Baghdad Science Journal

Volume 23 | Issue 6 Article 22

6-26-2026

Data Scheduling Using Distance Function with Single-Pass Processing

Mahmood Shakir Hammoodi
Computer Center, University of Babylon, Babylon, Iraq, pre.mahmood.shakir@uobabylon.edu.iq

Suhad Hatim Jihad
Computer Center, University of Babylon, Babylon, Iraq, suhad.jihad@uobabylon.edu.iq

Mithal Hadi Jebur
Computer Center, University of Babylon, Babylon, Iraq, mithalhadi@uobabylon.edu.iq

How to Cite this Article

Hammoodi, Mahmood Shakir; Jihad, Suhad Hatim; and Jebur, Mithal Hadi (2026) "Data Scheduling Using Distance Function with Single-Pass Processing," *Baghdad Science Journal*: Vol. 23: Iss. 6, Article 22.
DOI: https://doi.org/10.21123/2411-7986.5336

BAGHDAD SCIENCE JOURNAL 2026;23(6):2233–2243

RESEARCH ARTICLE

Data Scheduling Using Distance Function with Single-Pass Processing

Mahmood Shakir Hammoodi *, **Suhad Hatim Jihad**, **Mithal Hadi Jebur**

Computer Center, University of Babylon, Babylon, Iraq

ABSTRACT

Task scheduling is one of the most fundamental difficulties confronting numerous sectors, including education, healthcare, and industrial management. Scheduling quality has a direct impact on performance efficiency, resource utilization, and end user satisfaction. With the growing volume and diversity of data, there is a greater demand for intelligent systems that can handle the spatial and temporal complexity of jobs and resource allocation. As a result, modern data analysis and decision-making processes have become critical for ensuring workflow optimization while reducing time and space waste. This study proposes an innovative approach to task scheduling that combines multidimensional data analysis using Data Cubes (a business intelligence method that enables multidimensional data analysis) and spatial optimization using the Euclidean distance function. The main idea is to represent scheduling elements such as tasks, resources, times, and locations in a structured data cube, allowing for rapid execution of multidimensional queries and aggregations. By evaluating the Euclidean distance between jobs and available resources, the system may find the best match based on the shortest distance, ensuring efficient resource allocation. The use of a data cube to study scheduling patterns, resource utilization, and time allocation improves decision-making. The results indicate the model's effectiveness and scalability. Furthermore, merging business intelligence tools with engineering optimization methods creates a potential foundation for addressing complicated scheduling issues in dynamic situations.

Keywords: Business intelligence, Data cube, Dynamic environments, Euclidean distance, Multidimensional data analysis, Performance optimization, Task scheduling

Introduction

Data scheduling plays an important role in most manufacturing and service industries, in information processing and communication. It usually applies heuristic methods and/or mathematical techniques to identify specific resources to process the tasks in order to enhance the company's objectives and goals. A priority level with the nearest possible initialization time is assigned to each task. Hence, the objectives and goals are to minimize the time to accomplish the tasks as well as gain the best performance of the schedule. There are many types of data scheduling techniques, including heuristic rules, mathematical techniques, nearest neighbor, and artificial intelligence.[1] Fig. 1 shows an example of a data scheduling technique that consists of two main tasks: searching for available time for scheduling and task scheduling. Searching over the scheduled task with its time is applied to avoid conflict or collision during the process of task scheduling.[2]

Dynamic scheduling is a type of scheduling in which the tasks' preferences may change or update over time. Adapting dynamically to these changes is one of the main objectives of dynamic scheduling, unlike traditional scheduling methods that often rely on fixed plans.[3]

Dynamic scheduling adjusts its resource allocation depending on real-time system states, which mainly include task preferences, deadlines, and resource

Received 1 June 2025; revised 26 November 2025; accepted 9 December 2025.
Available online 26 June 2026

* Corresponding author.
E-mail addresses: pre.mahmood.shakir@uobabylon.edu.iq (M. S. Hammoodi), suhad.jihad@uobabylon.edu.iq (S. H. Jihad), mithalhadi@uobabylon.edu.iq (M. H. Jebur).

https://doi.org/10.21123/2411-7986.5336

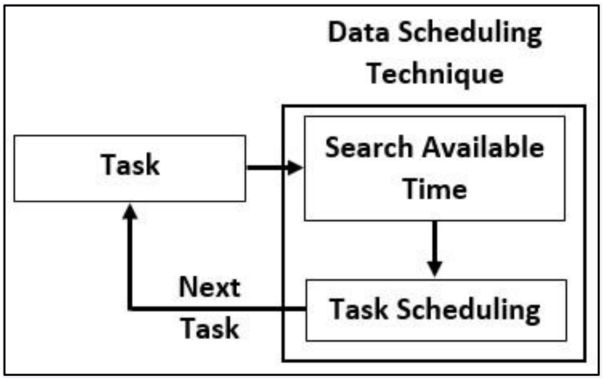

Fig. 1. Data scheduling technique.

constraints.[4] In addition, the optimization objectives aim to minimize characteristics such as execution time, energy consumption, and resource wastage, while ensuring high system throughput and reliability.[5] There are two types of techniques used for the purpose of dynamic scheduling Heuristic and Machine Learning-based.[6] In Heuristic, Rule-based methods are used to schedule tasks based on closeness, availability, and resource requirements. Whereas in Machine Learning-based approaches, real-time prediction of optimum task allocation is made using reinforcement learning or deep learning. Nowadays, dynamic data scheduling is used to handle diverse systems with a variety of challenges, such as cloud computing, quality of service, IoT environments, and distributed green data centers.[7,8]

Dynamic scheduling using data cubes integrates user-defined preferences and tasks into a multi-dimensional structure. By applying a distance function, tasks are allocated to the nearest cell based on predefined criteria such as resource availability, time slots, and task dependencies. This approach simplifies real-time scheduling and improves system responsiveness.[9]

Fig. 2 shows an example of a data cube with a multi-dimensional representation model for organizing and analyzing data in a manner that facilitates analysis and retrieval. Instead of putting data in normal relational tables, this allows multi-dimensional forms of data to exist in the database, enabling more complex relationships that exist between various measurements like time, product, or geographic location, among others, in a set of data. This facility plays a supporting role in doing complex analysis and getting correct results out of the source of data.[10]

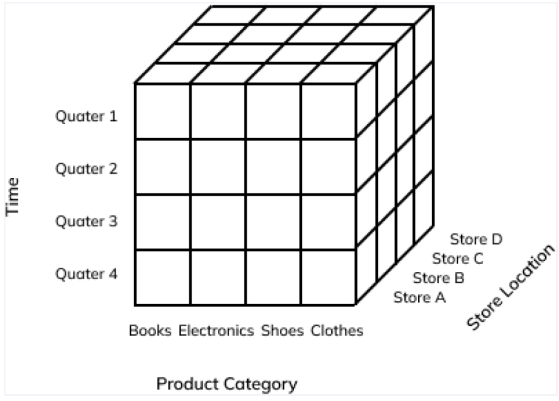

Fig. 2. Example of data cube.[9]

In a data cube, large data can be stored, manipulated, and viewed at different levels of detail. Data cube analysis is categorized as relational and multi-dimensional.[11] In relational, it can handle large data, but it might be slower when the volume of data increases, compared to the other alternatives. At the same time, multidimensional data analysis is much faster.[12] However, relational and multidimensional can be combined together. Some data are stored in relational form, while others are stored in multidimensional form, thereby improving performance and scalability.[13] Basic operations of a data cube are Roll-Up, Drill-Down, Slice, Dice, and Pivot. Roll-Up consolidates data from different levels to create an aggregated summary. Drill-Down examines summary data in a more detailed way. Slice selects one or a few of the dimensions. Dice selects a certain subset of data to analyze at the intersection of certain dimensions. Whereas Pivot cleans movement changes the universal scale-cube's orientation, allowing for new inspection angles and evaluating data from a different perspective.

Related works

This section discusses the existing approaches that handle the task scheduling problem in different domains.

Mohammadzadeh *et al.*[14] presented a very complicated workflow management problem in multi-site cloud computing environments concerning the need to manage workflows in places where data and resources are distributed among several data centers. This research study aims to improve the scheduling process using algorithmic techniques that take into account different factors such as resources, time, and reliability.

Tu *et al.*[15] proposed an approach to find optimum solutions to complex engineering problems by means of stochastic optimization techniques. An algorithm called the Colonial Selection Algorithm (CPA) was proposed, which simulates selection strategies in nature among animals like prey broadcasting, bailing, and favoring the most probable hunter to succeed and looking for new prey. It relies on an original mathematical model that uses the rate of success

to update strategy and imitates animal behavior in selective elimination. The algorithm also provides a new method of handling cross-boundary cases by replacing outliers with optimal values to enhance their ability to exploit. CPA was applied to five classical engineering problems to test its efficiency.

Raju *et al.*[16] proposed a scheduling strategy to optimize fog resource allocation in IoT environments while balancing minimizing service time and energy consumption and meeting time-critical task requirements. This scheduling strategy aims to allocate fog computing resources to meet IoT requests while adhering to deadlines and resource availability constraints of these requests. The scheduling problem is formulated as a hybrid nonlinear program (MINLP), which aims to minimize energy consumption and service time while considering time and resource constraints. To handle high-dimensional tasks in dynamic environments, a fuzzy logic-based reinforcement learning (FRL) mechanism is applied to minimize service time and energy consumption. Tasks are first prioritized using fuzzy logic, and then the prioritized tasks are scheduled using policy-based reinforcement learning (OPRL), which enhances long-term rewards compared to traditional methods.

Ghafari *et al.*[17] proposed an approach to schedule jobs effectively in a heterogeneous fog-cloud environment, taking cognizance of the task resource requirements and task deadline constraints, in addition to the capacity and availability of resources at fog nodes. A recommended fuzzy logic-based decision-making technique for a real-time task scheduling algorithm is applied in order to optimally allocate tasks in FIFO order between fog and cloud layers. The research methodology involves analyzing task requirements and spreading them over virtual machines (VMs) in the fog layer, focusing on tasking the minimal-loaded VMs with tasks to cut down on waiting time. Test-beds were prepared for simulation experiments, which then followed the evaluation of the proposed algorithm against several other related algorithms with metrics that included makespan, delay rate, as well as average execution time.

Mustapha *et al.*[18] proposed an approach to enhance resource utilization and increase the efficiency of cloud computing services through the provision of effective task scheduling and allocation, which most contribute toward improving resource utilization and operational efficiency. DBSCAN's clustering was applied for analysis and distribution of tasks in varying cloud computing environments.

SAIF *et al.*[19] proposed an approach to schedule tasks in the Cloud-Fog environment from a Fog Broker after analyzing and estimating requests from edge devices and then executing them through the MGWO (Multi-Objective Grey Wolf Optimization) algorithm. This approach aims to minimize the delay and energy consumption in meeting Quality of Service (QoS) objectives.

Karimunnisa *et al.*[20] proposed an approach that aims to reduce the make span for the cloud environment by increasing resource utilization efficiency and decreasing overall delay in job execution. In the first stage, tasks are preprocessed by applying the improved density-dependent clustering (IDCM) approach to classify them based on resource requirements and processing time. Then, in the second stage, an algorithm, the Cauchy mutation-based Enhanced Clustering Scheduling (ECOA-TS) algorithm, is applied to avoid trapping into local optima and find the best allocation of tasks to virtual machines.

Li *et al.*[21] developed an approach for handling various uncertainties in renewable energy, including human comfort and variability in building thermal inertia, which aims to improve the economic performance and balance makespan and efficiency of the system.

Huang *et al.*[22] proposed an approach for scheduling in a system with Internet of Things integrated cloud and fog computing environments using deep learning. This approach worked on how quality of service could be maintained while minimizing energy cost and execution time, as less resource consumption, better service, and low energy cost are possible if resource utilization is done optimally. DLA-BDTSS algorithm was applied, which operates under multi-objective strategies for obtaining an optimization between resource efficiency and delay minimization. It created division of labor based on priority and resources among IoT nodes, fog nodes, and the cloud. Deep learning was done through the Deep Q-Learning algorithm.

Pal *et al.*[23] proposed an approach that balances maximizing service providers' profits, minimizing the probability of task loss, and maintaining acceptable QoS levels in energy-intensive distributed green data centers. A multi-objective model was proposed that exploits the efficiency of the Simulated Annealing-based Bi-objective Differential Evolution (SBDE) algorithm for provisioning and handling the operation within Pareto-optimal solutions in connection with determining the task service rate and its distribution among Internet service providers, as well as the selection of the optimal solution to allocate tasks to ISPs using the Manhattan minimum distance measurement technique.

Song *et al.*[24] proposed an approach that handles the sharing and common use of Big Data hosted by or related to Remote Sensing. The "search and retrieve" mechanism of the proposed methodology can sort out such Big Data efficiently and with better quality in a multi-cloud environment through implementing an in-memory distributed data strategy based on Alluxio,

acceleration of spatial-temporal data discovery using the LSA algorithm, and data selection based on QF, i.e., quality-based filtering tactics applied for selecting the most relevant data.

Mahdi *et al.*[25] proposed an approach that inspires tourists by reducing travel times between attraction sites before adverse effects of the admissions. Combine access to tourism activity with an average share of 10 particular realistic criteria regarding travel, privacy, and security, and calculate the levels of safety and security at the activity locations by using geographic analysis guided by genetic algorithms. The reduction in travel times and the improvement in the tourists' experience have not been generalized into a common numerical measure of the difference in the algorithm's performance across scenarios. The study suggests incorporating traffic congestion patterns and new intelligent means like electric cars and enhancing the algorithms so that they better serve a group of tourists.

Abduljabbar *et al.*[26] proposed an approach for scheduling tasks using a Genetic Algorithm to generate a timetable based on courses, teachers, and time. However, a large amount of data is required for the purpose of training to avoid the issue of redundancy in scheduling.

Although different approaches have been proposed to handle the issue of scheduling tasks in different domains, none of them take into consideration handling the issue of scheduling tasks in combination with the preferences of scheduling tasks with a single pass over the data. This may enhance the process of scheduling, especially when the data is considered massive or big data, which requires dynamic scheduling.[27] This paper proposes a unique hybrid integration that has not been previously addressed. The aspects of the paper that can be considered novel are:

1. The possibility to embed scheduling attributes (tasks, resources, time, location) into a multidimensional OLAP-based data cube that enables analytical aggregation before scheduling on raw data.
2. Using Euclidean distance on cube-aggregated slices, which reduces noise and gives a more stable similarity measurement compared to traditional heuristic scheduling methods.
3. A single-pass scheduling pipeline that combines cube analysis with spatial optimization in a lightweight manner, both from the perspective of computations as well as memory usage.

The proposed technique

Data together with dimensions can be easily interpreted using a data cube. An n-dimensional range (x, y, z... n) of values is used to visualize the time sequence of data. Each dimension represents a certain feature of a given example, and the measure of interest is represented in a Data Cube cell, such as the nearest distance between a Data Cube and a task. In this research study, for better understanding and visualization, 3-D (x, y, z) is applied. x refers to Time. y refers to Day. At the same time, z refers to a task (Item, Subject, Place, Classroom, etc.). Data can be scheduled in a Data Cube cell $\{x, y, z\}$. Where $x = \{1, 2, 3... n\}$, $y = \{1, 2, 3, 4, 5, 6, 7\}$, and $z = \{$task1, task2, task3... task $m\}$. Fig. 3 shows an example of a Data Cube with 3-D (x, y, z). Each cell in the Data Cube consists of three dimensions: Time x, Day y, and Task z. For example, $cell_1$ equals (1, 1, task1), cell2 equals (1, 2, $task_2$), etc. In addition, each cell is attached with a list of preferences for the purpose of scheduling using the Distance Function. This will be explained in the next section.

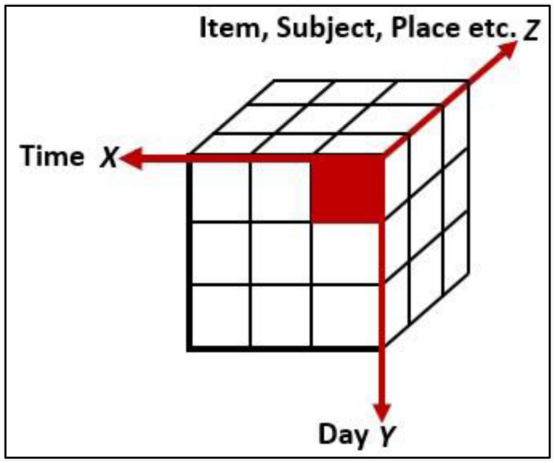

Fig. 3. Example of a data cube with 3-D.

A Data Cube is initialized with non-scheduled cells. One task at a time is scheduled at a Data Cube's cell using the Distance Function. Only the empty cells, ready for scheduling, are visited during the scheduling process. Fig. 4 shows the process of scheduling a stream of data, where Data refers to a task with a list of preferences. The Data Cube cells with Red color refer to tasks that are recently scheduled. The Data Cube cells with White color refer to the scheduled tasks at the current time. At the same time, the Data Cubes cells with Gray color refer to the empty Data Cube cells, i.e., ready for scheduling.

Data scheduling using distance function

Each task has its own preferences which can be identified by a user. For example, each classroom can be identified with a specific preference such as maximum number of students allowed in the classroom,

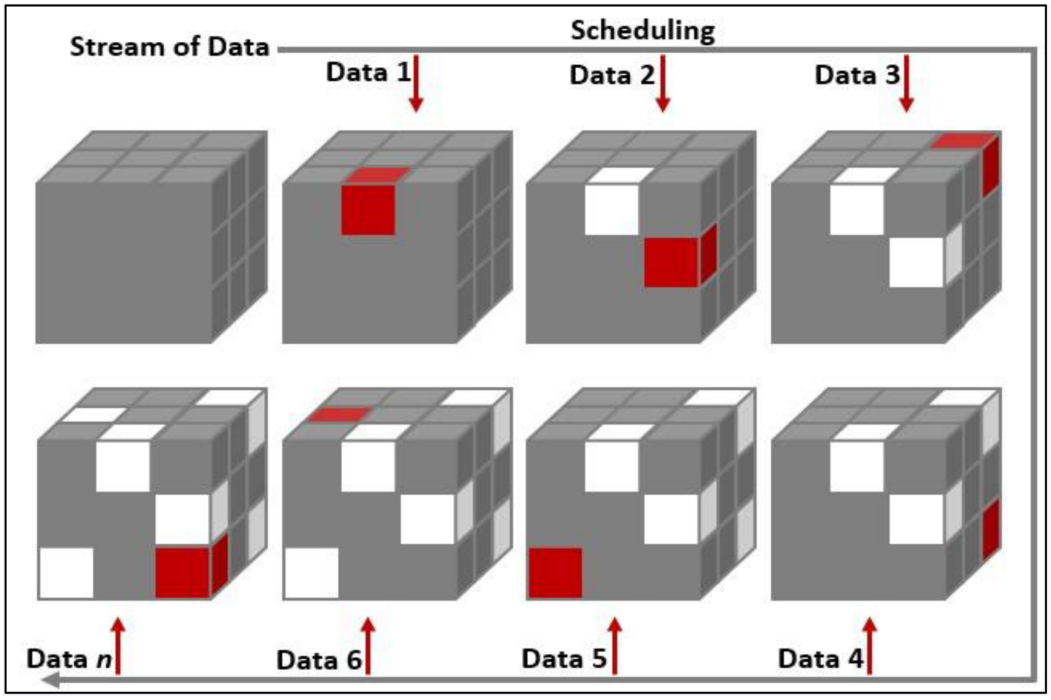

Fig. 4. Example of scheduling of stream of data.

smart devices, laptop, data show, time of availability, etc. These preferences can also be used for the lecturers as a preference. Hence, the Distance Function can be applied using the preferences of the classrooms and the lecturers, respectively, for the purpose of scheduling. The Euclidean distance is applied, measuring the shortest distance between two points with two or more dimensions. Eq. (1) shows the formula of the Euclidean distance.

$$distance\left(Task_p.\ Cell_p\right) = \sqrt{\sum_{i=1}^{n}\left(Task_{p_i} - Cell_{p_i}\right)^2} \quad (1)$$

Where p refers to preferences; Taski and $Cell_i$ at time i are both attached with a list of preferences $\{p_1, p_2, p_3 \ldots p_n\}$. $Task_i$ is then scheduled at the nearest Cell using Eq. (1) in terms of the list of preferences. This will be highlighted in more detail in the next section (case studies). Fig. 5 shows an example of scheduling based on nearest distance. Algorithm 1 shows the main steps of the scheduling process using Eq. (1) in terms of the list of preferences of both a task and Data Cube cells.

The methodology was applied to three case studies, the details of which are presented below:

Case Study 1: In this research study, the preferences are indexed for the purpose of identifying the nearest Data Cube cell. For example, assume that the axes of a 3-D Data Cube are x, y, z. Where x refers to

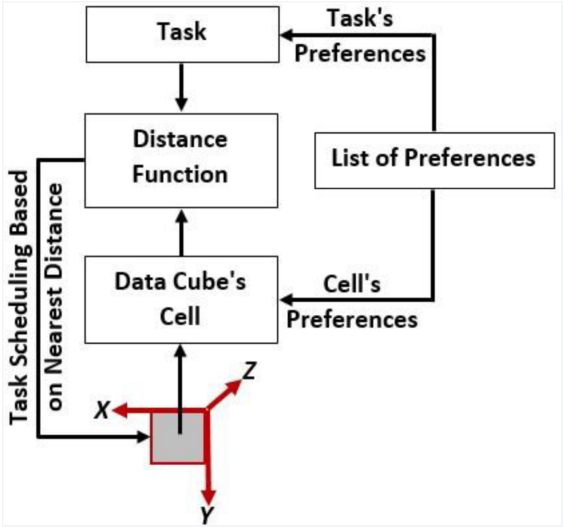

Fig. 5. Example of data scheduling using distance function.

Time, y refers to Day, and z refers to a classroom, as shown in Fig. 6.

A lecturer (L1) prefers a classroom with a maximum number of students (25 students), data show device, and the time of the lecture on Sunday at 10 am. Each preference is assigned to an integer number, either 0 or 1 value. The maximum number of students is assigned to 1, the data show device is assigned to 1, and the time of the lecture is assigned to 1 for both the day (Sunday) and the time (10 am). On the other

Algorithm 1: Scheduling Process

Input: New task t in combination with a list of preferences, Data Cube [cells]
Output: Scheduled task at nearest cell
1: Initialize $s \Leftarrow t$
2: $P_1 \Leftarrow$ List of t's preferences
3: **for** each empty cell in Data Cube **do**
4: $P_2 \Leftarrow$ List of cell's preferences
5: Compute Eq. (1) using P_1 and P_2 to identify nearest cell
6: $s \Leftarrow$ Scheduled task at nearest cell
7: **end for**
8: return s

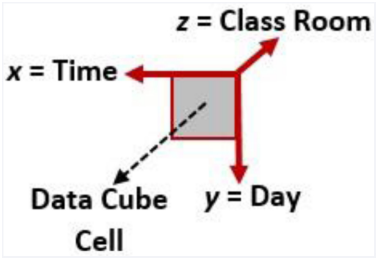

Fig. 6. Example of a data cube cell with 3-D for scheduling classrooms.

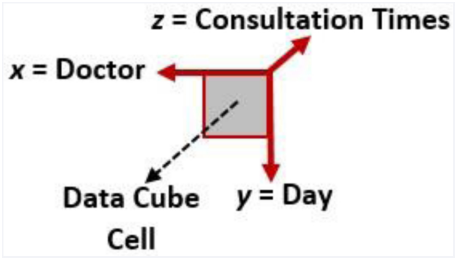

Fig. 7. Example of a data cube cell with 3-D for scheduling consultation times.

hand, the same approach can be used for the classrooms. Assume that the classroom (C1) is assigned the following preferences: a maximum of 20 students (0 value), no data shows device (0 value), and it is available on Sunday at 10 am (1 value for both). Assume that there is another classroom (C2) which is assigned the following preferences: maximum number of students is 25 (1 value), data show device (1 value), and it is available on Sunday at a different time (11 am as an example) (1 value for the day, 0 value for the time at 10 am). Then the Distance Function (1) can be used to calculate the distance between L1 and classrooms using the assigned preferences: distance (L1, C1) = 1.41, whereas distance (L1, C2) = 1. Based on the nearest distance value, L1 is assigned to C2 at 11 am. Table 1 shows the description of the above example.

Case Study 2: Another example is patient scheduling in the emergency department. Assume that the axes of a 3-D Data Cube are x, y, z. Where x refers to the doctor, y refers to the day, and z refers to consultation times, as shown in Fig. 7.

A patient (P1) prefers to be consulted by a doctor who specializes in Allergists, Autoimmune, and Immunologists, and the time of consultation is on Sunday at 10 am. Each preference is assigned to an integer number, either 0 or 1; medical specialties are assigned to 1, and the time of consultation is assigned to 1 for both the day (Sunday) and the time (10 am). On the other hand, the same approach can be used for the doctors. Assume that doctor (D1) is assigned to these preferences, specialists in Allergists and Immunologists; then they are assigned to 1 value except Autoimmune, and D1 is available on Sunday at 9 am, then 1 value for Sunday and 0 value for the time. Assume that there is another doctor (D2) who is assigned to these preferences: Allergists, Autoimmune, and Immunologists (1 value for each), and he is available on Sunday at 11 am; then 1 value for Sunday and 0 value for the time. Then the Distance Function (Eq. (1)) can be used to calculate the distance using the assigned preferences: distance (P1, D1) = 1.41, whereas distance (P1, D2) = 1. Based on the nearest

Table 1. The preferences assigned for a lecturer and the available classrooms.

	Maximum Number of Students (25) (either 0 or 1)	Data Show Device (either 0 or 1)	Day (Sunday) (either 0 or 1)	Time (10 am) (either 0 or 1)	Distance
L1	1	1	1	1	
C1	0	0	1	1	1.41
C2	1	1	1	0	1

Table 2. The preferences assigned for a patient and the available consultation times.

	Allergists (either 0 or 1)	Immunologists (either 0 or 1)	Autoimmune (either 0 or 1)	Day (Sunday) (either 0 or 1)	Time (10 am) (either 0 or 1)	Distance
P1	1	1	1	1	1	
D1	1	0	1	1	0	1.41
D2	1	1	1	1	0	1

distance value, P1 is assigned to D2 at 11 am. Table 2 shows the description of the above example.

Please note that the given examples in Case Study 1 and 2 assumed that there are 2 cases that need to be scheduled at specific options. However, in real life, options and cases are varied. In this research study, each Data Cube cell and a task are assigned preferences. Hence, data scheduling can be applied using the Distance Function, where a task is scheduled in terms of its preferences and nearest distance. The more preferences assigned to the Data Cube cells and tasks, the more accurate scheduling can be achieved. Once a Data Cube cell is assigned to its nearest task, the cell will be ignored from the next scheduling process. Hence, the number of empty Data Cube cells decreases over time after each scheduling process unless there are no canceled scheduling tasks. Only a single pass over the tasks is required here in this study.

Results of the application of the methodology, together with those two case studies above, show that a preference-based approach to scheduling-in combination with Euclidean Distance Function-works efficiently in finding best matches between tasks and resources in multi-dimensional environments. Similarity can be measured more by encoding both task attributes and resource attributes into binary preference vectors (0 or 1) across several dimensions than just single attribute matching; this brings an idea of total compatibility higher up on the priority list over partial compatibilities. The method found a classroom as well as a patient scheduler's nearest available option even when no option fully satisfied all preferences, hence demonstrating the ability to handle conflicts plus limited availability situations. Progressive elimination of Data Cube cells prevents duplicates and supports efficient one-pass scheduling, thereby making it flexible and scalable for different real-world scenarios.

Evaluation

The performance of the proposed approach is evaluated based on precision, which measures the number of tasks scheduled correctly divided by the total number of tasks, as shown in Eq. (2):

$$P = T_c / (T_c + T_i) \quad (2)$$

where T_c refers to the number of tasks scheduled correctly in terms of their preferences. T_i refers to the number of tasks scheduled incorrectly.

Data description

Classrooms can be categorized based on students' crowd,[28] as well as facilities that are required to meet natural teaching environments.[29] Hence, nine different types of preferences are created as shown in Table 3. In addition, each day of the week is divided into five different times, which are 1st time, 2nd time, 3rd time, 4th time, and 5th time, where each time equals 1 or 2 hours, as an example. Please note that here, a list of preferences consists of one or more properties. For example, a lecturer prefers a classroom with seating at Table 3, a smart board, a data show, interactive computers, 2nd time on Mondays, and 3rd time on Thursdays.

Table 3. Properties of classrooms and labs.

Index	Lab and Classroom Properties
1	Contains a Smart Board
2	Contains a Data Show
3	Contains a television (TV)
4	Capacity of 40 students, *as an example*
5	Capacity of more than 40 students, *as an example*
6	Stadium seating or sloped floor
7	Seating at tables (i.e., Seminar Classrooms)
8	Interactive computers
9	Contains desks facing teaching areas

Assumes that there are three classrooms and five labs. Table 4 shows an example of preferences of classrooms and labs. Where C refers to Classroom, and L refers to Lab.

It can be seen that C1 is available on Sundays and Wednesdays with a capacity of 40 students. Whereas C2 is not available on Mondays only, and it is available on the 1st and 2nd times during a day. Table 5 shows an example of preferences of subjects which are identified by lecturers.

Results and evaluation

This section presents an evaluation of the performance of the proposed approach in terms of the accuracy of scheduling streaming data with preferences.

Table 4. Preferences of classrooms and labs.

Classroom and Lab	Sun	Mon	Tue	Wed	Thu	1st time	2nd time	3rd time	4th time	5th time	Smart board	Capacity of 40 students	Capacity of more than 40 students
C1	1			1								1	
C2	1		1	1	1	1	1						1
C3													
L1													
L2													
L3		1			1	1	1	1			1		
L4		1				1	1						1
L5		1			1	1	1						1

Table 5. Preferences of subjects which are identified by lecturers.

Subjects	Sun	Mon	Tue	Wed	Thu	1st time	2nd time	3rd time	4th time	5th time	Smart board	Capacity of 40 students	Capacity of more than 40 students
Calculus					1	1	1						1
Computer Skills	1					1	1					1	
Digital Logic			1			1	1					1	
Discrete Structure				1		1	1						1
English	1					1	1						1
Freedom & Democracy			1			1	1						1
Programming Fun.				1		1	1					1	
Computer Skills Lab		1	1			1	1	1			1		
Digital Logic Lab		1			1	1	1						1
Programming Fun. Lab		1			1	1	1	1					1

Table 6 shows a comparison between applying dynamic scheduling and the ground truth. The preferences described in Tables 4 and 5 were used. C refers to a classroom. L refers to a Lab. Please note that here in this example, the ground truth scheduling is based on real data gathered from the University of Babylon, the Faculty of Information Technology (second year), for two groups of students, which were divided into two sub-groups for laboratory sessions. In Table 6, it can be seen that 14 corrected schedules were achieved after applying dynamic scheduling with the preferences. At the same time, 12 uncorrected schedules were recorded. Hence, the achieved accuracy equals 54%. However, the more preferences applied, the higher and greater accuracy might be achieved. Higher accuracy might be achieved through evaluating dynamic scheduling using well-known data stream classification techniques, i.e., test-then-train.[30] Hence, accuracy can be monitored on the fly. The model can be regenerated when the accuracy decreases. This could be one of the directions of future work. Another direction is using different real-world datasets. 54% scheduling accuracy is not fully representative since this was just a preliminary test on a small dataset with highly diverse tasks. There are several possible ways of enhancing the performance, such as imposing weighted Euclidean distance to emphasize critical attributes of the schedule and taking more than one nearby cube slice using a k-nearest-neighbor approach, balancing over-represented and under-represented task types by hybrid sampling, and refining the schedule based on initial results in an iterative manner. These will be attempted in future work aimed at improving overall scheduling accuracy so that better task allocation can be reported.

Results in Table 6 clearly support the effectiveness of this dynamic scheduling system relative to the ground-truth timetable. A large number of courses were perfectly matched (marked as “T”), which means that the algorithm has reproduced real scheduling decisions in all those cases where temporal constraints and resource availability do not strictly bind, thereby proving a high level of accuracy attainable by academic models using defined optimization criteria. However, a number of courses were flagged as mismatched (marked as “F”), clearly reflecting the presence of scheduling conflicts that the dynamic algorithm resolves differently from the actual timetable. The mismatches could be due to additional constraints not explicitly stated in the optimization model-instructor preferences, room availability limitations, or laboratory-specific requirements. In this case, the algorithm might have emphasized conflict resolution and workload balancing or even considered an overall feasible schedule rather than strictly duplicating the ground truth. Even with such variances, results in general clearly state a fact of being highly reliable and flexible, thus proving the capability to assist or even take over academic timetabling activities. Most importantly, results show the possibility of a dynamic approach as a decision support

Table 6. Comparison between applying dynamic scheduling and ground truth.

No.	Subject Title	Ground Truth	Dynamic Scheduling	Evaluation
1	Computer Skills 1	Sun at 1st time C1	Sun at 1st time C1	T
2	Computer Skills 2	Sun at 2nd time C1	Sun at 2nd time C2	T
3	Computer Skills Lab A	Mon at 1st time L3	Mon at 1st time L3	T
4	Computer Skills Lab B	Mon at 3Cd time L3	Mon at 2nd time L4	F
5	Computer Skills Lab C	Thu at 2nd time L3	Mon at 3Cd time L3	F
6	Computer Skills Lab D	Mon at 2nd time L3	Mon at 4th time L5	F
7	Programming Fun. 1	Wed at 2nd time C1	Wed at 1st time C1	T
8	Programming Fun. 2	Wed at 1st time C1	Wed at 2nd time C2	T
9	Programming Fun. Lab A	Mon at 2nd time L4	Mon at 2nd time L3	F
10	Programming Fun. Lab B	Thu at 1st time L3	Mon at 1st time L4	F
11	Programming Fun. Lab C	Mon at 1st time L4	Mon at 4th time L4	F
12	Programming Fun. Lab D	Mon at 3nd time L4	Mon at 3nd time L4	T
13	Digital Logic 1	Tue at 2nd time C1	Tue at 1st time C1	T
14	Digital Logic 2	Tue at 1st time C1	Tue at 2nd time C2	T
15	Digital Logic Lab A	Thu at 1st time L5	Mon at 3Cd time L5	F
16	Digital Logic Lab B	Mon at 1st time L5	Mon at 4th time L3	F
17	Digital Logic Lab C	Mon at 2nd time L5	Mon at 1st time L5	F
18	Digital Logic Lab D	Thu at 2nd time L5	Mon at 2nd time L5	F
19	Calculus 1	Thu at 2nd time C2	Sun at 3Cd time C3	F
20	Calculus 2	Thu at 1st time C2	Sun at 4th time C3	F
21	English 1	Sun at 1st time C2	Sun at 2nd time C1	T
22	English 2	Sun at 2nd time C2	Sun at 1st time C2	T
23	Freedom & Democracy 1	Tue at 1st time C2	Tue at 2nd time C1	T
24	Freedom & Democracy 2	Tue at 2nd time C2	Tue at 1st time C2	T
25	Discrete Structure 1	Wed at 1st time C2	Wed at 2nd time C1	T
26	Discrete Structure 2	Wed at 2nd time C2	Wed at 1st time C2	T

tool that can reduce manual human error and provide optimized alternatives in situations where classical timetabling fails due to hard constraint interactions.

The current evaluation was conducted using a single dataset from the University of Babylon due to restricted access to multi-institutional scheduling logs. To address this limitation, a baseline comparison was performed with a random scheduler and a first-come-first-served (FCFS) scheduler.

A complexity analysis shows that the proposed method operates in O(n) time, as each task is processed exactly once within the single-pass scheduling pipeline. Moreover, the model's scalability is supported theoretically, since cube aggregation and distance computation can efficiently handle larger datasets. Future work will focus on testing the approach on multi-institutional datasets to evaluate robustness and performance under varied scenarios.

Conclusion

The results of this study demonstrate that integrating data cubes with the Euclidean distance function provides an effective and innovative framework for task scheduling in complex, diverse environments, such as educational institutions and healthcare facilities. This approach provides a precise mechanism for analyzing various scheduling dimensions while enhancing resource allocation efficiency and minimizing temporal and spatial waste. By combining business intelligence techniques with spatial optimization tools, a flexible model was developed that adapts to changing scheduling needs across multiple domains. Therefore, the study recommends adopting this framework in future decision support systems and expanding it to include broader applications in other fields that require precise and efficient scheduling. Future research is also encouraged to explore alternative distance algorithms or machine learning techniques to further enhance allocation accuracy and overall system performance.

Authors' declaration

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Furthermore, any Figures and images that are not ours have been included with the necessary permission for republication, which is attached to the manuscript.
- No animal studies are present in the manuscript.
- No human studies are present in the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee at the University of Babylon.

Authors' contributions statement

M. S. H. designed the study. S. H. J. and M. H. J. performed the coding, experiments, and validation.

M. S. H. analyzed the data. M. S. H., S. H. J., and M. D. J. wrote the paper. M. S. H. did the revision, proofreading and reviewed the paper.

References

1. Tantalaki N, Souravlas S, Roumeliotis M. A review on big data real-time stream processing and its scheduling techniques. Int J Para E Dist. Sys. 2020 Sep 2;35(5):571–601. http://doi.org/10.1080/17445760.2019.1585848.
2. Mandelbaum A, Momčilović P, Trichakis N, Kadish S, Leib R, Bunnell CA. Data-driven appointment-scheduling under uncertainty: The case of an infusion unit in a cancer center. Manag. Sci. 2020 Jan;66(1):243–70. http://doi.org/10.1287/mnsc.2018.3218.
3. He S, Sim M, Zhang M. Data-driven patient scheduling in emergency departments: A hybrid robust-stochastic approach. Manag. Sci. 2019 Sep;65(9):4123–40. http://doi.org/10.1287/mnsc.2018.3145.
4. Ismail R, Othman Z, Bakar AA. Data mining in production planning and scheduling: A review. In2009 2nd Conf. on Data Mining and Optim. 2009 Oct 27 (pp. 154–159). IEEE. http://doi.org/10.1109/DMO.2009.5341895.
5. Arunarani AR, Manjula D, Sugumaran V. Task scheduling techniques in cloud computing: A literature survey. F Gen Comp Sys. 2019 Feb 1;91:407–15. http://doi.org/10.1016/j.future.2018.09.014.
6. Manikandan A, Madhu GC, Flora GD, Parvez MM, Begum MB. Hybrid Advisory Weight based dynamic scheduling framework to ensure effective communication using acknowledgement during Encounter strategy in Ad-hoc network. Int J Inf Tech. 2023 Dec;15(8):4521–7. http://doi.org/10.1007/s41870-023-01421-5
7. Senjab K, Abbas S, Ahmed N, Khan AU. A survey of Kubernetes scheduling algorithms. J C Comp. 2023 Jun 13;12(1):87. http://doi.org/10.1186/s13677-023-00471-1.
8. Jagadish Kumar N, Balasubramanian C. Hybrid gradient descent golden eagle optimization (HGDGEO) algorithm-based efficient heterogeneous resource scheduling for big data processing on clouds. Wire P Com. 2023 Mar;129(2):1175–95. http://doi.org/10.1007/s11277-023-10182-0.
9. Montero D, Kraemer G, Anghelea A, Aybar C, Brandt G, Camps-Valls G, Cremer F, Flik I, Gans F, Habershon S, Ji C. Earth system data cubes: Avenues for advancing earth system research. Env D Sci. 2024 Jan;3:e27. http://doi.org/10.1017/eds.2024.22.
10. Ha WY, Jiang ZP. Optimization of cube storage warehouse scheduling using genetic algorithms. In2023 IEEE 19th Int Conf Auto Sci Eng. (CASE) 2023 Aug 26:1–6. IEEE. http://doi.org/10.1109/CASE56687.2023.10260388.
11. Munteanu A. Data Cubes and Cloud-Native Environments for Earth Observation: An Overview. Scalable Computing: Prac Exp. 2024 Oct 1;25(6):5745–59. http://doi.org/10.12694/scpe.v25i6.4999.
12. Wardle J, Phillips Z. Examining Spatiotemporal Photosynthetic Vegetation Trends in Djibouti Using Fractional Cover Metrics in the Digital Earth Africa Open Data Cube. R Sens. 2024 Mar 31;16(7):1241. http://doi.org/10.3390/rs16071241.
13. Zhou D, Sheng M, Li J, Han Z. Aerospace integrated networks innovation for empowering 6G: A survey and future challenges. IEEE Com Surv Tut. 2023 Feb 16;25(2):975–1019. http://doi.org/10.1109/COMST.2023.3245614.
14. Mohammadzadeh A, Javaheri D, Artin J. Chaotic hybrid multi-objective optimization algorithm for scientific workflow scheduling in multisite clouds. J Oper Res Soc. 2024 Feb 1;75(2):314–35. http://doi.org/10.1080/01605682.2023.2195426.
15. Tu J, Chen H, Wang M, Gandomi AH. The colony predation algorithm. J Bio Eng. 2021 May;18(3):674–710. http://doi.org/10.1007/s42235-021-0050-y
16. Raju MR, Mothku SK. Delay and energy aware task scheduling mechanism for fog-enabled IoT applications: A reinforcement learning approach. Comp Net. 2023 Apr 1;224:109603. http://doi.org/10.1016/j.comnet.2023.109603.
17. Ghafari R, Kabutarkhani FH, Mansouri N. Task scheduling algorithms for energy optimization in cloud environment: a comprehensive review. Clus Comp. 2022 Apr;25(2):1035–93. http://doi.org/10.1007/s10586-021-03512-z
18. Mustapha SD, Gupta P. DBSCAN inspired task scheduling algorithm for cloud infrastructure. IoT Cyb. Sys. 2024 Jan 1;4:32–9. http://doi.org/10.1016/j.iotcps.2023.07.001.
19. Saif FA, Latip R, Hanapi ZM, Shafinah K. Multi-objective grey wolf optimizer algorithm for task scheduling in cloud-fog computing. IEEE Access. 2023 Jan 31;11:20635–46. http://doi.org/10.1109/ACCESS.2023.3241240.
20. Karimunnisa S, Pachipala Y. Task Classification and Scheduling Using Enhanced Coot Optimization in Cloud Computing. Int J Int Eng Sys. 2023 Sep 1;16(5). http://doi.org/10.22266/ijies2023.1031.43
21. Li Y, Han M, Shahidehpour M, Li J, Long C. Data-driven distributionally robust scheduling of community integrated energy systems with uncertain renewable generations considering integrated demand response. App Eng. 2023 Apr 1;335:120749. http://doi.org/10.1016/j.apenergy.2023.120749
22. Huang Z, Ravishankar S. Single-pass object-adaptive data undersampling and reconstruction for MRI. IEEE Tran Comp Img. 2022 Apr 14;8:333–45. http://doi.org/10.1109/TCI.2022.3167454.
23. Pal S, Jhanjhi NZ, Abdulbaqi AS, Akila D, Alsubaei FS, Almazroi AA. An intelligent task scheduling model for hybrid internet of things and cloud environment for big data applications. Sustainability. 2023 Mar 14;15(6):5104. http://doi.org/10.3390/su15065104.
24. Song J, Ma Y, Zhang Z, Liu P. An In-Memory Data-Cube Aware Distributed Data Discovery Across Clouds for Remote Sensing Big Data. IEEE J S Topics Applied Earth Obs R Sens. 2023 Apr 14;16:4529–48. http://doi.org/10.1109/JSTARS.2023.3267118.
25. Mahdi AJ, Esztergár-Kiss D. Supporting scheduling decisions by using genetic algorithm based on tourists' preferences. Applied S Comp. 2023 Nov 1;148:110857. http://doi.org/10.1016/j.asoc.2023.110857.
26. Abduljabbar IA, Abdullah SM. An evolutionary algorithm for solving academic courses timetable scheduling problem. Baghdad Sci. J. 2022 Apr 1;19(2):0399. http://doi.org/10.21123/bsj.2022.19.2.0399.
27. Souravlas S, Anastasiadou S. Pipelined dynamic scheduling of big data streams. Applied Sci. 2020 Jul 13;10(14):4796. http://doi.org/10.3390/app10144796.
28. Jiang W, Huang X, Zhao Q, Liu S. ClassRoom-Crowd: A Comprehensive Dataset for Classroom Crowd Counting and Cross-Domain Baseline Analysis. Eng Proc. 2025 Feb 11;78(1):10. http://doi.org/10.3390/engproc2024078010.
29. Hammoodi MS, Al-Azawei A. A proposed approach to discover nearest users on social media networks based on users' profiles and preferences. B Elect Eng Info. 2023 Aug 1;12(4):2464–73. http://doi.org/10.11591/eei.v12i4.4436.
30. Hammoodi MS, Stahl F, Badii A. Real-time feature selection technique with concept drift detection using adaptive micro-clusters for data stream mining. Knowledge-Based Sys. 2018 Dec 1;161:205–39. http://doi.org/10.1016/j.knosys.2018.08.007.

جدولة البيانات باستخدام دالة المسافة مع المعالجة أحادية المرور

محمود شاكر حمودي، سهاد حاتم جهاد، مثال هادي جبور

مركز الحاسوب، جامعة بابل، بابل، العراق.

الخلاصة

تُعد جدولة المهام إحدى أهم الصعوبات الأساسية التي تواجه العديد من القطاعات، بما في ذلك التعليم والرعاية الصحية والإدارة الصناعية. وتؤثر جودة الجدولة بشكل مباشر على كفاءة الأداء، واستخدام الموارد، ورضا المستخدم النهائي. ومع تزايد حجم البيانات وتنوعها، يتزايد الطلب على الأنظمة الذكية القادرة على التعامل مع التعقيد المكاني والزماني للوظائف وتخصيص الموارد. ونتيجةً لذلك، أصبحت عمليات تحليل البيانات واتخاذ القرارات الحديثة بالغة الأهمية لضمان تحسين سير العمل مع تقليل هدر الوقت والمكان. تقترح هذه الدراسة نهجًا مبتكرًا لجدولة المهام يجمع بين تحليل البيانات متعدد الأبعاد باستخدام مكعبات البيانات (وهي طريقة تُمكّن من تحليل البيانات متعدد الأبعاد) والتحسين المكاني باستخدام دالة المسافة الإقليدية Euclidean . وتتمثل الفكرة الرئيسية في تمثيل عناصر الجدولة، مثل المهام والموارد والأوقات والمواقع، في مكعب بيانات مُهيكل، مما يسمح بالتنفيذ السريع للاستعلامات والتجميعات متعددة الأبعاد. من خلال تقييم المسافة الإقليدية بين الوظائف والموارد المتاحة، قد يجد النظام أفضل تطابق بناءً على أقصر مسافة، مما يضمن كفاءة تخصيص الموارد. يُحسّن استخدام مكعب البيانات لدراسة أنماط الجدولة، واستخدام الموارد، وتخصيص الوقت، عملية اتخاذ القرار. تشير النتائج إلى فعالية النموذج وقابليته للتوسع. علاوة على ذلك، يُوفر دمج أدوات ذكاء الأعمال مع أساليب التحسين الهندسي أساسًا مُحتملًا لمعالجة مشكلات الجدولة المُعقدة في المواقف الديناميكية.

الكلمات المفتاحية: ذكاء الأعمال، مكعب البيانات، البيئات الديناميكية، المسافة الإقليدية، تحليل البيانات متعدد الأبعاد، تحسين الأداء، جدولة المهام.