Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

A SYNERGISTIC APPROACH TO WORKFORCE OPTIMIZATION IN AIRPORT SCREENING USING MACHINE LEARNING AND DISCRETE-EVENT SIMULATION

Lauren A. Cravy and Eduardo Pérez

Industrial Eng., Ingram School of Eng., Texas State University, San Marcos, Texas, USA

ABSTRACT

This study explores the integration of machine learning (ML) clustering techniques into a simulation-optimization framework aimed at enhancing the efficiency of airport security checkpoints. Simulation-optimization is particularly suited for addressing problems characterized by evolving data uncertainties, necessitating critical system decisions before the complete data stream is observed. This scenario is prevalent in airport security, where passenger arrival times are unpredictable, and resource allocation must be planned in advance. Despite its suitability, simulation-optimization is computationally intensive, limiting its practicality for real-time decision-making. This research hypothesizes that incorporating ML clustering techniques into the simulation-optimization framework can significantly reduce computational time. A comprehensive computational study is conducted to evaluate the performance of various ML clustering techniques, identifying the OPTICS method as the best found approach. By incorporating ML clustering methods, specifically the OPTICS technique, the framework significantly reduces computational time while maintaining high-quality solutions for resource allocation.

1 INTRODUCTION

Over the past two decades, airlines have faced numerous terrorist threats, prompting heightened attention to aviation security. The Transportation Security Administration (TSA) is tasked with safeguarding the transportation systems in the United States to ensure the free movement of people and commerce. TSA's current aviation security resources include approximately 14,000 units of transportation security equipment (TSE) deployed across 440 airports. To effectively fulfill its security responsibilities, TSA must efficiently allocate technology and human resources to adapt to the dynamic operational needs of various airport facilities. These challenges underscore the complexity of planning the optimal allocation of security screening resources at airport terminal checkpoints.

This research presents a comparative study aimed at improving the computational efficiency of an existing simulation-optimization framework designed to support the effective management of resources at airport security screening checkpoints (SSCPs) (Pérez et al. 2021). The study investigates the integration of ML clustering techniques into the simulation-optimization framework to enhance the efficiency of airport security operations. Simulation-optimization is particularly suitable for addressing problems involving dynamic data uncertainties, where critical system decisions must be made before the complete data stream is observed (Pérez et al. 2023; Kothamasu et al. 2022; Pérez et al. 2020). This is especially relevant in airport security, where passenger arrival times are unpredictable, and resource allocation must be planned in advance. However, the computational intensity of simulation-optimization limits its practicality for real-time decision-making. This research hypothesizes that incorporating ML clustering techniques into the framework can reduce computational time. A computational analysis is conducted to evaluate the performance of various ML clustering methods and identify the most effective approaches for this application. The remainder of the paper is structured as follows: Section 2 reviews related literature, Section 3 discusses the methodology, and Section 4 presents a computational study and Section 5 concludes the paper.

2 LITERATURE REVIEW

Airport security checkpoints are essential for ensuring passenger safety while maintaining operational efficiency. Given the high variability in passenger flow, optimizing resource allocation is crucial. This literature review examines research that integrates discrete event simulation (DES), optimization methods, and ML techniques—specifically clustering—to enhance real-time decision-making in airport SSCPs.

DES is extensively used to model airport security operations, simulating passenger movement and identifying bottlenecks. Research demonstrates its effectiveness in evaluating different configurations and resource allocations to improve efficiency (Ruiz and Cheu 2020). DES enables scenario testing, allowing security planners to anticipate peak times and adjust staff deployment accordingly (Pérez et al. 2021). Optimization models, including linear programming, integer programming, and meta-heuristic algorithms, have been employed to allocate Transportation Security Officers (TSOs) effectively. Studies show that strategic workforce scheduling can reduce wait times and improve throughput (Hanumantha et al. 2020). Multi-objective optimization methods balance security effectiveness with operational costs (Concho and Ramirez-Marquez 2012).

ML techniques, particularly clustering methods, have been explored to process large datasets efficiently. Clustering passenger profiles based on travel behavior aids in dynamically allocating resources. K-means and hierarchical clustering have shown promise in segmenting travelers for predictive staffing models (Tian et al. 2021). Combining clustering with DES can reduce computational complexity and enable real-time solutions. Although few studies have explicitly integrated these three methodologies, hybrid approaches incorporating DES, reinforcement learning, and optimization are emerging. Real-time analytics frameworks leveraging ML for data-driven decision-making in airport security are an area of growing interest (Martinez et al. 2023; Zertuche et al. 2024). While substantial work has been done individually on DES, optimization, and ML in airport security, limited research integrates these approaches comprehensively. Future research should explore hybrid models that leverage real-time passenger data for dynamic resource allocation.

3 METHODOLOGY

3.1 Discrete-event Simulation Model

This research utilizes the discrete-event simulation model developed by Pérez et al. (2021), which simulates the operations of two airport SSCPs at Phoenix Sky Harbor Airport (PHX). The simulation model was implemented using Simio (Pegden 2007). Each checkpoint includes two screening lanes (SL) with one travel document checker (TDC) designated for PreCheck passengers. At full capacity, both airport SSCPs can accommodate six screening lanes and six TDCs for general boarding passengers. On average, four TSOs are required per screening lane to manage travelers and operate the conveyors equipped with X-ray machines, Advanced Imaging Technology (AIT) body scanners, and Walk-Through Metal Detectors (WTMDs). An overall description of the discrete event simulation model is presented in Figure 1.

Figure 1 illustrates the overall layout of an airport SSCP and a snapshot of passenger arrival rates for a single day which are both inputs to the simulation model in Simio. The simulation is then used to obtain results for one performance metric: time in system. The simulation model alone can evaluate the effectiveness of personnel assignments to stations based on the level of service provided to passengers. TSOs are required to remain at their assigned stations for a minimum of 30 minutes. Determining the optimal number of TDCs and SLs to open daily within the constraints of maximum capacity (i.e., available resource hours) is a complex problem. Additionally, finding the ideal combination of assignments using only simulation is challenging due to the vast number of potential solutions. Therefore, supplementary tools are necessary to identify the best assignments based on anticipated demand.

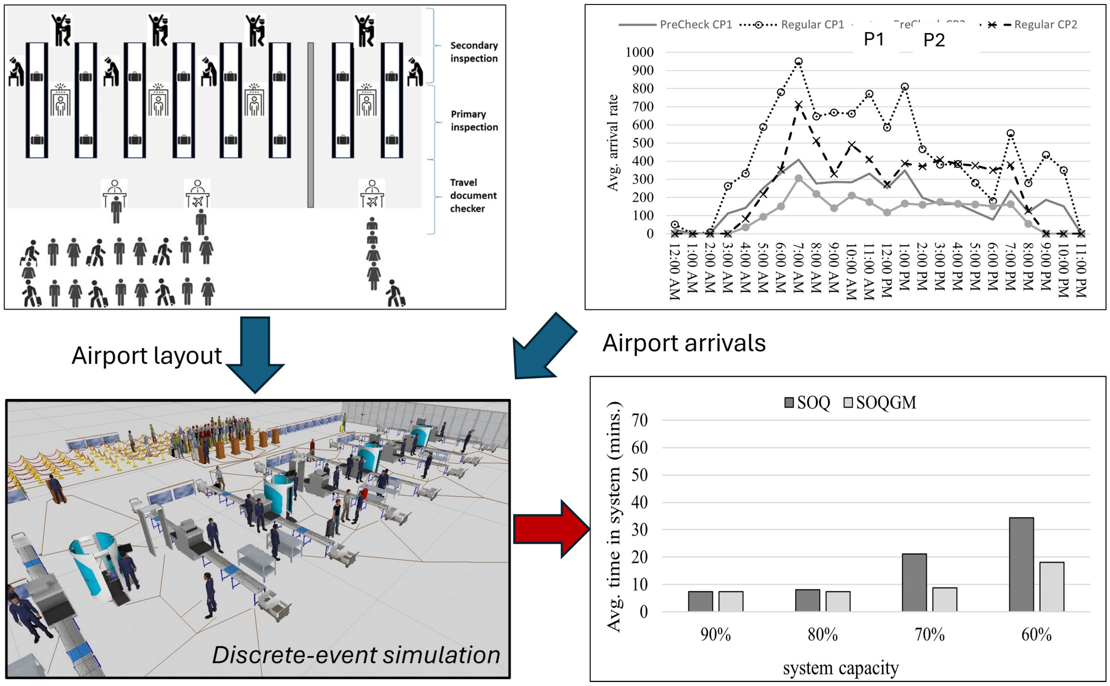

Figure 1: Airport security screening checkpoint simulation model in Simio.

3.2 System Optimization

A simulation-optimization framework was developed in (Martinez et al. 2023) to deal with the problem of identifying TSO assignments given a predetermined level of resource hours for a day. The framework utilizes the simulation model to compute performance measurements based on the assignment of resources. The performance measurements are then used by the optimization model to reconfigure the resource allocations to improve the performance metrics. The iterative process continues until no improvement is founds. In general, a simulation optimization problem is formulated as follows:

- ξ : the randomness in the system (e.g., passenger demand)
- x : the set of decision variables (e.g., TSO allocation per time-period)
- $f(x,\xi)$: the output for the objective for one replication of the simulation logic (e.g., passenger cycle time)
- Θ : the search space (e.g., all possible TDC and SL allocations)
- $min_{x\in\theta}(g(x))$: is the objective function where $g(x) = Ef(x,\xi)$

The objective function implemented minimizes the passengers time in system at the airport SSCPs. The optimization model is constrained to the number of TSOs available per 30-minute time-periods. The optimization models decides how many TDCs and SLs to open in the airport SSCPs based on the performance feedback provided by the discrete-event simulation model. The optimization models were implemented using the OptQuest Add-In in Simio (Sturrock and Pegden 2011). OptQuest find solutions by running multiple simulation trials, each representing a different configuration of the decision variables (i.e. number of TSOs asigned to SL and TDC stations per time period), capturing the variability and randomness inherent in real-world processes. After each trial, it evaluates the performance based on the defined objectives, using statistical analysis to assess how well each configuration meets the goal of minimizing passenger time in system (i.e., objective function). OptQuest employs intelligent search methods to explore the solution space, iteratively adjusting the decision variables guided by the results of previous trials to hone in on the most promising configurations. Through repeated trials and evaluations, OptQuest converges on a near-optimal solution, balancing trade-offs between different objectives and constraints to find the best possible configuration within the given parameters. Once the optimization process is completed, OptQuest provides the optimal solution along with detailed performance metrics, enabling users to make informed decisions about resource allocation and operational strategies. The resulting framework helped in determining the best staff configurations per time period based on capacity limitations (i.e., 70%, 80%, and 90%). However, these techniques are computationally demanding, requiring a significant amount of time to run numerous experiments.

3.3 Machine Learning

This paper identifies ML clustering methods that could be integrated to the existing simulation-optimization framework to reduce the computational time. Figure 2 illustrates the proposed integration of the ML clustering techniques within the simulation-optimization framework. The ML clustering techniques are used to analyze a predetermined number of trials (i.e., experiments) produced by OptQuest, the simulation-optimization add-in provided by the Simio software. The clustering analysis in then used to determine additional constraints that could be added to the optimization model with the goal of limiting the solution search space and reduce the computational time. The proposed method performs the following tasks:

- Initial Exploration: We first run the simulation-optimization for a limited number of trials (e.g., 600 trials in our study). Each trial represents a unique solution, which is a complete set of values for our decision variables (i.e., the number of TDCs and SLs assigned to each checkpoint for every 30-minute period).
- Identify High-Performing Solutions: We then use the ML clustering methods, each individually, to group these 600 trials based on the similarity of their decision variable values. A statistical analysis (as described in Section 3.3.2) allows us to identify the single "best-performing cluster"—the one containing solutions that most consistently yield low passenger wait times.
- Select Best ML Clustering Method: The ML clustering method that demonstrates greater consistency in resource assignments within its optimal cluster is selected.
- Derive New Constraints: The core of our method is to analyze the decision variable values exclusively within the best-performing cluster for the selected clustering method. For each decision variable (e.g., the number of TDCs at Checkpoint 1 during the 9:00 AM-9:30 AM time period), we determine its minimum and maximum observed values across all the trials within that elite cluster.
- Constrain the Search Space: These minimum and maximum values become the new, tighter bounds for their respective decision variables in the full optimization run.

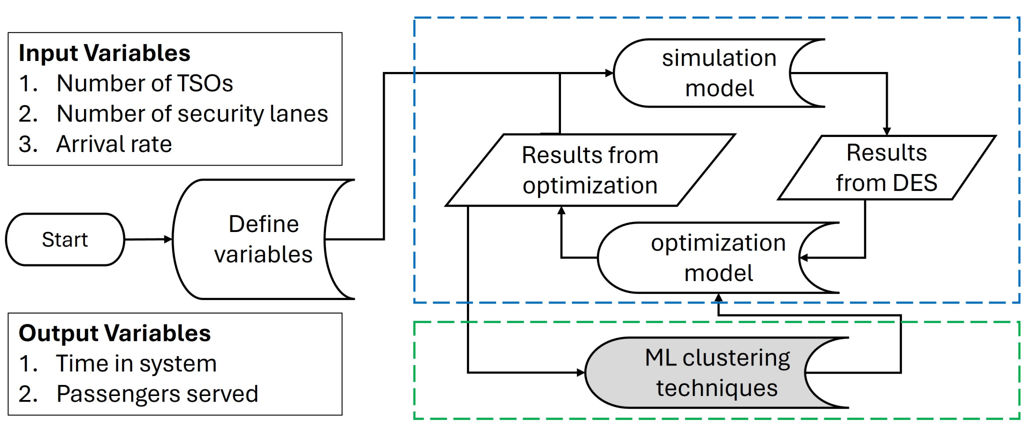

Figure 2: Simulation-optimization framework with ML integration.

Figure 3 illustrates the interaction between the ML clustering techniques and the optimization model implemented in OptQuest. The results of the optimization model are tabulated in three columns, where the first two columns present the values of the decision variables and the third column presents the value of the objective function. Each row represents a simulation trial. It will take OptQuest about 1000 trials to converge to near optimal solution in the proposed application which represents more than 24 hours of computational time. The proposed method integrates ML clustering methods to help the simulation-optimization framework converges to near-optimal solutions faster. Sections 3.3.1 – 3.3.4 describe in detail the process for identifying the best clustering method for the application under study (i.e., resource allocation in airport SSCPs). This process is individually applied to each candidate clustering method (i.e., OPTICS, k-means, and DBSCAN). Subsequently, as outlined in Section 3.3.5, the superior clustering method is chosen based on the strategy discussed in Section 3.3.4.

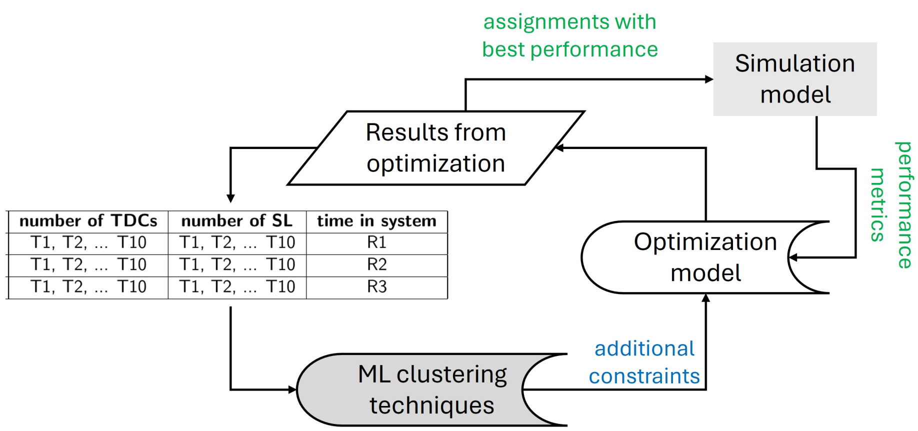

Figure 3: Machine learning clustering techniques interaction with optimization model implemented in OptQuest.

3.3.1 Determine the Optimal Number of Clusters

Determining the optimal number of clusters for clustering methods is essential for effective data segmentation. Several techniques can be employed based on the characteristics of your data and the clustering algorithm used (Mirkin 2011). The Elbow Method is particularly useful for partitioning clustering algorithms like K-means and K-medoids, involving plotting the within-cluster sum of squares against the number of clusters and identifying the "elbow" point where the rate of decrease sharply slows (Syakur et al. 2018). The Silhouette Method, suitable for various clustering algorithms including K-means, hierarchical clustering, and Gaussian Mixture Models (GMM), measures how similar each point is to its own cluster compared to other clusters, with the optimal number of clusters maximizing the average silhouette score (Januzaj et al. 2023). The Gap Statistic can be applied to partitioning clustering algorithms like K-means and hierarchical clustering, comparing the total within-cluster variation for different numbers of clusters with their expected values under a null reference distribution to identify the optimal number where the gap statistic is maximized. Cross-validation is useful for clustering algorithms that can be validated through splitting the data into training and validation sets, such as K-means and hierarchical clustering, assessing the stability and performance of clusters by evaluating how well the clustering generalizes to unseen data. Information criteria like the Bayesian Information Criterion (BIC) or Akaike Information Criterion (AIC) are suitable for model-based clustering algorithms such as GMM, evaluating model fit and complexity to determine the optimal number of clusters by balancing goodness of fit with model simplicity. Reachability plots, a key component of the OPTICS algorithm, help visualize the clustering structure and determine the optimal number of clusters by plotting the reachability distance of points, with valleys indicating dense clusters and peaks indicating sparse regions or noise (Mai et al. 2016). These methods provide a systematic approach to identifying the best number of clusters for your data, ensuring effective and meaningful segmentation.

3.3.2 Selection of the Cluster with the Best Performance

A statistical analysis was performed to determine the cluster with the best performance. The evaluation criteria included: (1) low average passenger time in the system per airport SSCP, (2) low standard deviation with minimal differences between checkpoint 1 (CP1) and checkpoint 2 (CP2), and (3) a high experimental count within each cluster, ideally above 10. When multiple clusters exhibit low average and standard deviation values, the experimental count becomes the primary factor. The cluster with the higher count should be chosen, even if the first two criteria are not as optimal, as a larger count indicates more reliable performance.

3.3.3 Capacity Assignments for the Cluster with the Best Performance

Once the best-performing cluster is identified, the next step is to analyze the frequency with which the same capacity level is selected for each station, specifically the TDCs and SLs, during each time period. Each airport SSCP can assign up to 6 TDCs and open up to 6 SLs per time period. By examining the frequency of these selections, valuable insights are gained into the consistency and reliability of the chosen capacity levels. This analysis aids in understanding operational patterns and ensures that resource allocation is optimized for efficiency.

3.3.4 Frequency Analysis for the Cluster with the Best Performance

In this step, the performance of each ML clustering method is benchmarked by evaluating how frequently the experiments within the selected cluster assign the same capacity level to each station and time period. The analysis involves computing the percentage of time the same capacity level is selected and counting the occurrences per time period. This benchmarking provides insights into the consistency and reliability of the capacity assignments across different clustering methods.

3.3.5 Selection of the Best Method based on Frequency Results

Upon completing the computational analysis, the performance of all ML clustering methods is compared using the results from the previous section. The ML clustering method that exhibits higher frequency results is deemed the best, as it demonstrates greater consistency in resource assignments within its optimal cluster.

3.3.6 Guidelines and Protocols

Once the best ML clustering method is selected, the resource assignment results from its optimal cluster are used to develop additional constraints for the optimization model. The optimal cluster helps identify the assignments per time period that have the highest potential to improve passengers' time in the system. Following this, a computational study is presented to demonstrate the advantages of the proposed method, particularly in finding near-optimal solutions with reduced computational time.

4 COMPUTATIONAL STUDY

In this study, data sets produced by OptQuest contain about 600 experiments. Each experiment contains the number of resources assigned to each airport SSCP station in 30-minute increments and performance metrics, such as passenger wait times and the number of passengers processed. There are two types of stations, (1) travel document checkers (TDC) and (2) security lanes (SL). A total of six data sets were considered in this study that vary in terms of resource availability and whether or not passenger types were considered.

The ML clustering methods are used to group scenarios producing similar results (i.e., resource assignments) and identify those clusters producing the best results. The information generated by the selected clusters is used to incorporate additional constraint to the optimization model, reducing the search space and solution times. Ten clustering methods were studied in this research including k-Means, k-Medoids, DBSCAN, OPTICS, DENCLUE, AGNES, DIANA, BIRCH, Chameleon, and Probabilistic. Given the space limitation, out of the ten methods, the "best-performing" clustering method was selected and discussed in this paper. Clustering method OPTICS, at 70% capacity, was chosen for its recognition as the most effective ML clustering approach. The chosen method, particularly at 70% capacity, was selected to show the most challenging scenario. The following sections will explore the research underlying the selection of OPTICS, detailing its advantages and how it effectively addresses the complexities of resource allocation in the airport SSCPs operation.

4.1 Clustering Results: Optimal Number of Clusters

The optimal number of clusters for OPTICS was determined using an expanded process from the DBSCAN method. Unlike the elbow method for k-Means and k-Medoids or the silhouette score for DBSCAN, OPTICS generates a reachability plot that visualizes data points within a densely populated clustering structure based on their reachability distance, identifying clusters of varying densities and shapes. Consequently, OPTICS does not require a predefined number of clusters but instead defines them based on the density connectivity of the data points. The reachability plot displays valleys and plateaus: valleys indicate areas of high density, representing clusters, while plateaus suggest sparse regions or noise. Clusters are identified by locating areas of low reachability distance, which signify tight density connections. As illustrated in Figure 4, the reachability chart contains numerous data points above the threshold height (i.e., 8), indicating many "noise" points. Nonetheless, the sum of these valleys indicated a total of 20 clusters.

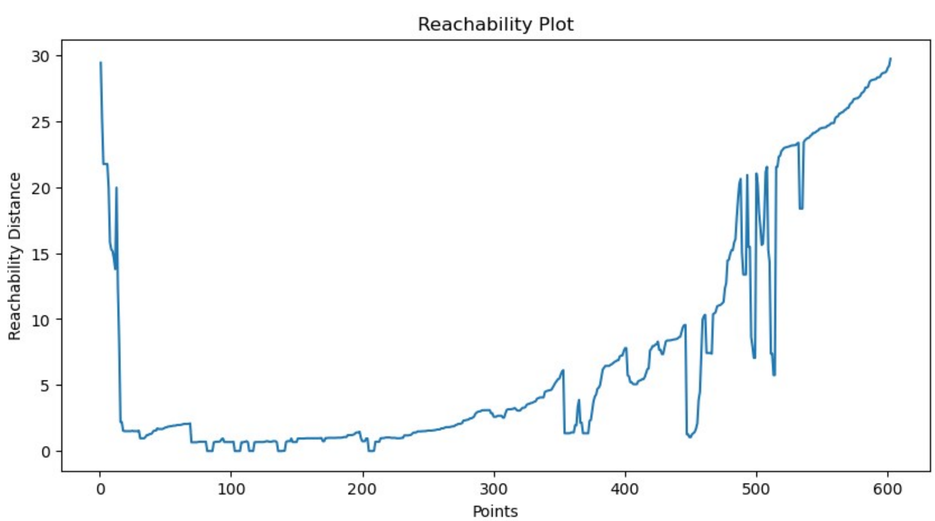

Figure 4: OPTICS reachability plot.

4.2 Clustering Results: Selection of the Cluster with the Best Performance

A statistical analysis was performed to selected the cluster with the best performance. Table 1 illustrates the summary of the analysis. The key elements to look for in the table are (1) low average passenger time in system per checkpoint (CP), (2) low standard deviation value with a minimum difference between checkpoint 1 (CP1) and checkpoint 2 (CP2), and (3) average to high experimental count within each cluster (preferably above a value of 10). However, if the data shows multiple clusters with a low average and standard deviation, the experiment count should be considered the main factor. The cluster with the higher count should be selected, even if the first two factors are not as low in comparison since a larger count represents a more accurate performance. Table 1 provides a range of clusters with better cluster performance – clusters 2 through 7. Clusters 2 and 5 have the highest number at 11. As a result, the determining factor was the difference in standard deviation, leading to cluster 2 as the selected choice as the best-performing cluster group.

Table 1: OPTICS statistics for passengers time in system.

Cluster	Avg. CP1	Avg. CP2	Std. Dev CP1	Std. Dev CP2	Count	Max CP1	Max CP2	Min CP1	Min CP2
-1	15.54	19.92	13.43	20.33	431	67.28	151.07	8.28	9.23
0	44.15	40.11	14.88	9.05	7	63.49	51.97	20.84	26.99
1	8.59	9.95	0.12	0.37	12	8.90	10.48	8.42	9.47
2	**8.59**	**9.45**	**0.01**	**0.04**	**11**	**8.64**	**9.46**	**8.59**	**9.33**
3	8.63	9.23	0.00	0.02	9	8.64	9.28	8.63	9.23
4	8.63	9.27	0.01	0.06	6	8.65	9.37	8.62	9.24
5	8.56	9.29	0.15	0.09	11	8.71	9.56	8.33	9.24
6	8.34	9.33	0.01	0.04	5	8.35	9.40	8.33	9.31
7	8.67	9.34	0.08	0.19	6	8.84	9.72	8.62	9.26
8	8.49	10.43	0.06	0.08	13	8.56	10.51	8.35	10.19
9	8.58	12.23	0.10	0.88	12	8.62	15.00	8.27	11.75
10	8.50	11.48	0.21	0.71	12	8.96	13.71	8.36	10.95

4.3 Clustering Results: Capacity Assignments for the Cluster with the Best Performance

After selecting the best cluster, we compute the frequency of selecting the same capacity level for each station (i.e., TDC and SL) per time period. Each station can assign up to 6 TDCs and open up to 6 SLs per time period per airport SSCP. Figures 5-8 illustrate the frequency per time period for TDC CP1, TDC CP2, SL CP1, and SL CP2 respectively. Each line color represent a different capacity level from 1 to 6. In examining the capacity levels in Figure 5, it is evident that 3 TDCs is the predominant choice across all 48 half-hour periods in CP1. It is observed in Figure 6 that the majority of the day depicts 2 TDCs in CP2. Figure 7 exhibits a mixture of capacity levels 3 to 5 for SLs in CP1 throughout the day. In the SP CP2 density plot displayed in Figure 8, all except two instances fall within capacity levels 2 to 4.

4.4 Clustering Results: Frequency Analysis for the Cluster with the Best Performance

In this step, we want to benchmark the performance of the selected cluster in terms of how often the experiments within the cluster assign the same capacity level per station and per time period. For example, the cluster selected for OPTICS has 11 experiments. For each time period (out of the 48 half hour periods) we compute the percentage of time the same capacity level was selected and count how many times it happens per time period. Table 2 reveals high-frequency counts (more than 40 time periods) above the 0.9 threshold for both airport SSCPs.

4.5 Clustering Results: Selection of the Best Method based on Frequency Results

After completing the frequency analysis for all ten clustering techniques, we performed a comparative evaluation to select the most suitable method for integration into the simulation-optimization framework. The primary selection criterion was the ability of a method to identify a 'best' cluster where the resource assignments showed high consistency. This consistency, measured by the frequency count detailed in Section 4.4, is critical because it indicates that the algorithm has found a stable and reliable high-performance region of the solution space.

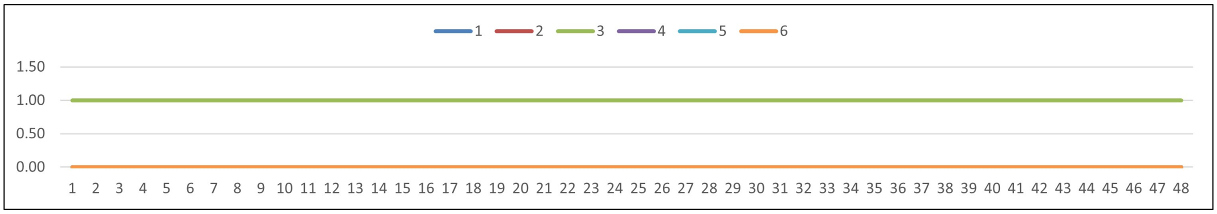

Figure 5: OPTICS TDC CP1 Capacity Levels - 70%.

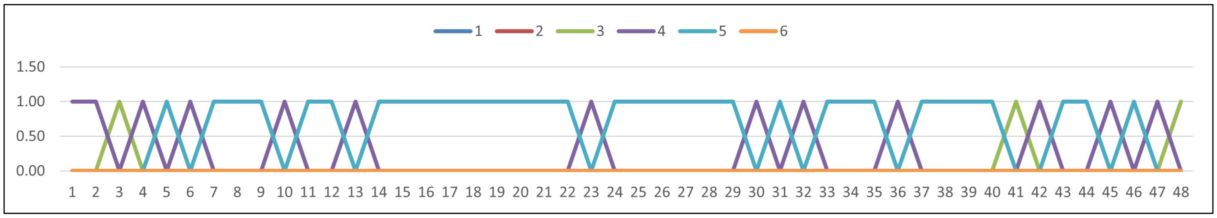

Figure 6: OPTICS TDC CP2 Capacity Levels - 70%.

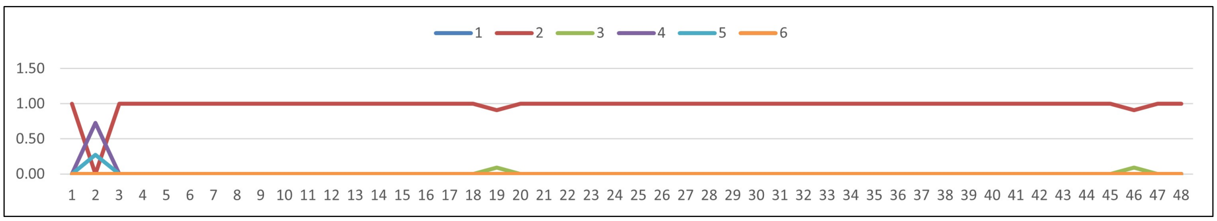

Figure 7: OPTICS SL CP1 Capacity Levels - 70%.

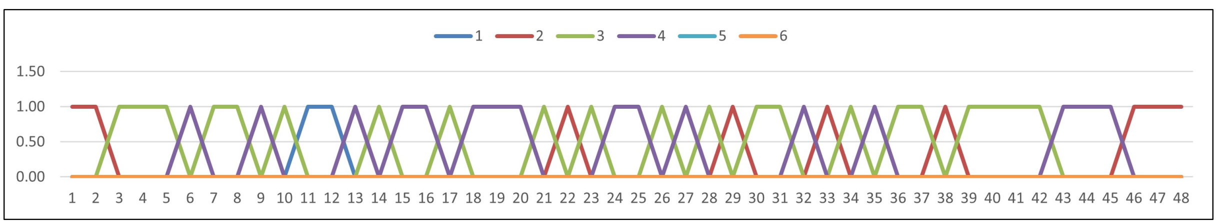

Figure 8: OPTICS SL CP2 Capacity Levels - 70%.

Table 2: OPTICS frequency at 70% capacity.

70% St							
TDC CP1		**TDC CP2**		**SL CP1**		**SL CP2**	
Range	**Frequency**	**Range**	**Frequency**	**Range**	**Frequency**	**Range**	**Frequency**
1-.90	48	1-.90	47	1-.90	48	1-.90	48
.89-.80	0	.89-.80	0	.89-.80	0	.89-.80	0
.79-.70	0	.79-.70	1	.79-.70	0	.79-.70	0
.69-.60	0	.69-.60	0	.69-.60	0	.69-.60	0
.59-.50	0	.59-.50	0	.59-.50	0	.59-.50	0
.49-0	0	.49-0	0	.49-0	0	.49-0	0

While several methods, including K-Means and DBSCAN, identified useful patterns, OPTICS was demonstrably superior in our application. It was the only method whose optimal cluster showed high-frequency (>90%) assignments for more than 40 of the 48 time periods across all station types. This superior consistency made it the most robust choice for generating reliable constraints. Due to conference page limits, we have presented the detailed results for OPTICS as it represents the best-in-class method from our comprehensive comparison.

4.6 Guidelines and Protocols

As stated earlier, OPTICS proved to be the most effective ML clustering technique for our application and was selected to be implemented within the framework presented in Figure 3. This section explains how the selected ML clustering technique is integrated to generate new, targeted constraints for the optimization model.

The cluster identified as the "best performer" (via the method in Section 3.3.2) contains a set of trials that consistently produced superior results. The key insight of our approach is that the decision variable values within this elite cluster represent high-potential resource allocations. To leverage this, we analyze the values for each decision variable (e.g., number of TDCs for a specific time period) across all trials within that best cluster. We identify the minimum and maximum value assigned to that variable within the cluster.

These minimum and maximum values become the new, tighter bounds for that specific decision variable in a subsequent, full optimization run. For instance, if the analysis of the best cluster revealed that the number of security lanes at Checkpoint 1 for the 8:00 AM time slot never dropped below 3 and never exceeded 5 among the best solutions, we would add the constraints $SL\_CP1\_T16 \geq 3$ and $SL\_CP1\_T16 \leq 5$ to the optimization model. This process is repeated for all decision variables, effectively using the clustering results to prune the search space and guide the optimizer toward near-optimal solutions more rapidly. The following illustrations show examples of the type of protocols that can be generated using this insight.

The following illustration show an example of the type of protocols that can be generated using the insight from the computational study. In Figure 9, CP1 is revealed to only range between capacity levels 3 and 5, representing the suggested number of staffing agents required throughout the day. TDC CP1 shows only three agents are necessary, while a majority of SL CP1 presents a total of five agents (with a mixture of three and four spread out). The next step of this research is to add those capacity constraints to the optimization model with the goal of limiting the search space so a solution can be found using less computational time.

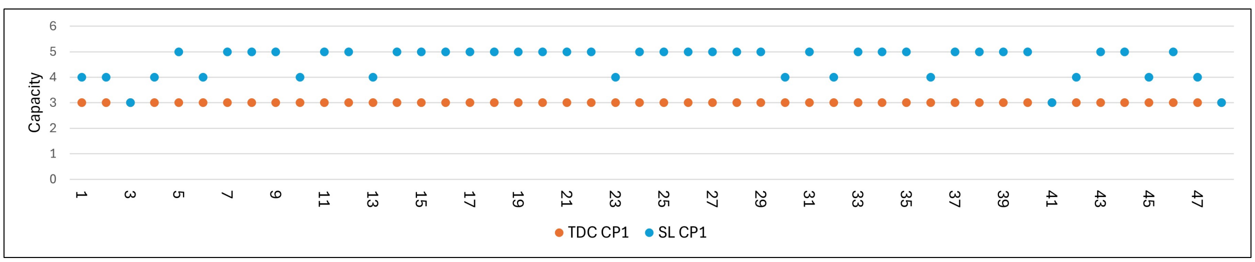

Figure 9: Recommended resource capacity assignments for CP1.

4.7 Computational Results

Table 3 shows the computational results for two implementation strategies for the simulation optimization framework. The experiments were run using a DELL OptiPlex 9010 Workstation with 3.4 GHz Intel Core i7 Processors and 16GB RAM. The first strategy was named simulation-optimization (SO) and it uses the optimization model described in Section 3.2 which objective function tries to minimize passengers time in the system. The second strategy was named simulation-optimization with ML (SOML) and it also applies the optimization model described in Section 3.2 plus the clustering technique developed in this paper. The computational results include the average time in system time, average throughput, and computational time.

Table 3: Computational results.

Implementation Strategy	Airport SSCP1		Airport SSCP2		Computational time (minutes)
	Average time in system	Average throughput	Average time in system	Average throughput	
SO	10.71	10,417.20	6.11	6,128.30	1,536
SOML	12.55	10,398.62	8.67	6,114.62	792

The SO strategy provided the best performance in terms of passenger average time in system and throughput. However, the difference in results relatively small when compared to the savings obtained in terms of computational time. The SOML showed about 64% reduction in computational time when finding a near optimal solution. The computational results show the potential of the proposed integration of ML clustering techniques within a simulation-optimization framework like the one provided by Simio.

5 CONCLUSIONS

This research proposes integrating machine learning into a simulation-optimization framework to expedite the discovery of high-quality solutions for allocating resources at airport SSCPs. The study identifies the OPTICS method as the most suitable ML clustering technique for this application. It outlines a structured approach for incorporating OPTICS into the simulation-optimization framework to enhance computational efficiency.

The integration of machine learning clustering techniques into the simulation-optimization framework for airport security screening has demonstrated significant potential in enhancing computational efficiency and operational effectiveness. By leveraging the OPTICS clustering method, this research has shown that it is possible to reduce the computational time required to find near-optimal solutions for resource allocation at airport security checkpoints. The clustering techniques help in identifying patterns and grouping similar resource assignments, which in turn allows for the incorporation of additional constraints into the optimization model, thereby narrowing the search space and expediting the solution process.

It is important to acknowledge the trade-off between solution quality and computational speed inherent in our findings. The computational results show that the standard SO approach produced mathematically superior solutions in terms of average wait time. However, the SOML framework produced solutions of high quality from a practical standpoint—with average wait times remaining within acceptable operational targets—while reducing the computational burden by nearly half. In a dynamic environment like an airport, the ability to generate a robust resource plan in under 14 hours versus over 25 hours provides a significant strategic advantage. This positions the SOML framework as a powerful tool for agile, operational decision-making, where the value of speed can often outweigh the benefit of a marginal improvement in the optimal solution.

Future research should focus on further refining the integration of machine learning techniques within simulation-optimization frameworks and exploring their application in other complex, real-time decision-making environments. Additionally, the development of more advanced clustering algorithms and their application to different aspects of airport operations could provide even greater improvements in efficiency and effectiveness. The findings of this study pave the way for more innovative and technologically advanced approaches to managing airport security, ultimately contributing to safer and more efficient air travel.

ACKNOWLEDGMENTS

This research was performed under an appointment to the U.S. Department of Homeland Security (DHS) Science & Technology (S&T) Directorate Office of University Programs Summer Research Team Program for Minority Serving Institutions, administered by the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy (DOE) and DHS.

ORISE is managed by ORAU under DOE contract number DESC0014664. All opinions expressed in this paper are the author's and do not necessarily reflect the policies and views of DHS, DOE or ORAU/ORISE.

REFERENCES

Concho, A. L., and J. E. Ramirez-Marquez. 2012. "A Mathematical Framework for Passenger Screening Optimization via a Multi-objective Evolutionary Approach". *Computers & Industrial Engineering* 62(4):839–850 https://doi.org/https://doi.org/10.1016/j.cie.2011.11.031.

Hanumantha, G. J., B. T. Arici, J. A. Sefair, and R. Askin. 2020. "Demand Prediction and Dynamic Workforce Allocation to Improve Airport Screening Operations". *IISE Transactions* 52(12):1324–1342 https://doi.org/https://doi.org/10.1080/24725854.2020.1749765.

Januzaj, Y., E. Beqiri, and A. Luma. 2023. "Determining the Optimal Number of Clusters using Silhouette Score as a Data Mining Technique". *International Journal of Online & Biomedical Engineering* 19(4):174–182 https://doi.org/https://doi.org/10.3991/ijoe.v19i04.37059.

Kothamasu, M., E. Pérez, and F. A. Mendez-Mediavilla. 2022. "Modeling and Simulation of Food Bank Disaster Relief Operations". In *2022 Winter Simulation Conference (WSC)*, 1614–1624 https://doi.org/10.1109/WSC57314.2022.10015480.

Mai, S. T., I. Assent, and A. Le. 2016. "Anytime OPTICS: An Efficient Approach for Hierarchical Density-Based Clustering". In *International Conference on Database Systems for Advanced Applications*. April 16-19, Dallas, USA, 164-179.

Martinez, D., D. Zertuche, and E. Pérez. 2023. "Simulation Based Strategy for Dynamic Workforce Allocation and Checkpoint Configuration in Airport Screening Facilities". In *Proceedings of the IISE Annual Conference & Expo*. May 21-23, New Orleans, USA.

Mirkin, B. 2011. "Choosing the Number of Clusters". *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1(3):252–260 https://doi.org/https://doi.org/10.1002/widm.15.

Pegden, C. D. 2007. "SIMIO: A New Simulation System Based on Intelligent Objects". In *2007 Winter Simulation Conference (WSC)*, 2293–2300 https://doi.org/10.1109/WSC.2007.4419867.

Pérez, E., V. Anandhan, and C. Novoa. 2020. "A Simulation-Based Planning Methodology for Decreasing Patient Waiting Times in Pure Walk-in Clinics". *International Journal of Information Systems in the Service Sector (IJISSS)* 12(3):34–54 https://doi.org/https://doi.org/10.4018/IJISSS.2020070103.

Pérez, E., Y. V. Marthak, and F. A. M. Mediavilla. 2023. "Analysis and Forecast of Donations at Domestic Hunger Relief Organizations Impacted by Natural Disasters". *Socio-Economic Planning Sciences* 86:101470 https://doi.org/https://doi.org/10.1016/j.seps.2022.101470.

Pérez, E., L. Taunton, and J. A. Sefair. 2021. "A Simulation-Optimization Approach to Improve the Allocation of Security Screening Resources in Airport Terminal Checkpoints". In *2021 Winter Simulation Conference (WSC)*, 1–11 https://doi.org/10.1109/WSC52266.2021.9715421.

Ruiz, E., and R. L. Cheu. 2020. "Simulation Model to Support Security Screening Checkpoint Operations in Airport Terminals". *Transportation Research Record* 2674(2):45–56 https://doi.org/https://doi.org/10.1177/0361198120903242.

Sturrock, D. T., and C. D. Pegden. 2011. "Recent Innovations in Simio". In *2011 Winter Simulation Conference (WSC)*, 52–62 https://doi.org/10.1109/WSC.2011.6147739.

Syakur, M. A., B. K. Khotimah, E. Rochman, and B. D. Satoto. 2018. "Integration k-Means Clustering Method and Elbow Method for Identification of the Best Customer Profile Cluster". In *IOP Conference Series: Materials Science and Engineering*. November 9-10, Surabaya, Indonesia.

Tian, H., M. Presa-Reyes, Y. Tao, T. Wang, S. Pouyanfar, A. Miguel, *et al*. 2021. "Data Analytics for Air Travel Data: a Survey and New Perspectives". *ACM Computing Surveys (CSUR)* 54(8):1–35.

Zertuche, D., D. Martinez, and E. Pérez. 2024. "A Simulation Based Metaheuristic for Workforce Allocation in Airport Screening Facilities". In *Proceedings of the IISE Annual Conference & Expo*. May 18-21, Montreal, Canada.

AUTHOR BIOGRAPHIES

LAUREN A. CRAVY is a research assistant in the Ingram School of Engineering at Texas State University. She is currently pursuing a degree in industrial engineering. Her research interests include simulation design and analysis and optimization via simulation. Her email address is vam123@txstate.edu.

EDUARDO PÉREZ is a Professor in the Ingram School of Engineering at Texas State University. He earned his Ph.D. in Industrial and Systems Engineering from Texas A&M University. His research interests include discrete event and agent-based simulation, systems engineering, and stochastic programming. His email address is eduardpr@txstate.edu and his website is https://https://imoss.engineering.txst.edu/.