DATAMIX: A Dynamic Adaptive Task Assignment Algorithm Based on Mixed Heuristics

21CSB0B07 B Bharadwaj

1. Introduction

Efficient task scheduling is a core problem in heterogeneous computing environments, where tasks must be assigned to machines (or virtual machines, VMs) to minimize the total completion time. This work introduces **DATAMIX**, a novel dynamic scheduling algorithm that blends two classic heuristics—Minimum Completion Time (MCT) and Minimum Execution Time (MET)—through an adaptive, threshold-based strategy. DATAMIX is shown to outperform traditional MCT on 9 out of 12 standard benchmark datasets, demonstrating its effectiveness in handling variability in task-machine mappings.

2. Problem Statement

The problem involves a matrix of task execution times, where each row represents a task and each column corresponds to a machine. Each cell in the matrix indicates the time required for a specific task to execute on a particular machine. The objective is to assign each task to exactly one machine in such a way that the maximum completion time across all machines — also known as the **makespan** — is minimized.

3. Motivation

The core motivation behind DATAMIX is that no single heuristic is universally optimal. MCT performs well when load balancing is key, whereas MET is better when tasks have diverse execution times. DATAMIX dynamically selects the better option per task, balancing between:

- **Load-awareness** (MCT)
- **Execution efficiency** (MET)

This adaptive nature helps it avoid performance pitfalls of static heuristics.

4. Algorithm

The input to the algorithm is provided as a file or matrix, where each entry represents the execution time of a specific task on a particular machine. This matrix forms the foundation for all subsequent scheduling decisions. During initialization, a vector is created to track the total completion time for each virtual machine (VM). All values in this vector are initially set to zero, indicating that no tasks have been assigned yet. The task assignment procedure is carried out iteratively for each task in the dataset. For every task,

the algorithm identifies two candidate machines. The first, referred to as **Machine 1 (mn1)**, is the machine with the minimum expected completion time when both the task execution time and the current machine load are considered—this reflects the Minimum Completion Time (MCT) heuristic. The second, **Machine 2 (mn2)**, is the machine that offers the fastest execution for the task, ignoring load—this corresponds to the Minimum Execution Time (MET) heuristic. A dynamic threshold is computed based on the statistical characteristics (such as the range) of the task's execution times across machines. The algorithm then compares the completion times for assigning the task to mn1 and mn2. If the difference in completion times is below the threshold, the task is assigned to mn2, prioritizing execution efficiency. Otherwise, it is assigned to mn1, promoting load balancing. After the assignment, the selected machine's total completion time is updated accordingly. The final output of the algorithm is the maximum value in the completion time vector. This value represents the makespan—the total time taken to complete all tasks when scheduled according to the algorithm.

5. Results

The following table presents the performance (makespan) of various scheduling algorithms across 12 standard benchmark instances. The compared algorithms include OLB, MCT, MET, SA, kPB, and our proposed method DATAMIX. Lower values indicate better performance.

Instance	OLB	MCT	MET	SA	kPB	DATAMIX
u_c_hihi.0	14 376 662.175	11 422 624.494	47 472 299.429	16 213 221.101	12 496 863.706	**12 588 764.440**
u_c_hilo.0	221 051.823	185 887.404	1 185 092.968	194 549.794	201 153.956	**190 692.488**
u_c_lohi.0	477 357.019	378 303.624	1 453 098.003	426 271.390	400 291.050	**390 061.114**
u_c_lolo.0	7306.595	6360.054	39 582.297	8167.052	6846.273	**6342.293**
u_i_hihi.0	26 102 017.618	4 413 582.982	4 508 506.791	4 692 192.002	4 508 655.928	**3 978 575.718**
u_i_hilo.0	272 785.200	94 855.913	96 610.481	98 002.382	93 005.897	**84 634.440**
u_i_lohi.0	833 605.654	143 816.093	185 694.594	143 905.246	131 816.083	**130 200.243**
u_i_lolo.0	89 380.269	3137.350	3399.284	3485.290	3122.956	**2997.179**
u_s_hihi.0	19 464 875.910	6 693 923.896	25 162 058.136	7 127 729.951	6 514 162.148	**6 501 910.287**
u_s_hilo.0	250 362.113	126 587.591	605 363.772	149 050.289	123 543.792	**117 815.334**
u_s_lohi.0	603 231.467	186 151.286	674 689.535	194 318.366	187 955.955	**176 034.107**
u_s_lolo.0	8938.389	4436.117	21 042.413	5836.962	4405.247	**4268.668**

Table 1: Makespan comparison across 12 datasets. Bold values indicate the best performance.

6. Conclusion

DATAMIX is a practical, lightweight, and adaptive task scheduling algorithm designed for heterogeneous computing environments. By dynamically switching between MCT and MET based on a threshold, it achieves better makespan in most test cases and

demonstrates robust generalization across datasets. This makes it a strong candidate for cloud VM scheduling, grid computing, and similar distributed task assignment scenarios.

Keywords

Task Scheduling, Adaptive Heuristic, Minimum Completion Time (MCT), Minimum Execution Time (MET), Makespan Minimization, DATAMIX