table 3. Test site asset parameters.

Asset	Parameter	Value
Test site microgrid	Grid connection	Isolated
Solar PV system	Rated power Power range	12 kW 0–12 kW
BESS	Technology Total capacity Minimum SoC Max charging power Max discharging power Bus connection Estimated life Batt cost	Lithium ion 53.9 kWh 10%, 5.39 kWh 26.6 kW 53.9 kW Direct connection/no converter 5000 cycles 500 €/kWh
EV charger	Rated power Efficiency	50 kW 95%
Hydrogen system/fuel cell	Max power Min power Efficiency	100 kW 10 kW 50–60%
Hydrogen storage	Capacity	41 kg 200 bar
Hydrogen system/electrolyser	No electrolyser	Hydrogen refuelling from external sources

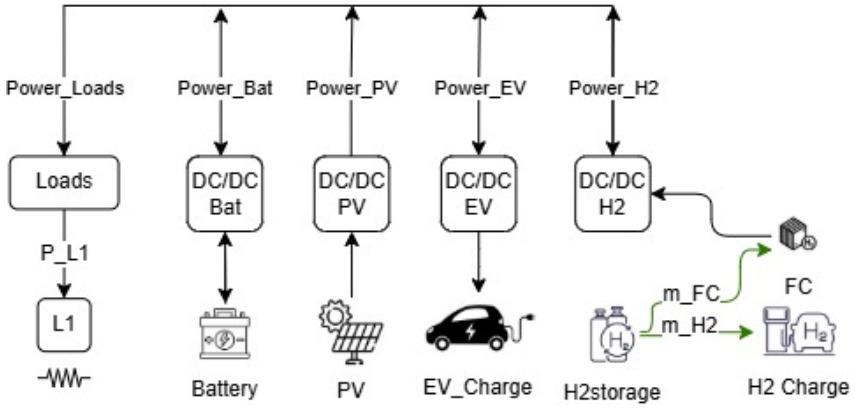


Figure 4. Basic diagram of the isolated microgrid used in the paper (reduction of the general schema in Figure 3).

3. Results

This section presents the experimental results and validation of the proposed compute-aware time-mesh design procedure on the selected microgrid case study. The results are reported by comparing (i) the predefined pool of market baseline configurations and (ii) the additional candidates generated by the Greedy-VoI search. Performance is quantified through comparative KPIs that jointly capture operational quality and computational practicality: optimization results and execution time.

The algorithm described has been applied to the optimization scenario described in previous paragraphs, in which, as indicated, the objective function is to minimize the operating costs of an isolated electric and hydrogen vehicle charging station. The objective function aims to reduce costs, so a negative value implies positive revenue.

As shown in Table 4, the solution [(4x15), (23x60), (24x120); 4320] not only yields the best objective function value but also has the shortest execution time among the three-day solutions.

Table 4. Predefined pool of solutions evaluation for a 1 h market.

Market Step	H [Horizon]	Solution	n_1	Δt_1	n_2	Δt_2	n_3	Δt_3	$[(n_1 x \Delta t) \ldots (n_B x \Delta t_B); H]$	Optimization Result [€]	Execution Time [s]
1 h	3 days 72 h 4320 min	0/reference	72	60					[(72x60); 4320]	−110.386	0.095
	3 days 72 h 4320 min	1	4	15	71	60			[(4x15), (71x60); 4320]	−110.386	0.125
	3 days 72 h 4320 min	2	4	15	23	60	24	120	[(4x15), (23x60), (24x120); 4320]	−110.793	0.04
	2 days 48 h 2880 min	3	4	15	47	60			[(4x15), (47x60); 2880]	−96.318	0.036
	2 days 48 h 2880 min	4	4	15	23	60	12	120	[(4x15), (23x60), (12x120); 2880]	−96.563	0.026

It can therefore be observed that the ideal solution from the pool of predefined solutions, [(4x15), (23x60), (24x120); 4320], improves the results of a basic solution for an hourly market, [(72x60); 4320]: the optimum is 0.37% better, but the execution time is 42.11% of the resolution time of the reference solution.

As shown in Table 4 and Figures 5 and 6, the best solution among the ones proposed in the predefined solution pool, is also the best solution for an optimization horizon of 3 days and is the one identified as [(4x15), (23x60), (24x120); 4320], while for a 2-day optimization horizon, it is identified as [(4x15), (23x60), (12x120); 2880].

Figure 6 confirms the fact that a greater number of steps implies a longer execution time. No direct relationship is observed between computational cost and the number of steps.

This phenomenon is also evident in Figure 7, which displays the results of applying the Greedy-VoI algorithm to this use case and hourly market. It can be seen that optimization horizons with more steps imply longer execution times, although no direct relationship is observed. It should be noted that to generate the 300 solutions evaluated and shown in Figure 7, the additional monotone-coarsening constraint in Equation (10) was not enforced, and no maximum block limit was imposed, to explore a broader set of feasible meshes.

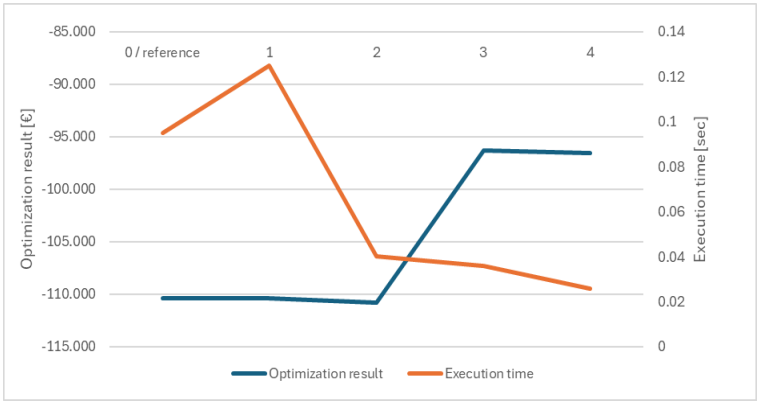


Figure 5. Predefined pool of solutions evaluation for 1 h market, optimization results vs. execution time (constraints: Equation (10) ON; maximum block limit ON).

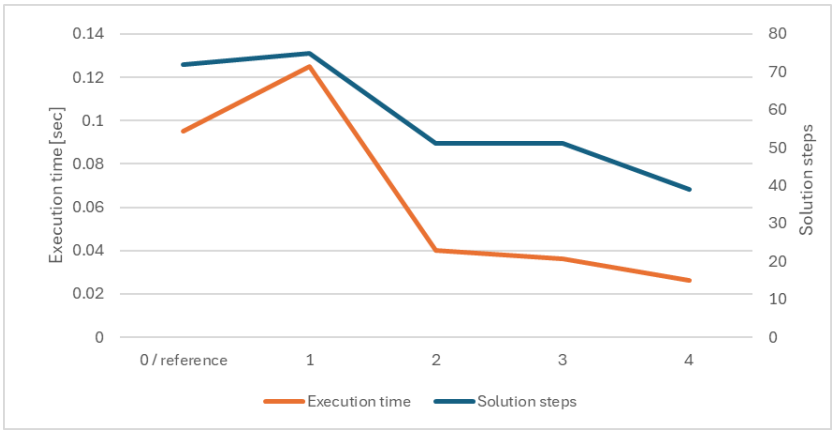


Figure 6. Predefined pool of solutions evaluation for 1 h market, execution time vs. solution steps (constraints: Equation (10) ON; maximum block limit ON).

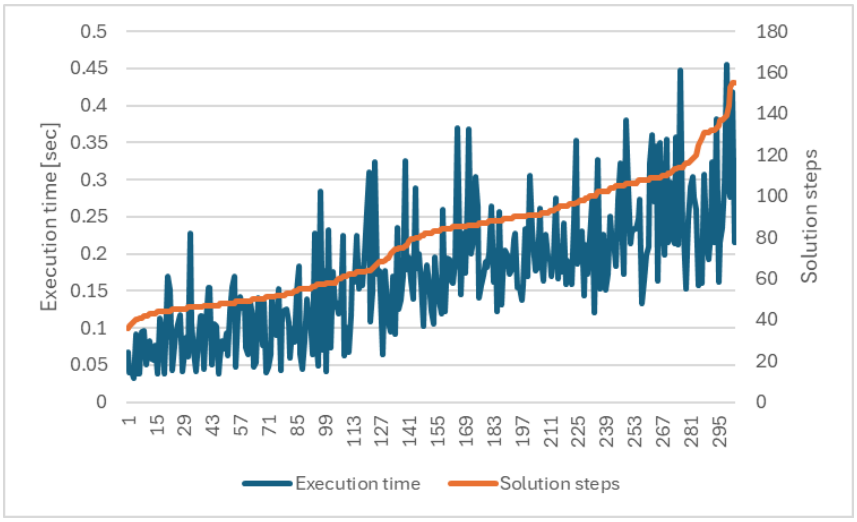


Figure 7. Results of applying the Greedy-VoI algorithm without the restriction of Equation (10) and without maximum block limits (constraints: Equation (10) OFF; maximum block limit OFF).

As indicated in Section 2, after evaluating the predefined solution pool, the Greedy-VoI algorithm is applied to search for a better solution, yielding 300 solutions. Among the better solutions obtained, the next are obtained because they are the best for one, two, and three blocks, starting with 5 min, 15 min, and 1 h steps:

- Three-day optimization horizon, one block in 2 h steps (H4320 | B1 | [36x120])
  - Optimization result: −110.891 €
  - Execution time: 0.067 sec (36 steps)
- Three-day optimization horizon, two blocks in 1 h and 2 h steps (H4320 | B2 | [2x60;35x120])
  - Optimization result: −110.891 €
  - Execution time: 0.04 sec (37 steps)
- Three-day optimization horizon, three blocks in 15 min, 1 h, and 2 h steps (H4320 | B3 | [4x15;1x60;35x120])
  - Optimization result: −110.891 €
  - Execution time: 0.039 sec (40 steps)
- Three-day optimization horizon, three blocks in 5 min, 15 min, and 2 h steps (H4320 | B3 | [3x5;7x15;35x120])
  - Optimization result: −110.891 €
  - Execution time: 0.055 sec (45 steps)

These solutions have very similar objective values (optimization results), and differences must be examined beyond the fourth decimal place, which may be due to discretization and granularity adaptation errors in the input signals. These solutions slightly improve the objective function by 0.09% relative to the best solution in the predefined solution pool (−110.793 €) and maintain similar overall execution times.

It is important to note that the best solutions proposed by the Greedy-VoI algorithm, in addition to improving the objective value, maintain a much shorter execution time compared to slower solutions of the 300 solutions provided by the algorithm:

- Three-day optimization horizon, three blocks in 5 min, 1 h, and 2 h steps (H4320 | B3 | [24x5;8x60;31x120])
  - Optimization result: −110.890 €
  - Execution time: 0.153 s (63 steps)
- Three-day optimization horizon, two blocks in 5 min and 2 h steps (H4320 | B2 | [48x5;34x120]])
  - Optimization result: −110.890 €
  - Execution time: 0.16 s (82 steps)
- Three-day optimization horizon, two blocks in 1 h and 2 h steps (H4320 | B2 | [48x5;34x120]
  - Optimization result: −110.891 €
  - Execution time: 0.228 s (46 steps)

As can be seen, despite having similar "optimization results" to those of faster execution solutions, execution times are up to 5 times longer.

It is important to highlight the stability of the proposed solutions and, consequently, the consistency of the proposed method. Interesting details can be observed after a detailed analysis of the 300 solutions:

- Optimization result [€]:
  - Minimum: −110.89142
  - Average: −110.85491
  - Maximum: −110.38627
  - Standard deviation: 0.07465
- Execution time [s]:
  - Minimum: 0.032

- Average: 0.105
- Maximum: 0.228
- Standard deviation: 0.0541

Across all evaluated configurations for the analyzed use case, the optimization outcomes are stable while execution times are more variable:

- The objective results reach a minimum of −110.8914 € and a maximum of −110.38627 € with a mean of −110.8549 € and a standard deviation of 0.0747 €, which corresponds to a coefficient of variation of ~0.067% (0.0747/110.8549). This indicates a very close clustering around the best solution; most candidates deliver near-identical economic performance.
- Execution times range from 0.032 s to 0.284 s, with a mean of 0.105 s and a standard deviation of 0.054 s, i.e., a coefficient of variation of ~51% (0.054/0.105). This larger relative spread is consistent with computational effort being more sensitive to discretization size and mesh structure than the objective value itself.

The proposed workflow helps select solutions that achieve good optimization results with lower, more predictable execution times, thereby confirming that the method is stable and consistent.

When analyzing the pool of predefined solutions for 15-min markets (Table 5), similar trends can be observed to those of the predefined solutions for hourly markets. The solution that combines 5 min, 15 min and hourly blocks in a 3-day horizon, [(3x5), (95x15), (48x60); 4320], provides a better solution (−110.424 €) with the shortest execution time (0.666 s) for this optimization horizon (3 days/4320 min), compared to a homogeneous solution based on market inputs, [(288x15); 4320]. Shorter optimization horizons yield faster results but worse optimization results.

Table 5. Predefined pool of solutions evaluation for a 15-min market.

Market Step	H [Horizon]	Solution	n_1	Δt_1	n_2	Δt_2	n_3	Δt_3	$[(n_1 x \Delta t) \ldots (n_B x \Delta t_B); H]$	Optimization Result [€]	Execution Time [s]
15 min	3 days 72 h 4320 min	0/reference	288	15					[(288x15); 4320]	−99.596	1.06
	3 days 72 h 4320 min	1	3	5	287	15			[(3x5), (287x15); 4320]	−99.596	1.264
	3 days 72 h 4320 min	2	3	5	95	15	48	60	[(3x5), (95x15), (48,60); 4320]	−110.424	0.666
	2 days 48 h 2880 min	3	3	5	191	15			[(3x5), (191x15); 2880]	−57.200	0.289
	2 days 48 h 2880 min	4	3	5	95	15	24	60	[(3x5), (95x15), (24x60); 2880]	−70.96	0.189

The use of the Greedy-VoI algorithm provides an improved result:

- Three-day optimization horizon, 3 blocks in 5 min, 1 h, and 2 h steps (H4320 | B3 | [12x5;31x60;20x120]
  - Optimization result: −110.793 €
  - Execution time: 0.275 s (63 steps)

4. Discussion

This section discusses the results from an engineering and deployment-oriented perspective, focusing on how the time-mesh design (number of steps and block structure) and the optimization horizon affect economic performance and computational effort under market-alignment constraints.

A key practical insight is that temporal granularity directly controls two error sources. First, the discretization error decreases as Δt is reduced; second, a finer mesh reduces forecast-induced error in the optimization layer. However, this improved fidelity comes at the cost of a larger optimization problem (more steps $\rightarrow$ more variables/constraints), where execution time increases with the number of steps. Even if the present case study does not show critical runtimes, this scaling behaviour becomes decisive for larger formulations (more assets, more binaries, additional network constraints) or for embedded/edge EMS platforms with limited computing capabilities.

From a scalability and deployment perspective, the proposed workflow is mainly independent of the specific microgrid case study, optimization solver, or execution platform, as it operates as an external design layer that evaluates candidate $\langle$B, H$\rangle$ configurations through a common EMS pipeline. When applied to more complex EMS formulations (e.g., larger MILPs with tighter binary couplings and/or explicit network power-flow constraints) longer and more variable solve times are expected; nevertheless, the method should remain applicable because real-time feasibility is treated as an explicit design constraint via the running-cycle budget $T \leq T_{\max}$. The same rationale holds for the execution platform: the approach can be deployed on edge devices capable of running Python and the required libraries, where compute resources are limited and runtime budgets are most critical. In fact, the ability to steer the mesh–horizon design towards configurations that respect strict compute budgets makes the proposed workflow particularly relevant for embedded/edge EMS deployments, where conservative runtime criteria (e.g., high-percentile solve times), strict time limits, and fallback policies (e.g., reusing the last feasible set-points) are essential to ensure deadline compliance.

Within this context, Figure 7 illustrates the value of a variable-resolution design: Greedy-VoI explores market-valid neighbours that reallocate resolution across the horizon (e.g., refining near-term blocks where actions are executed and coarsening the tail where decisions mainly provide boundary guidance). This produces configurations that can improve (or match) the objective while explicitly controlling the number of steps and, consequently, solver time. From a deployment perspective, the primary outcome is not a single "best Δt", but rather a systematic approach to identifying a mesh–horizon pair that achieves near-optimal economics while remaining compatible with the available compute budget and the market settlement structure.

The dispersion of the optimization results across the market-feasible mesh–horizon candidates is very small relative to the objective magnitude, which supports the stability and consistency of the proposed selection workflow. In particular, the objective exhibits a tight clustering (CV $\approx$ 0.067%), suggesting that multiple market-feasible meshes deliver equivalent economic performance. In contrast, execution time shows a higher spread (CV $\approx$ 51%), which aligns with the expected sensitivity of solver effort to discretization size (steps) and block structure rather than to the operating cost itself.

The proposed Greedy-VoI-based algorithm should be understood as a compute-aware heuristic local search guided by a cost–runtime (objective function–execution time) criterion. While classical approximation guarantees for greedy algorithms can be established under specific assumptions, no submodularity assumption is made or proven in the present mesh–horizon selection setting; therefore, no formal optimality bounds are claimed. Instead, Greedy-VoI is positioned as a practical and reproducible design tool: candidate

configurations are evaluated using a consistent rolling-horizon principle and a common set of KPIs and solution quality is assessed empirically against a benchmark pool and comparative runtime–performance evidence.

5. Conclusions

This work is motivated by a persistent gap in the deployment of EMS/MPC. While high temporal fidelity/low granularity can reduce forecast and discretization-induced errors and improve results, complex microgrid formulations become computationally prohibitive when solved repeatedly in rolling-horizon operation. In rolling-horizon control, only the first set-points are implemented before re-optimization, which means that the time discretization should be treated as a design variable rather than a fixed modelling choice: it must be sufficiently fine where actions are executed, and constraints are likely to bind, and increasingly coarse where decisions primarily serve as feasibility and boundary guidance.

Accordingly, this paper presents a general, market-aware methodology for co-designing the EMS time mesh and the optimization horizon under explicit real-time constraints. The approach formalizes each candidate configuration as a pair $\langle B, H \rangle$, introduces an auditable mesh signature to uniquely represent block schedules, and enforces practical feasibility conditions, such as admissible time steps, exact horizon tiling, and a per-cycle compute budget. Building on this formulation, the proposed workflow first benchmarks a pool of predefined market-consistent solutions. Then, it applies the Greedy-VoI procedure to generate and evaluate new candidates using the same KPIs, enabling the systematic identification of near-optimal discretization–horizon combinations for a given microgrid and market framework.

In the case study presented, the search yields multiple viable configurations on the market with very similar economic performance, although there are clear differences in terms of discretization size (i.e., the number of steps) and observed resolution time. This supports two practical conclusions. First, the methodology is effective as a design aid: it can systematically propose and rank deployable $\langle B, H \rangle$ options that balance objective quality with computational effort, avoiding ad hoc choices of Δt and horizon. Second, even when the optimization is not yet runtime-limited, treating the time mesh as a controllable design variable remains valuable because it provides a transparent way to anticipate scalability limits and to select conservative configurations for future expansions (additional assets, tighter binaries, network constraints) or for execution on constrained hardware. Overall, the proposed workflow operationalizes the intuition that the "best" EMS discretization is not necessarily the finest one, but the one that achieves near-optimal economics within market and compute constraints, with resolution concentrated where it yields the highest value of information. In addition, the proposed Greedy-VoI layer is deliberately positioned as a compute-aware heuristic, and its effectiveness is demonstrated empirically through a reproducible benchmark pool and consistent evaluation. Future work should examine whether stronger theoretical properties of the mesh–horizon selection problem can be established and exploited to derive approximation guarantees or to improve selection criteria.

A key outcome of the experimental validation is the high stability of the recommended mesh–horizon designs: across a large set of market-feasible candidates, the optimization results remain tightly concentrated around the best solution, while runtime varies substantially with the number of steps and the block structure. This indicates that, for the studied use case, the proposed workflow reliably finds multiple "economically equivalent" configurations and therefore enables a principled secondary selection based on computational constraints (deadline compliance) and implementation robustness. In practice, this supports the central claim of the paper: treating $\langle B, H \rangle$ as a design variable provides

an actionable way to maintain near-optimal operational performance while controlling real-time computational effort, which is particularly relevant as EMS formulations grow in size (more assets, binaries, and network constraints) or when deploying on compute-limited platforms.

Future work could extend this study along four complementary directions. First, the proposed Greedy-VoI procedure could be benchmarked against representative state-of-the-art alternatives for horizon/discretization design under the same KPI definitions and compute-budget constraints. Second, the generalisability of the approach could be assessed by applying it to a broader set of microgrid configurations, asset mixes, and operating contexts. Third, robustness under forecast uncertainty could be evaluated by analyzing how the selected ⟨granularity, horizon⟩ configurations change under forecast errors (PV, demand, prices) (e.g., via error models or scenario ensembles) and how this impacts closed-loop costs, feasibility, and deadline compliance. Finally, the current offline design could be reformulated as an online/adaptive layer that periodically re-tunes the recommended ⟨B, H⟩ as the microgrid, forecasts, market signals, or computational resources evolve.

Author Contributions: Conceptualization, G.F. and J.F.S.O.; methodology, G.F. and J.F.S.O.; software, G.F.; validation, G.F., and J.F.S.O.; formal analysis, G.F. and J.F.S.O.; investigation, G.F.; resources, G.F.; data curation, G.F.; writing—original draft preparation, G.F.; writing—review and editing, G.F., J.F.S.O., A.A., A.C. and M.T.; visualization, G.F., J.F.S.O., A.A., A.C. and M.T.; supervision, J.F.S.O.; project administration, G.F. All authors have read and agreed to the published version of the manuscript.

Funding: REEFLEX has received funding from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101096192.

Data Availability Statement: Data supporting the reported results are not publicly available due to confidentiality restrictions; no publicly archived datasets were generated in this study.

Acknowledgments: During the preparation of this study, the authors used [Chat GPT 5.1] for the purposes of checking the correct wording of some parts of the text, reviewing, and modifying some time series used in this work. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

EMS	Energy Management Systems
MPC	Model Predictive Control
DER	Distributed Energy Resource
Greedy-VoI	Greedy-Value-of-Information
PV	Photovoltaic
EV	Electric Vehicle
EDS	Exponential Decay of Sensitivity
MILP	Mixed Integer Linear Programming
ORoHS	Optimal Rolling-Horizon Strategy
UC	Unit Commitment

References

1. Fang, X.; Misra, S.; Xue, G.; Yang, D. Smart Grid—The New and Improved Power Grid: A Survey. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 944–980. [CrossRef]
2. Staffell, I.; Brett, D.; Brandon, N.; Hawkes, A. A review of domestic heat pumps. *Energy Environ. Sci.* **2012**, *5*, 9291–9306. [CrossRef]

3. Haque, M.M.; Wolfs, P. A review of high PV penetrations in LV distribution networks: Present status, impacts and mitigation measures. *Renew. Sustain. Energy Rev.* **2016**, *62*, 1195–1208. [CrossRef]
4. Shen, F.; Huang, S.; Wu, Q.; Repo, S.; Xu, Y.; Østergaard, J. Comprehensive Congestion Management for Distribution Networks based on Dynamic Tariff, Reconfiguration and Re-profiling Product. *IEEE Trans. Smart Grid* **2019**, *10*, 4795–4805. [CrossRef]
5. Lund, P.D.; Mikkola, J.; Salpakari, J.; Lindgren, J. Review of energy system flexibility measures to enable high levels of variable renewable electricity. *Renew. Sustain. Energy Rev.* **2015**, *45*, 785–807. [CrossRef]
6. Denholm, P.; Hand, M. Grid flexibility and storage required to achieve very high penetration of variable renewable electricity. *Energy Policy* **2011**, *39*, 1817–1830. [CrossRef]
7. Palensky, P.; Dietrich, D. Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads. *IEEE Trans. Ind. Inform.* **2011**, *7*, 381–388. [CrossRef]
8. Meng, L.; Sanseverino, E.R.; Hernandez, A.C.L.; Dragicevic, T.; Vasquez, J.C.; Guerrero, J.M. Microgrid supervisory controllers and energy management systems: A literature review. *Renew. Sustain. Energy Rev.* **2016**, *60*, 1263–1273. [CrossRef]
9. Parag, Y.; Sovacool, B.K. Electricity market design for the prosumer era. *Nat. Energy* **2016**, *1*, 16032. [CrossRef]
10. Parrish, B.; Heptonstall, P.; Gross, R.; Sovacool, B.K. A systematic review of motivations, enablers and barriers for consumer engagement with residential demand response. *Energy Policy* **2020**, *138*, 111221. [CrossRef]
11. Fernández, G.; Hedar, A.S.; Torres, M.; Apostolidou, N.; Koltsaklis, N.; Spiliopoulos, N. System Requirements for Flexibility Markets Participation: A Stakeholder-Centric Survey from REEFLEX Project. *Appl. Sci.* **2025**, *15*, 10426. [CrossRef]
12. Afram, A.; Janabi-Sharifi, F. Theory and applications of HVAC control systems—A review of model predictive control (MPC). *Build. Environ.* **2014**, *72*, 343–355. [CrossRef]
13. Parisio, A.; Rikos, E.; Glielmo, L. A Model Predictive Control Approach to Microgrid Operation Optimization. *IEEE Trans. Control. Syst. Technol.* **2014**, *22*, 1813–1827. [CrossRef]
14. Mayne, D.Q.; Rawlings, J.B.; Rao, C.V.; Scokaert, P.O.M. Constrained model predictive control: Stability and optimality. *Autom.* **2000**, *36*, 789–814. [CrossRef]
15. Qin, S.J.; Badgwell, T.A. A survey of industrial model predictive control technology. *Control. Eng. Pract.* **2003**, *11*, 733–764. [CrossRef]
16. Oldewurtel, F.; Parisio, A.; Jones, C.N.; Gyalistras, D.; Gwerder, M.; Stauch, V.; Lehmann, B.; Morari, M. Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy Build.* **2012**, *45*, 15–27. [CrossRef]
17. Spantideas, S.T.; Giannopoulos, A.E.; Trakadas, P. Autonomous Price-Aware Energy Management System in Smart Homes via Actor-Critic Learning with Predictive Capabilities. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 15018–15033. [CrossRef]
18. Hoffmann, M.; Kotzur, L.; Stolten, D.; Robinius, M. A Review on Time Series Aggregation Methods for Energy System Models. *Energies* **2020**, *13*, 641. [CrossRef]
19. Hu, J.; Shan, Y.; Guerrero, J.M.; Ioinovici, A.; Chan, K.W.; Rodriguez, J. Model predictive control of microgrids—An overview. *Renew. Sustain. Energy Rev.* **2021**, *136*, 110422. [CrossRef]
20. Marquant, J.F.; Evins, R.; Carmeliet, J. Reducing Computation Time with a Rolling Horizon Approach Applied to a MILP Formulation of Multiple Urban Energy Hub System. *Procedia Comput. Sci.* **2015**, *51*, 2137–2146. [CrossRef]
21. Hönen, J.; Hurink, J.L.; Zwart, B. Dynamic Rolling Horizon-Based Robust Energy Management for Microgrids Under Uncertainty. *arXiv* **2023**, arXiv:2307.05154. [CrossRef]
22. Hoffmann, M.; Priesmann, J.; Nolting, L.; Praktiknjo, A.; Kotzur, L.; Stolten, D. Typical periods or typical time steps? A multi-model analysis to determine the optimal temporal aggregation for energy system models. *Appl. Energy* **2021**, *304*, 117825. [CrossRef]
23. Hoffmann, M.; Kotzur, L.; Stolten, D. The Pareto-optimal temporal aggregation of energy system models. *Appl. Energy* **2022**, *315*, 119029. [CrossRef]
24. Gao, Z.; Gazzani, M.; Tejada-Arango, D.A.; Siqueira, A.S.; Wang, N.; Gibescu, M.; Morales-España, G. Fully flexible temporal resolution for energy system optimization. *Appl. Energy* **2025**, *396*, 126267. [CrossRef]
25. Forough, A.B.; Roshandel, R. Multi objective receding horizon optimization for optimal scheduling of hybrid renewable energy system. *Energy Build.* **2017**, *150*, 583–597. [CrossRef]
26. Zhang, N.; Yan, J.; Hu, C.; Sun, Q.; Yang, L.; Gao, D.W.; Guerrero, J.M.; Li, Y. Price-Matching-Based Regional Energy Market with Hierarchical Reinforcement Learning Algorithm. *IEEE Trans. Ind. Inform.* **2024**, *20*, 11103–11114. [CrossRef]
27. Shin, S.; Zavala, V.M. Controllability and Observability Imply Exponential Decay of Sensitivity in Dynamic Optimization. *IFAC-PapersOnLine* **2021**, *54*, 179–184. [CrossRef]
28. Shin, S.; Zavala, V.M. Diffusing-Horizon Model Predictive Control. *IEEE Trans. Autom. Control* **2023**, *68*, 188–201. [CrossRef]
29. Wang, Z.; Tao, H.; Cai, W.; Duan, Y.; Wu, D.; Zhang, L. Study on the multitime scale rolling optimization operation of a near-zero energy building energy supply system. *Energy Convers. Manag.* **2022**, *270*, 116255. [CrossRef]
30. Wei, S.; Gao, X.; Zhang, Y.; Li, Y.; Shen, J.; Li, Z. An improved stochastic model predictive control operation strategy of integrated energy system based on a single-layer multi-timescale framework. *Energy* **2021**, *235*, 121320. [CrossRef]

31. Behrunani, V.N.; Cai, H.; Heer, P.; Smith, R.S.; Lygeros, J. Distributed multi-horizon model predictive control for network of energy hubs. *Control. Eng. Pract.* **2024**, *147*, 105922. [CrossRef]
32. Abdulla, K.; De Hoog, J.; Steer, K.; Wirth, A.; Halgamuge, S. Multi-Resolution Dynamic Programming for the Receding Horizon Control of Energy Storage. *IEEE Trans. Sustain. Energy* **2019**, *10*, 333–343. [CrossRef]
33. Yu, Z.; Zhong, H.; Ruan, G.; Yan, X. Network-Constrained Unit Commitment with Flexible Temporal Resolution. *IEEE Trans. Power Syst. Early Access* **2024**, *1*, 73–84. [CrossRef]
34. Tao, J.; Li, R.; Pineda, S. Unit Commitment with Cost-Oriented Temporal Resolution. *arXiv* **2025**, arXiv:2506.02707. [CrossRef]
35. Marcy, C.; Goforth, T.; Nock, D.; Brown, M. Comparison of temporal resolution selection approaches in energy systems models. *Energy* **2022**, *251*, 123969. [CrossRef]
36. Aguilar, D.; Quinones, J.; Pineda, L.; Ostanek, J.; Castillo, L. Optimal scheduling of renewable energy microgrids: A robust multi-objective approach with machine learning-based probabilistic forecasting. *Appl. Energy* **2024**, *369*, 123548. [CrossRef]