Optimization of Quantum Time in Round Robin Scheduling using Clustering Algorithms
Progress Report

Sait Suer
Kennesaw State University
Marietta, Georgia, USA
ssuer@students.kennesaw.edu

ABSTRACT

This project proposes a novel approach for optimizing the quantum time in the Round Robin scheduling algorithm using unsupervised clustering techniques, specifically K-means and DBSCAN. Quantum time, a critical factor in the performance of Round Robin scheduling, is traditionally chosen arbitrarily, often leading to inefficiencies such as increased average waiting time and frequent context switching. Our approach leverages the ability of clustering algorithms to categorize processes based on their arrival time, burst time, and completion characteristics. By grouping similar processes into clusters, the scheduler can dynamically assign an optimal quantum time for each cluster, adapting to workload variations and improving overall performance. K-means is employed to group processes with similar burst times, while DBSCAN helps in identifying outliers and special cases where unique quantum times are required. This dynamic, data-driven quantum time adjustment is expected to outperform traditional fixed quantum time approaches, reducing average waiting time, minimizing context switches, and improving the overall throughput of the system. The effectiveness of this method will be validated through simulation, comparing results with traditional scheduling approaches.

CCS CONCEPTS

• **Computer systems organization** → **Real-time systems**; **Real-time operating systems**;

KEYWORDS

Performance comparison, OS scheduling policies, Performance index, Kmeans, Dbscan, Round Robin, RR

ACM Reference Format:
Sait Suer. 2024. Optimization of Quantum Time in Round Robin Scheduling using Clustering Algorithms Progress Report. In *2024 ACM Southeast Conference (ACMSE 2024), April 18–20, 2024, Marietta, GA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3564746.35YYYYY

1 RESEARCH STATEMENT AND CONJECTURE

The challenge of selecting an optimal quantum time in the Round Robin algorithm affects the overall system performance, particularly the waiting time and context switching. While previous approaches relied on classification and regression models, we hypothesize that clustering algorithms like K-means [4] and DBSCAN[3] can effectively group processes based on burst times and arrival patterns, allowing for more adaptive and context-aware quantum time selection. These clustering methods can categorize processes with similar characteristics, allowing the scheduler to assign optimal quantum times dynamically, leading to better system resource management and reduced job completion time.

2 RELATED WORKS

Round Robin (RR) scheduling is one of the most widely used scheduling algorithms due to its simplicity and fairness, though it suffers from inefficiencies stemming from the use of a fixed quantum time. Researchers have extensively studied methods to improve its performance by dynamically adjusting the quantum time.

Helmy et al.[2] introduced a weighted Round Robin scheduling technique aimed at balancing low overhead with a preference for shorter jobs. In their approach, higher-weighted processes receive proportionally more quantum time, allowing shorter tasks to complete faster and thus reduce the overall waiting time.

In [5], another algorithm continuously adjusts the quantum based on the average burst times of processes in the ready queue. This method uses registers to track the remaining and average burst times, ensuring that quantum time adapts to real-time system conditions rather than being fixed.

More recently, machine learning techniques have been applied to further optimize time quantum assignment [1]. Some studies have explored multi-categorical classification of schedules using supervised learning methods. In this approach, each schedule is assigned to a class representing a unique optimum time quantum value. K-Nearest Neighbors (KNN) and Random Forest Classifiers are commonly used. KNN classifies processes by identifying the 'k' nearest neighbors using distance metrics like Euclidean, Manhattan, or Minkowski distances. On the other hand, Random Forest creates an ensemble of decision trees, where each tree selects random subsets of features to split nodes and aggregates predictions across trees, leading to a robust and generalized model.

In contrast, our research takes a novel approach by utilizing clustering algorithms such as K-means and DBSCAN to group processes based on their burst times and arrival characteristics. While K-means is effective for grouping processes with similar attributes, DBSCAN excels in identifying noise and outliers—processes with unique or irregular behavior that may require special handling. By leveraging DBSCAN's ability to detect these outliers, we can assign unique quantum times to such processes, ensuring they do not

ACMSE 2024, April 18–20, 2024, Marietta, GA, USA

ACM ISBN 979-8-4007-0237-2/24/04…$15.00
https://doi.org/10.1145/3564746.35YYYYY

adversely affect overall scheduling efficiency. This dynamic, data-driven method to determine optimal quantum time for each cluster is expected to outperform existing machine learning techniques by minimizing average waiting times, reducing context switches, and improving overall system throughput.

Previous studies have focused on using machine learning models such as *Random Forests* and *Artificial Neural Networks* to optimize quantum time in Round Robin scheduling [1]. Clustering algorithms like *K-means* have been used in CPU scheduling to categorize jobs with similar features and optimize resource allocation [4]. *DBSCAN* (Density-Based Spatial Clustering of Applications with Noise) has been applied in anomaly detection and grouping jobs with outliers, which could benefit scheduling decisions by separating high-burst or high-priority jobs from typical workloads [3].

3 METHODOLOGY

3.1 Dataset and Simulation Setup

For the purposes of this study, we used synthetic data to simulate the scheduling environment. Processes were manually generated with specific values for *process ID*, *arrival time*, and *burst time*. These values were input to create a diverse set of processes for clustering and scheduling. The arrival times and burst times of processes were varied to simulate real-world variability in workload.

3.2 Clustering Approach

Two clustering algorithms, *K-means* and *DBSCAN*, were used to group processes based on their *arrival times* and *burst times*. Clustering was performed using *Euclidean distance* as the metric to calculate the similarity between processes.

- **K-means**: In K-means clustering, the number of clusters was fixed at *2*, and the processes were divided into two groups. For each cluster, the *quantum time* was dynamically assigned as the *maximum burst time* within that cluster. This ensures that processes with longer burst times are grouped together, and the quantum time is reflective of the process characteristics in that cluster.
- **DBSCAN**: In the case of DBSCAN, a hyperparameter search was conducted to find the best values for *minPts* and *eps*. After testing, the values *minPts = 3* and *eps = 10* were selected as the most effective parameters for our dataset. Unlike K-means, DBSCAN has the ability to identify noise and outliers—processes that do not fit into any cluster. For processes identified as noise, a *fixed quantum time of 3 time units* was assigned, reflecting their unique or irregular nature. For the other clusters, the quantum time was set as the *maximum burst time* of the processes within each cluster.

3.3 Quantum Time Assignment

For both clustering approaches, the quantum time was not fixed but dynamically assigned based on the characteristics of the clustered processes:

- In *K-means*, the quantum time for each cluster was the *maximum burst time* of the processes within the cluster.
- In *DBSCAN*, the quantum time for regular clusters followed the same rule as K-means, while processes classified as noise received a *fixed quantum time of 3 time units*.

This dynamic quantum assignment allowed for more efficient handling of processes with varying burst times, as processes within each cluster were handled based on their specific workload characteristics.

3.4 Simulation Process

The simulation compared the performance of the traditional Round Robin (RR) scheduling algorithm with the adaptive quantum approaches derived from *K-means* and *DBSCAN* clustering. In the traditional RR algorithm, a fixed quantum time was used, while in the adaptive versions, the quantum time was determined based on the clustering results. The simulations were performed for all three approaches to evaluate the effectiveness of the proposed methodology.

3.5 Evaluation Metrics

To measure the performance of the scheduling algorithms, the following metrics were used:

- **Average Waiting Time**: The average time that processes spent waiting in the ready queue before execution.
- **Average Turnaround Time**: The average time between process arrival and completion.

The results of the adaptive quantum approaches (K-means and DBSCAN) were compared against the traditional Round Robin method. The goal was to demonstrate that the dynamic, cluster-based quantum time assignment could reduce both *average waiting time* and *average turnaround time*, leading to more efficient process scheduling.

4 PROGRESS

So far, the project has progressed according to the initial proposal. The key accomplishments include:

- **Literature Review**: A thorough review of existing work on Round Robin scheduling and relevant clustering algorithms such as K-means and DBSCAN has been completed.
- **Implementation**: The Python version of the Round Robin scheduling algorithm has been developed, and both the K-means and DBSCAN algorithms have been adapted to fit within the Round Robin framework.
  - **K-means Clustering**: For the K-means algorithm, clusters were formed based on process burst times and arrival times. Quantum times for each cluster were computed separately using minimum, average, and maximum values within each cluster. After comparing results, the best performance was achieved when the maximum value was used to assign quantum time to clusters.
  - **DBSCAN Clustering**: A hyperparameter search for DBSCAN was performed to find the optimal values for `minPts` and `eps`. The algorithm was successfully implemented and integrated with Round Robin scheduling, with special handling for noise (outliers) by assigning them a fixed quantum time of 3 units.

- **Simulations**: Simulations were conducted using the traditional Round Robin, K-means, and DBSCAN-based versions of the scheduler on datasets containing noise. The results were compared to evaluate the impact of clustering on scheduling performance.

4.1 Next Steps

The next stage of the project will involve:

- Running further experiments on noisy datasets, continuing to collect results and analyze performance.
- Applying the algorithms to datasets with different characteristics to observe how they perform across varying scenarios.
- Analyzing and comparing the results to assess the robustness of the K-means and DBSCAN approaches, particularly in environments with noisy data.

5 PRELIMINARY RESULTS AND ANALYSES

In this section, we present the preliminary results obtained from the simulations conducted using the traditional Round Robin scheduling algorithm, as well as the K-means and DBSCAN clustering algorithms. The original dataset utilized for this study is as follows:

```
original\_process\_list = [
    Process(pid='P1', arrival\_time=1, burst\_time=10),
    Process(pid='P2', arrival\_time=2, burst\_time=15),
    Process(pid='P3', arrival\_time=3, burst\_time=17),
     Process(pid='P4', arrival\_time=4, burst\_time=60)
]
```

In this dataset, process P4 is clearly identified as an outlier due to its significantly larger burst time.

5.1 Traditional Round Robin Results

Using a quantum time of 3, the results obtained from the traditional Round Robin scheduling algorithm are summarized in Table 1:

Table 1: Traditional Round Robin Results

PID	Wait Time	Turnaround Time
P1	27	37
P2	33	48
P3	38	55
P4	39	99

The average wait time and turnaround time for the traditional Round Robin algorithm are as follows:

- Average Wait Time: 34.25
- Average Turnaround Time: 59.75

5.2 K-means Round Robin Results

For the K-means approach, the quantum times for clusters were determined using the MAX method, resulting in the values of 17 and 60. The Round Robin results for this method are summarized in Table 2:

The average wait time and turnaround time for the K-means Round Robin algorithm are as follows:

- Average Wait Time: 52.75

Table 2: K-means Round Robin Results

PID	Cluster	Quantum Time	Wait Time	Turnaround Time
P4	1	60.00	0	60
P1	0	17.00	59	69
P3	0	17.00	68	85
P2	0	17.00	84	99

- Average Turnaround Time: 78.25

5.3 DBSCAN Round Robin Results

In the case of the DBSCAN algorithm, process P4 was marked as noise. The quantum times for clusters were found to be 17 and 0. The results for the Round Robin scheduling using DBSCAN are shown in Table 3:

Table 3: DBSCAN Round Robin Results

PID	Cluster	Quantum Time	Wait Time	Turnaround Time
P1	0	17.00	-1	9
P2	0	17.00	8	23
P3	0	17.00	22	39
P4	-1	3.00	38	98

The average wait time and turnaround time for the DBSCAN Round Robin algorithm are as follows:

- Average Wait Time: 16.75
- Average Turnaround Time: 42.25

5.4 Analysis

The results indicate that the traditional Round Robin algorithm had an average wait time of 34.25 and an average turnaround time of 59.75, while the K-means approach produced higher wait and turnaround times. In contrast, the DBSCAN approach demonstrated superior performance with significantly lower average wait and turnaround times.

This suggests that the DBSCAN algorithm's ability to identify noise processes (such as P4) may contribute to its enhanced performance by minimizing the negative impact of outliers on scheduling efficiency. Further analysis will focus on evaluating the effectiveness of these algorithms across different datasets and configurations.

6 CONCLUSION AND FUTURE WORK

In this research, we explored the use of clustering algorithms, specifically K-means and DBSCAN, to enhance the traditional Round Robin scheduling algorithm. Our findings indicate that while the traditional method serves as a useful baseline, the integration of clustering techniques significantly improves scheduling performance by minimizing average waiting times and average turnaround time.

The K-means approach, which utilized the maximum burst time for quantum time assignment, provided valuable insights into process grouping. Meanwhile, DBSCAN effectively identified noise processes, such as outlier P4, facilitating more accurate quantum time determination and leading to a more efficient scheduling framework.

In conclusion, this research lays the groundwork for further exploration at the intersection of clustering techniques and scheduling algorithms, paving the way for advancements in process optimization.

Future work will involve analyzing a broader range of datasets, focusing on those with and without noise characteristics to assess the robustness of our algorithms in managing outliers and their performance in ideal scenarios. Additionally, we aim to incorporate real-world data to enhance the applicability of our findings.

Overall, this future research will deepen our understanding of how clustering techniques can enhance process scheduling and lead to the development of more adaptable and effective scheduling algorithms.

REFERENCES

[1] Paarshva Chitaliya et al. "Time Quantum Optimization in Round Robin Algorithm". In: *2023 International Conference on Network, Multimedia and Information Technology (NMITCON)*. 2023, pp. 1–6. doi: 10.1109/NMITCON58196.2023.10276146.

[2] Tarek Helmy and Abdelkader Dekdouk. "Burst Round Robin as a Proportional-Share Scheduling Algorithm". In: *Proceedings of The Fourth IEEE-GCC Conference on Towards Techno-Industrial Innovations*. Bahrain, 2007, pp. 424–428.

[3] Kamran Khan et al. "DBSCAN: Past, present and future". In: *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*. IEEE. 2014, pp. 232–238.

[4] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. "The global k-means clustering algorithm". In: *Pattern recognition* 36.2 (2003), pp. 451–461.

[5] Abbas Noon, Ali Kalakech, and Seifedine Nimer Kadry. "A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average". In: *CoRR* abs/1111.5348 (2011). arXiv: 1111.5348. url: http://arxiv.org/abs/1111.5348.