Multi-Stage Scenario Reduction for Optimal Energy Storage Operation in Real-Time Markets

1st Robin Bruneel
Department of Electrical Engineering
ESAT-STADIUS
Leuven, Belgium
robin.bruneel@kuleuven.be

2nd Panagiotis Patrinos
Department of Electrical Engineering
ESAT-STADIUS
Leuven, Belgium
panos.patrinos@kuleuven.be

***Abstract*—The growing share of renewable energy sources in the electricity grid has led to increased volatility in real-time balancing prices. While this volatility creates valuable opportunities for energy storage systems (ESS), the inherent uncertainty of future prices significantly complicates the optimization process. In this paper, we propose a multi-stage scenario-based stochastic model predictive control (SMPC) strategy to optimize the operation of ESS in real-time balancing markets. Central to our approach is a stagewise refinement algorithm, inspired by optimal transport theory. This algorithm constructs non-symmetric multi-stage scenario trees through clustering methods and is benchmarked in a simulation on the Belgian real-time balancing market with a battery energy storage system. Results demonstrate that our approach, particularly when combined with k-means clustering, yields higher profits compared to traditional multi-stage reduction techniques.**

***Index Terms*—Clustering algorithms, Energy storage systems, Model predictive control, Real-time electricity markets, Scenario reduction, Stochastic optimization**

I. Introduction

In recent years, renewable energy sources have gained a substantial share in the energy mix. While this shift represents a positive move toward more sustainable energy systems and potentially lower electricity prices, the inherent unpredictability of renewables also contributes to larger grid imbalances and increased volatility in real-time balancing prices. Energy storage systems (ESS) have emerged as valuable assets to benefit from this volatility while supporting grid stability. However, the inherent uncertainty of future prices makes the optimal operation of ESS in real-time markets a complex task.

To address this challenge, we propose a multi-stage scenario-based stochastic model predictive control (SMPC) strategy. A key element of our approach is the construction of a scenario tree that effectively captures the multi-stage uncertainty in future price trajectories. To maintain computational tractability, scenario reduction is typically applied as a preprocessing step to limit the size of the scenario tree. A commonly used technique is the moment-matching approach, which generates a reduced scenario set by preserving selected statistical moments of the original probability distribution [1]. However, as highlighted by Hochreiter and Pflug, moment-matching can sometimes produce counterintuitive and unreliable results [2]. As an alternative, they suggest minimizing a probability metric between the original and reduced scenario set to obtain a more optimal reduction. Although solving this problem exactly is NP-hard, it can be efficiently approximated using heuristic algorithms. Notable methods include greedy approaches such as the forward selection and backward elimination algorithm, initially introduced by Dupacova et al. [3] and later extended by Heitsch and Römisch [4], and clustering-based algorithms such as the k-means method [5], [6].

While these techniques are well established for two-stage problems, their extension to multi-stage problems remains relatively underexplored. In multi-stage problems, a common strategy is to construct a symmetric scenario tree with a fixed branching structure [5]–[8]. However, this approach suffers from a combinatorial explosion in the number of scenarios as the prediction horizon increases. To address this, Heitsch and Römisch introduced a multi-stage forward selection algorithm that incrementally adds branches in the regions with the largest approximation error. This procedure generates a non-symmetric tree with fewer nodes that still closely approximates the original distribution [9]. In this paper, we further refine this methodology. The main contributions of this paper are:

- We propose a stagewise refinement algorithm that constructs a non-symmetric multi-stage scenario tree using clustering-based techniques.
- We evaluate and compare the performance of various scenario reduction strategies through a battery energy storage system simulation in the Belgian real-time balancing market.

The remainder of this paper is organized as follows. Section II outlines the scenario generation starting from a probabilistic real-time price model. Section III introduces various reduction techniques and details the stagewise refinement algorithm. In Section IV, we present a case study involving a simulated battery energy storage system in the Belgian real-time market. Finally, Sections V and VI offer a discussion of the results and summarize the main conclusions of the paper.

This work was supported by Flanders Innovation & Entrepreneurship (VLAIO) under grant code "Baekeland Mandaat HBC.2022.0695", Research Foundation Flanders (FWO) research projects G081222N, G033822N, and G0A0920N, and Research Council KUL grant C14/24/103.

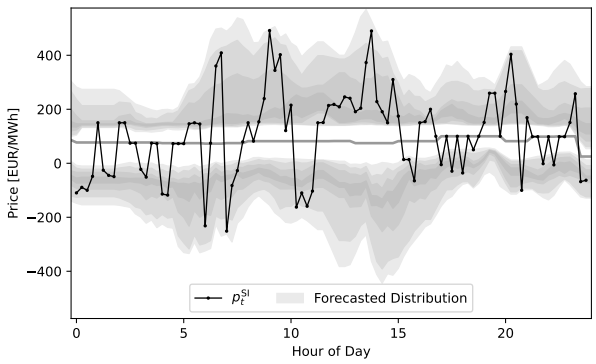


Fig. 1. Forecasted distribution of the Belgian imbalance price p_t^{SI} on January 7, 2025, based on the mixture model proposed in [11]. The observed imbalance prices fall well within the predicted distribution range.

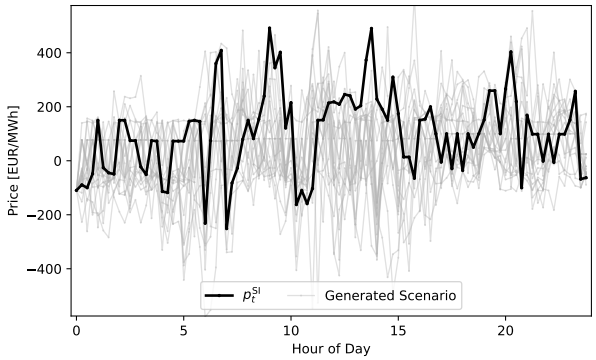


Fig. 2. Generated scenarios from the forecasted distribution of the Belgian imbalance price p_t^{SI} on January 7, 2025, using the sampling technique proposed by Pinson et al. [12].

II. Scenario Generation

An essential component of the proposed control strategy is generating future price trajectories. In the case study, we focus on the Belgian system imbalance price, a real-time price settled every 15 minutes reflecting the activation costs of balancing reserves [10]. This price exhibits pronounced regime-switching behavior, which we capture in the scenario generation process using the probabilistic mixture model introduced in [11]. The resulting forecasted price distribution is illustrated in Fig. 1.

While this model provides a distribution for each individual timestep, it does not account for the temporal dependence between consecutive stages. To generate realistic price trajectories, we adopt the method proposed by Pinson et al. in [12], which introduces time dependence by modeling the correlation of normalized residuals. The generated scenarios, visualized in Fig. 2, are statistically consistent with our forecasted distribution.

III. Scenario Reduction

In the previous section, we outlined a methodology to generate price scenarios. We now shift our focus to the optimal construction of a multi-stage scenario tree. We begin by discussing several scenario reduction techniques, which serve as a foundation for the stagewise refinement algorithm introduced in the following section.

A. Scenario Reduction Techniques

As indicated by Pflug [7], the optimal scenario reduction problem can be formulated as a Monge-Kantorovich mass transportation problem, where the goal is to find a reduced scenario set that minimizes the Kantorovich distance to the original distribution. Let $(\Omega, \mathcal{F}, Q)$ and $(\Omega', \mathcal{F}', Q')$ be discrete probability spaces representing the original and reduced scenario sets, where $\Omega = \{\boldsymbol{\omega}_i\}_{i=1}^N$ with associated probabilities $\pi_i = Q(\{\boldsymbol{\omega}_i\})$, and $\Omega' = \{\boldsymbol{\omega}'_j\}_{j=1}^M$ with reduced probabilities π'_j. The distance between these two discrete probability distributions is measured with the Kantorovich distance [13]:

$$\begin{aligned} D_{\mathrm{K}}(Q, Q') = &\min_{\eta} \sum_{\boldsymbol{\omega}_i \in \Omega} \sum_{\boldsymbol{\omega}'_j \in \Omega'} c(\boldsymbol{\omega}_i, \boldsymbol{\omega}'_j)\, \eta(\boldsymbol{\omega}_i, \boldsymbol{\omega}'_j) \\ \text{s.t.} \quad &\forall \boldsymbol{\omega}_i \in \Omega, \boldsymbol{\omega}'_j \in \Omega' : \eta(\boldsymbol{\omega}_i, \boldsymbol{\omega}'_j) \geq 0, \\ &\forall \boldsymbol{\omega}_i \in \Omega : \sum_{\boldsymbol{\omega}'_j \in \Omega'} \eta(\boldsymbol{\omega}_i, \boldsymbol{\omega}'_j) = \pi_i, \\ &\forall \boldsymbol{\omega}'_j \in \Omega' : \sum_{\boldsymbol{\omega}_i \in \Omega} \eta(\boldsymbol{\omega}_i, \boldsymbol{\omega}'_j) = \pi'_j \end{aligned} \tag{1}$$

Here, $c : \Omega \times \Omega' \to \mathbb{R}_+$ is a cost function that quantifies the distance between individual scenarios. This formulation calculates the minimal "cost" required to redistribute the original probability mass onto a smaller scenario set. Provided that the support of Ω' is sufficiently sparse (i.e. $M \ll N$), the Kantorovich distance can be approximated as [14]:

$$D_{\mathrm{K}}(Q, Q') = \sum_{\boldsymbol{\omega}_i \in \Omega} \pi_i \min_{\boldsymbol{\omega}'_j \in \Omega'} c(\boldsymbol{\omega}_i, \boldsymbol{\omega}'_j) \tag{2}$$

Under this approximation, the probability mass of each original scenario $\boldsymbol{\omega}_i$ is fully assigned to its nearest reduced scenario $\boldsymbol{\omega}'_j$. Constructing the reduced scenario set involves minimizing the Kantorovich distance, which is then equivalent to solving the following optimization problem:

$$\begin{aligned} \min_{Q', \Omega'} D_{\mathrm{K}}(Q, Q') &= \min_{Q', \Omega'} \sum_{\boldsymbol{\omega}_i \in \Omega} \pi_i \min_{\boldsymbol{\omega}'_j \in \Omega'} c(\boldsymbol{\omega}_i, \boldsymbol{\omega}'_j) \\ &= \min_{Q', \Omega'} \sum_{C_j \in \boldsymbol{\mathcal{C}}} \sum_{i \in C_j} \pi_i\, c(\boldsymbol{\omega}_i, \boldsymbol{\omega}'_j) \end{aligned} \tag{3}$$

This formulation corresponds to the well-known clustering problem, where the objective is to partition the original scenarios into M clusters C_j such that the total cost of assigning each scenario $\boldsymbol{\omega}_i$ to its corresponding cluster center $\boldsymbol{\omega}'_j$ is minimized. Here, $\boldsymbol{\mathcal{C}}$ is the collection of clusters formed by assigning each original scenario $\boldsymbol{\omega}_i$ to the nearest reduced scenario $\boldsymbol{\omega}'_j$:

$$\boldsymbol{\mathcal{C}} = \left\{ C_j = \left\{ i \,\middle|\, c(\boldsymbol{\omega}_i, \boldsymbol{\omega}'_j) = \min_k c(\boldsymbol{\omega}_i, \boldsymbol{\omega}'_k) \right\} \right\}_{j=1}^{M}$$

These clusters form a partition of the index set $\{1, \ldots, N\}$, such that

$$\bigcup_{j=1}^{M} C_j = \{1, \ldots, N\}, \quad C_j \cap C_{j'} = \emptyset \quad \text{for } j \neq j'.$$

Each reduced scenario $\boldsymbol{\omega}'_j$ is assigned an aggregated probability to ensure that the total probability mass of the original distribution is preserved:

$$\pi'_j = \sum_{i \in C_j} \pi_i.$$

While the general clustering problem is NP-hard, more efficient solution methods may be available depending on the choice of the cost function c. We now discuss several practical approaches:

1) Fast Forward Selection: The fast forward selection algorithm is a greedy scenario reduction approach that iteratively selects the scenario whose inclusion leads to the largest reduction in Kantorovich distance [3], [4]. This method is general and can be used with any cost function c.

2) K-means Clustering: A special case of the scenario reduction problem is when the cost function c is the squared Euclidian distance. In that specific case, Eq. (3) is equivalent to the k-means clustering problem [15]. Although k-means clustering does not guarantee an optimal solution, it converges quickly and is widely used in practice.

3) K-medians Clustering: An alternative to k-means clustering is the k-medians algorithm, which minimizes the sum of Manhattan distances rather than the squared Euclidean distances [16]. Compared to k-means, the clusters produced by k-medians are generally more robust to outliers.

4) K-medoids Clustering: Another commonly used clustering method is the k-medoids algorithm, in which each cluster center (medoid) is required to be an actual data point from the original dataset [17]. Unlike k-means or k-medians, it can be applied for any cost function c.

5) Hierarchical Agglomerative Clustering: Hierarchical agglomerative clustering (HAC) when combined with the Ward linkage criterion, serves as a greedy approximation to the k-means problem and can therefore also be used for scenario reduction [18].

B. Stagewise Refinement Algorithm

The previous section introduced several scenario reduction techniques grounded in optimal transport theory. In this section, we present a stagewise refinement algorithm for constructing non-symmetric multi-stage scenario trees using these methods. Our approach builds upon the multi-stage forward selection algorithm introduced by Heitsch and Römisch [9, Algorithm 4.5]. Their method incrementally introduces new branches in a stage-by-stage manner. At each stage, the algorithm iteratively selects the scenario from the original set whose inclusion yields the largest reduction in approximation error for that stage. As a greedy selection process, it does not reevaluate or modify scenarios once they have been selected within a stage.

Algorithm 1 Stagewise Refinement Algorithm

Input: $\Omega = \left\{\boldsymbol{\omega}_k \in \mathbb{R}^T\right\}_{k=1}^{N}, Q$
Output: $\Omega' = \left\{\boldsymbol{\omega}'_k \in \mathbb{R}^T\right\}_{k=1}^{mT+1}, Q'$
$\boldsymbol{\mathcal{C}}_0 \leftarrow \{\{1, \ldots, N\}\}$
for each stage $t = 1, \ldots, T$ **do**
  $\boldsymbol{\mathcal{C}}_t \leftarrow \boldsymbol{\mathcal{C}}_{t-1}$
  for $r = 1, \ldots, m$ **do**
    Let $\boldsymbol{\mathcal{C}}_t(C)$ denote the child subpartition of cluster $C \in \boldsymbol{\mathcal{C}}_{t-1}$ at stage t. That is, the set of clusters in $\boldsymbol{\mathcal{C}}_t$ that are descendants of C.
    Identify parent cluster with the highest approximation error:

$$C^* \leftarrow \operatorname*{argmax}_{C \in \boldsymbol{\mathcal{C}}_{t-1}} \sum_{C_k \in \boldsymbol{\mathcal{C}}_t(C)} \sum_{i \in C} \pi_i \cdot c(\omega_{t,i}, \omega'_{t,k})$$

    Let $\boldsymbol{\mathcal{C}}_t(C^*) = \{C_i\}_{i=1}^{k}$ be the child subpartition of C^*.
    Refine $\boldsymbol{\mathcal{C}}_t(C^*)$ with a scenario reduction technique to obtain new clusters $\{\widetilde{C}_i\}_{i=1}^{k+1}$, centers $\{\widetilde{\omega}'_{t,i}\}_{i=1}^{k+1}$, and probabilities $\{\widetilde{\pi}'_{t,i}\}_{i=1}^{k+1}$
    Replace the child subpartition with the refined partition.

$$\boldsymbol{\mathcal{C}}_t(C^*) \leftarrow \{\widetilde{C}_i\}_{i=1}^{k+1}$$

  end for
end for
Termination:
Scenario tree is defined by $\boldsymbol{\mathcal{C}}_0, \boldsymbol{\mathcal{C}}_1, \ldots, \boldsymbol{\mathcal{C}}_T$
For all leaf nodes at stage T, backtrack through their parents to assemble complete scenario paths $\boldsymbol{\omega}'_k \in \mathbb{R}^T$, each having probability $\pi'_{T,k}$
Return reduced scenarios Ω' and probabilities Q'

Inspired by their work, we propose a more general stagewise refinement algorithm that integrates clustering techniques into the scenario tree construction process. The core idea is to traverse the tree in a stagewise fashion, iteratively refining the child subpartitions with the highest approximation error. Each refinement introduces a new branch to the selected subpartition using a scenario reduction technique, thereby systematically reducing the overall approximation error. A key advantage of our approach is that, unlike greedy methods which fix previously selected scenarios, our refinement strategy allows for reoptimization of the entire child subpartition.

In this paper, we fix the number of branches to be added at each stage to m. The stagewise refinement process is shown in Fig. 3 and formally described in Algorithm 1. The resulting scenario tree features a dynamic, non-symmetric branching structure. Fig. 4 illustrates an example of the multi-stage scenario tree structure generated in the case study.

IV. Case Study

This case study evaluates the performance of various multi-stage scenario reduction techniques on a simulated battery energy storage system (BESS) operating in the Belgian real-time market. The BESS has a power rating of $u_{\max} = 100$ kW, an energy capacity of $E_{\max} = 200$ kWh, an efficiency of $\nu = 97\%$, and a usage cost of $\mu = 30$ €/MWh to discourage low-profit operations and help preserve battery life. We consider a prediction horizon of $T = 40$, which corresponds to 10 hours in the future due to the 15-minute settlement interval of

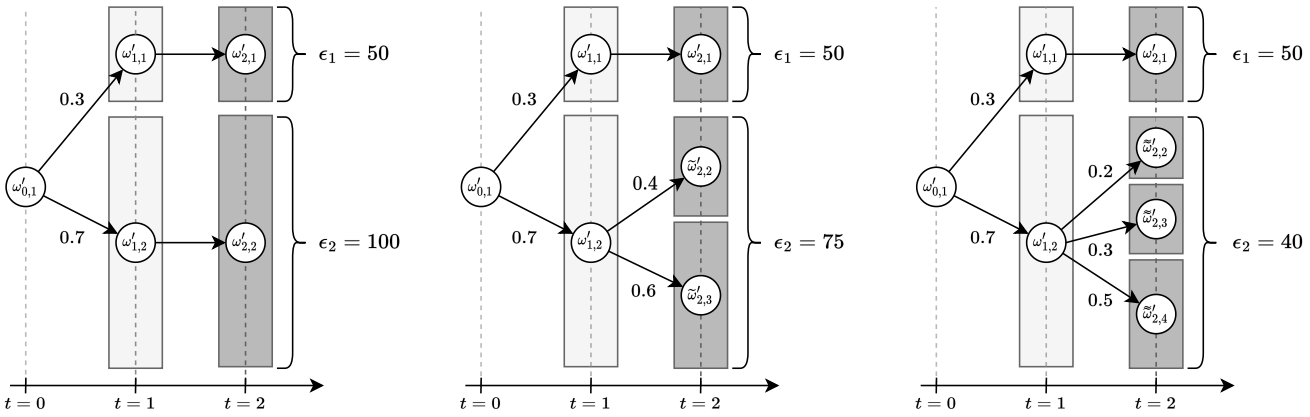


Fig. 3. Visualization of the stagewise refinement algorithm. At stage $t = 2$, the algorithm identifies the parent cluster with the highest approximation error ϵ_i and refines its child subpartition by adding a new branch. After each refinement, the reduced scenarios of the subpartition are updated and the approximation error is recalculated. This process is repeated until the desired number of inserted branches is reached. The algorithm then progresses to the next stage ($t = 3$) and applies the same procedure.

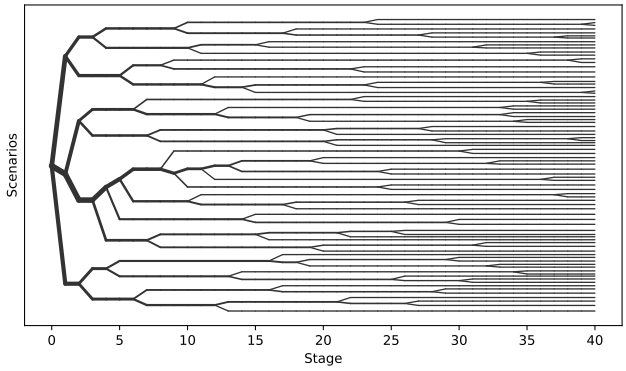


Fig. 4. Visualization of the branching structure of a reduced multi-stage scenario tree generated in the case study (using k-means reduction with $m = 2$). The stagewise refinement algorithm branches less frequently in branches having a lower probability of occurring.

the imbalance price. For each optimization cycle, $N = 2000$ scenario vectors are generated.

The clustering-based methods are benchmarked against the multi-stage fast-forward selection (FFS) algorithm proposed by Heitsch and Römisch [9]. The optimization is implemented as a rolling horizon model predictive control (MPC) framework: every 15 minutes, a new scenario tree is generated, the problem is re-solved, and the first control action is executed. The stochastic optimization problem is solved using the CLARABEL solver [19] on a Macbook Pro M1 in Python. For the clustering-based scenario reductions, we use the pyclustering and scikit-learn libraries [20], [21]. The dataset spans from February 1, 2024 to February 1, 2025. This section first formulates the optimization problem and then presents the results of the case study.

A. Problem Formulation

We formulate a multi-stage stochastic optimization problem to minimize the expected energy cost of operating a battery energy storage system (BESS) over the prediction horizon:

$$\min_{\boldsymbol{u}^{\mathrm{c}},\boldsymbol{u}^{\mathrm{d}}} \quad \sum_{t=0}^{T}\sum_{i=1}^{M_t} \pi'_{t,i}\left[\omega'_{t,i}(u^{\mathrm{c}}_{t,i} - u^{\mathrm{d}}_{t,i}) + \mu(u^{\mathrm{c}}_{t,i} + u^{\mathrm{d}}_{t,i})\right]\Delta t + \sum_{i=1}^{N_T} \pi'_{T,i} V_T(x_{T,i}) \tag{4}$$

$$\text{s.t.} \quad x_{0,1} = x_0 \tag{5}$$

$$x_{t,i} = x_{t-1,j} + \left(\nu u^{\mathrm{c}}_{t-1,j} - \tfrac{1}{\nu} u^{\mathrm{d}}_{t-1,j}\right)\Delta t \quad , \quad \forall t \in \{1{:}T\},\ i \in \{1{:}M_t\},\ j = \alpha_{t-1}(i) \tag{6}$$

$$0 \le x_{t,i} \le E_{\max}, \quad \forall t \in \{1{:}T\},\ i \in \{1{:}M_t\} \tag{7}$$

$$0 \le u^{\mathrm{c}}_{t,i} \le u_{\max}, \quad \forall t \in \{0{:}T\},\ i \in \{1{:}M_t\} \tag{8}$$

$$0 \le u^{\mathrm{d}}_{t,i} \le u_{\max}, \quad \forall t \in \{0{:}T\},\ i \in \{1{:}M_t\} \tag{9}$$

In this formulation, $x_{t,i}$ denotes the state of charge, while $u^{\mathrm{c}}_{t,i}$ and $u^{\mathrm{d}}_{t,i}$ represent the charge and discharge power. The function $\alpha_{t-1} : \mathbb{N} \to \mathbb{N}$ maps each node index at stage t to its parent in the previous stage. The quantity $M_t = |\boldsymbol{\mathcal{C}}_t|$ denotes the number of nodes at stage t, and V_T represents a terminal cost that accounts for the value of the energy remaining in the battery at the end of the prediction horizon. In this paper, we define the terminal cost as $V_T(x) = \omega_T\, x$, where ω_T is a terminal price set to the average predicted imbalance price over the optimization horizon.

B. Results

Table I summarizes the optimization results for various scenario reduction techniques and distance metrics. This table reports the total profit earned by the BESS, the average scenario reduction error D_{K}, the average scenario tree construction time t_{red}, and the average optimization problem solve time t_{sol} over the simulation period. Table II further examines the impact of the branching rate m on the performance for the k-means reduction technique.

TABLE I
RESULTS WITH DIFFERENT REDUCTION TECHNIQUES ($m = 5$)

$c(\omega, \omega')$	Reduction	Profit [€]	D_{K}	t_{red} [s]	t_{sol} [s]
$\|\omega - \omega'\|_2^2$	FFS	55408	17862	**0.367**	0.570
	k-means	**55702**	**16142**	0.375	0.566
	k-medoids	55491	17030	1.020	0.583
	HAC	55698	16180	0.484	**0.564**
$\|\omega - \omega'\|_1$	FFS	55221	93.50	**0.357**	**0.553**
	k-medians	**55241**	**92.29**	0.397	0.565
	k-medoids	55231	92.62	0.897	0.571

TABLE II
INFLUENCE OF THE BRANCH GROWTH RATE m (K-MEANS REDUCTION)

m	Nodes	Profit [€]	D_{K}	t_{red} [s]	t_{sol} [s]
0	41	53793	22810	**0.006**	**0.030**
1	861	54779	18882	0.061	0.087
2	1681	55361	18061	0.143	0.181
5	4141	55702	16142	0.375	0.566
10	8241	55723	14617	1.005	1.014
15	12341	**55727**	**13872**	1.641	1.489

V. DISCUSSION

The results in Table I show that the clustering-based methods consistently outperform the multi-stage fast forward selection (FFS) method by Heitsch and Römisch in terms of profit and approximation error. The k-means method, in particular, achieves the highest profit and lowest approximation error. Notably, reduction methods based on the Manhattan distance resulted in significantly lower profits compared to those using the squared Euclidean distance. This may be attributed to the fact that the squared Euclidean distance is more sensitive to outliers, leading to clusters that are more dispersed toward the extremes—a characteristic that appears beneficial for the optimization problem at hand.

As shown in Table II, increasing the branch growth rate m improves both profit and approximation error, though it comes at the expense of increased computational effort. For sufficiently large values of m, the profit converges to a maximum value. As anticipated, the optimization problem solve time t_{sol} increases proportionally with the number of nodes in the tree. A key advantage of the proposed approach is its flexibility: once a scenario tree of future prices is generated, it can be reused across multiple optimization problems involving different BESS configurations. This reusability is especially valuable in the context of distributed energy resources.

VI. CONCLUSION

We proposed a multi-stage scenario-based stochastic model predictive control strategy for optimizing the operation of energy storage systems in real-time markets. Central to this approach is a stagewise refinement algorithm based on clustering techniques to construct non-symmetric multi-stage scenario trees. In a case study on the Belgian real-time market, our method, particularly when combined with k-means clustering, demonstrated superior profitability and scenario tree quality compared to traditional techniques.

REFERENCES

[1] K. Høyland and S. W. Wallace, "Generating scenario trees for multistage decision problems," *Management Science*, vol. 47, no. 2, pp. 295–307, 2001.

[2] R. Hochreiter and G. C. Pflug, "Financial scenario generation for stochastic multi-stage decision processes as facility location problems," *Annals of Operations Research*, vol. 152, no. 1, pp. 257–272, 2007.

[3] J. Dupačová, N. Gröwe-Kuska, and W. Römisch, "Scenario reduction in stochastic programming," *Mathematical Programming*, vol. 95, no. 3, pp. 493–511, 2003.

[4] H. Heitsch and W. Römisch, "Scenario reduction algorithms in stochastic programming," *Computational Optimization and Applications*, vol. 24, no. 2, pp. 187–206, 2003.

[5] K. Šutienė, D. Makackas, and H. Pranevičius, "Multistage k-means clustering for scenario tree construction," *Informatica*, vol. 21, no. 1, pp. 123–138, 2010.

[6] D. Xu, Z. Chen, and L. Yang, "Scenario tree generation approaches using k-means and lp moment matching methods," *Journal of Computational and Applied Mathematics*, vol. 236, no. 17, pp. 4561–4579, 2012.

[7] G. C. Pflug, "Scenario tree generation for multiperiod financial optimization by optimal discretization," *Mathematical Programming*, vol. 89, no. 2, pp. 251–271, 2001.

[8] A. Flores-Quiroz and K. Strunz, "A distributed computing framework for multi-stage stochastic planning of renewable power systems with energy storage as flexibility option," *Applied Energy*, vol. 291, p. 116736, 2021.

[9] H. Heitsch and W. Römisch, "Scenario tree modeling for multistage stochastic programs," *Mathematical Programming*, vol. 118, no. 2, pp. 371–406, 2009.

[10] J. Baetens, J. Laveyne, G. Van Eetvelde, and L. Vandevelde, "Imbalance pricing methodology in belgium: Implications for industrial consumers," in *2020 17th International Conference on the European Energy Market (EEM)*, 2020, pp. 1–6.

[11] R. Bruneel, M. Schuurmans, and P. Patrinos, "Optimal intraday power trading for single-price balancing markets: An adaptive risk-averse strategy using mixture models," *Applied Energy*, vol. 389, p. 125754, 2025.

[12] P. Pinson, H. Madsen, H. A. Nielsen, G. Papaefthymiou, and B. Klöckl, "From probabilistic forecasts to statistical scenarios of short-term wind power production," *Wind Energy*, vol. 12, no. 1, pp. 51–62, 2009.

[13] G. Monge, "Memoire sur la theorie des deblais et des remblais," *Mem. Math. Phys. Acad. Royale Sci.*, pp. 666–704, 1781.

[14] L. Mi, W. Zhang, X. Gu, and Y. Wang, "Variational wasserstein clustering," in *Computer Vision – ECCV 2018*. Springer International Publishing AG, 2018, pp. 336–352.

[15] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. University of California Press, 1967, pp. 281–297.

[16] M. Moshkovitz, S. Dasgupta, C. Rashtchian, and N. Frost, "Explainable k-means and k-medians clustering," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 7055–7065.

[17] L. Kaufman and P. J. Rousseeuw, "Partitioning around medoids (program pam)," in *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, Ltd, 1990, ch. 2, pp. 68–125.

[18] J. H. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.

[19] P. J. Goulart and Y. Chen, "Clarabel: An interior-point solver for conic programs with quadratic objectives," 2024.

[20] A. Novikov, "PyClustering: Data mining library," *Journal of Open Source Software*, vol. 4, no. 36, p. 1230, apr 2019.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.