Resource Allocation in AI/HPC Server using Multi-objective Optimization

Rashmikant*, Hiroshi Ito, Takashi Minabe, and Masamichi Nakamura

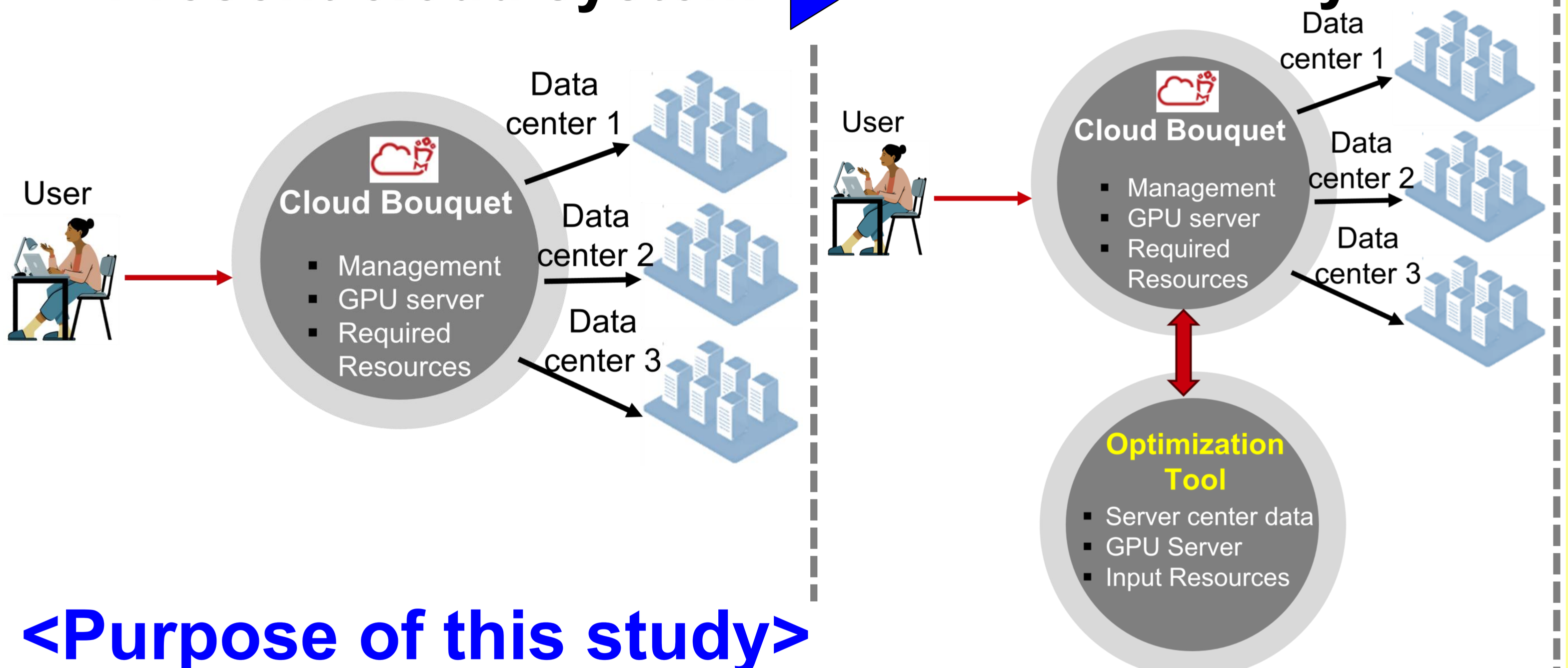
Morgenrot Inc. Japan

Keywords: Multi-objective Optimization, Resource Allocation, AI/HPC Server, Genetic Algorithm, Heuristic Allocation Methods.

Abstract

◆ Concept of cloud system for efficient resource utilization.

Present cloud system ▶ Aimed cloud system



<Purpose of this study>

Multi-objective optimization tool for HPC/AI data center.

Problem Setup and Input Data

- Server availability is dynamic i.e., varying with time and session
- Heterogeneous resource (CPUs, GPUs, sockets).
- Real-world case study of GPU-enabled data centers operated by Morgenrot.Inc

Table 1: Capacity of data center and each server

GPU server model	Number of GPU server	Total number of GPU core	Total Number of GPU card
H100	5	384	8
L40S	5	256	4

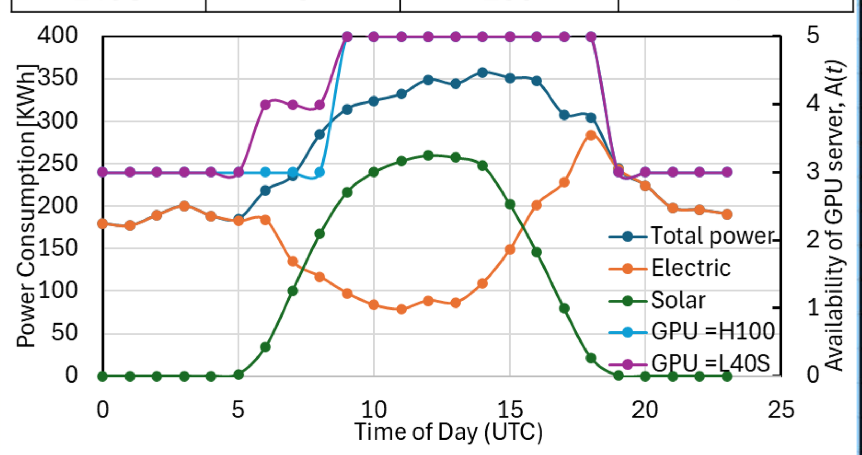


Fig. 1 Power consumption and Server availability for one day

➢ **Time resolution**: 1-minute granularity

Table 2: Number of VMs and corresponding required resource

VMs	CPU	GPU	GPU_Model	Duration	Start	Priority
A	18	1	H100	90	30	3
B	38	2	H100	120	60	3
C	76	4	H100	180	90	3
D	152	8	H100	360	0	3
E	18	1	H100	60	30	2
F	38	2	H100	90	120	2
G	76	4	H100	540	180	2
H	152	8	H100	540	360	2
I	18	1	H100	1080	60	2
J	38	2	H100	120	90	1
K	76	4	H100	180	240	1
L	152	8	H100	540	120	1
M	18	1	H100	120	30	1
N	38	2	H100	90	0	1
O	76	4	H100	180	90	1
P	152	8	H100	540	180	1
Q	50	1	L40S	120	0	1
R	100	2	L40S	90	30	1
S	200	4	L40S	360	60	1
T	50	1	L40S	180	540	1
U	100	2	L40S	540	180	1
V	200	4	L40S	1080	0	1
W	50	1	L40S	900	90	1
X	100	2	L40S	180	600	1

Introduction

◆ Due to growth of computational demand, an effective resource server management system required in next generation data center.

◆ Optimization for resource (CPU only) allocation in a static server using single objective GA based [1-2].

◆ Development of Energy-aware and load-balanced virtual machine (VM) placement schemes [3].

<Significance of this study>

- Multi-objective: Max. heterogeneous resource and min. power consumption.
- Dynamic server availability

Multi-objective Optimization Method

- Heuristic Methods; First cum first serve (FCFS)
- Multi objective optimization methodology; (1) GA, (2) NSGA-III, and (3) Bayesian
- Surrogate modelling based Multi objective optimization methodology

Table 3 : Comparison among various optimizing algorithm.

Feature	GA	Bayesian	Surrogate	NSGA-III
Type	Evolutionary	Probabilistic	Model-based	Multi-objective
Uses Surrogate	No	Yes	Yes	Hybridized
Output	Single	Single	Few	Pareto Front
Strength	Global Search	Efficient Sampling	Fast Prediction	Balanced Trade-off

Genetic Algorithm

- Because of simplicity, adaptability, and inherent ability to explore large, complex solution effectively, GA is used at first [4].

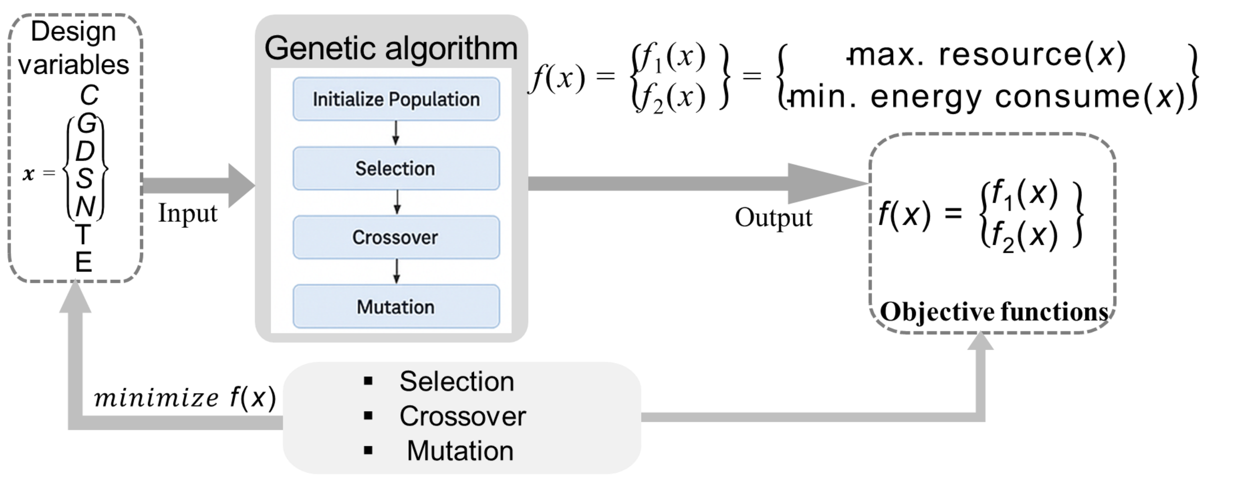


Fig. 2 GA based optimization for resource allocation.

C — Number of CPUs, G — Number of GPUs, D — Duration of VM, S — Number of sockets, N — Number of Servers, T — Start time, E — Energy consumption.

Optimizing Methodology

Genetic Algorithm

❑A VM_j is allocated to a server s at a time t only when

$$CPU_j \leq CPU_s(t) \text{ and } GPU_j \leq GPU_s(t)$$

❑**Fitness function** guiding GA optimization is defined as;

$$F = N_{VMs} + \alpha(U_{CPU} + U_{GPU})$$

N_{VMs}= number of assigned VMs, U_{CPU} and U_{GPU} = normalized utilization ratios of CPU_s and GPU_s & α = 0.5.

➢ **Allowable waiting time t_{allow} (min) based on priority p**

p = 3, t_{allow} = 30; p= 2, t_{allow} = 360; p = 1, t_{allow} = 900 min

➢ POPULATION =30; GENERATION =60; MUTATION =0.35

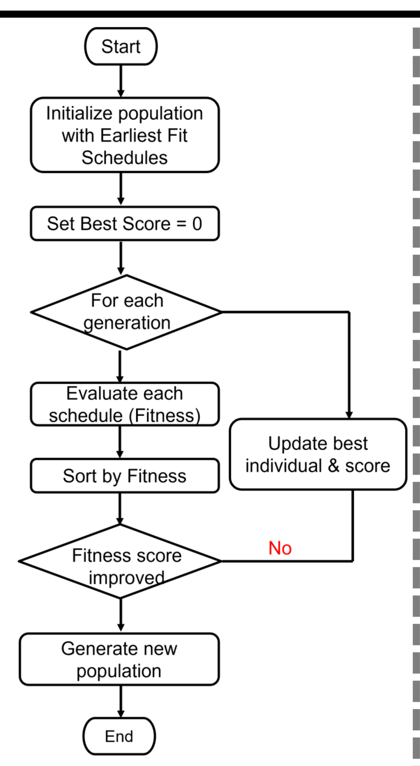


Fig. 3 GA flow chart.

Heuristic methods

Heuristic methods mainly first-cum first serve (FCFS) are used here.

➢ FCFS are used for complex problems where finding the perfect answer is too slow like cloud computing.

➢ FCFS can be easily Implemented.

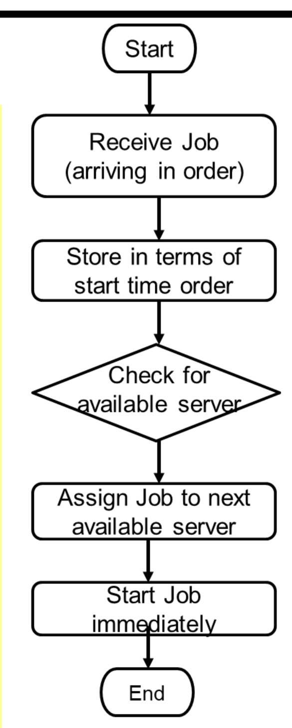


Fig. 4 FCFS flow chart.

Metrix parameters

Various parameters are used to compare the optimizing results [5].

➢ **Makespan :** Finishing time of the last task

➢ **Throughput :** Total number of VMs assigned per unit time.

➢ **Utilization Efficiency :**

$$U = \frac{\sum_i (C_i + G_i)}{\sum_i N_i}, \text{ s.t. } A(t) \leq N_j, \forall j, t$$

Results

Notes

1. Resources has been optimized per server basis as resource can't interchange between servers.
2. If a full GPU has been assigned to a VM of a server, then the assigned CPU will be used and remaining CPU will not be utilized.

Optimizing Server

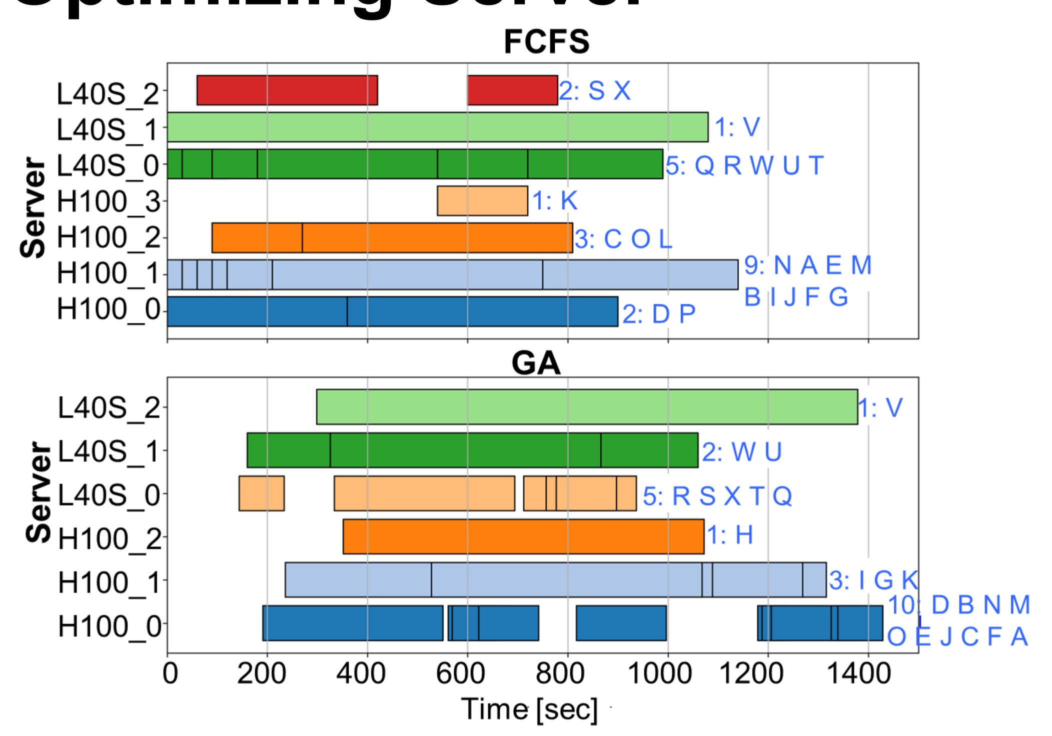


Fig. 5 Number of VM, assign to each server.

Reason for high variance in GA, for each VMs **random start time ≥ its minimum start** and is assigned to a **random server**.

Fitness Score

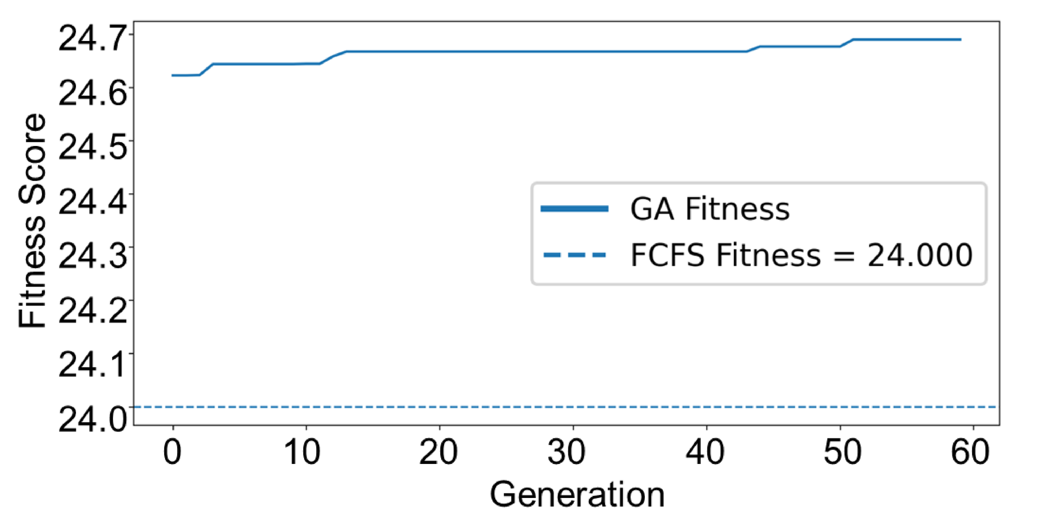


Fig. 6 Fitness score against each generation.

❑Fitness score shows, how many number of VMs successfully assigned to server.

❑In FCFS, all VMs assigned instantly and it's not evolving, so it is a straight line.

❑In GA, fitness score is evolving and calculated by "Fitness function formula".

Metrix Comparison

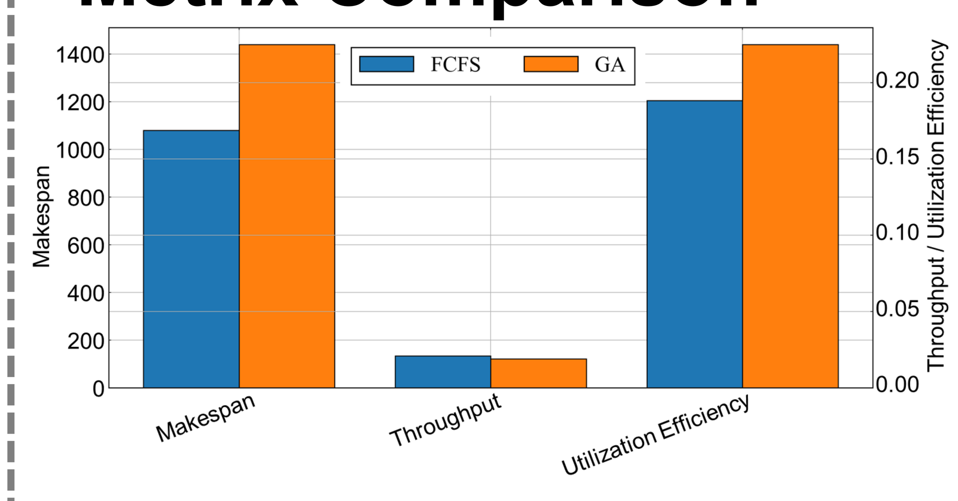


Fig. 7 Fitness score against each generation.

❑Execution time depend on ***U***.

❑GA yields a higher throughput than the FCFS.

❑GA consolidate workloads to utilize servers more efficiently, whereas FCFS leave resources idle.

Conclusion

1. GA adapts to workload heterogeneity and outperforms traditional FCFS in efficiency.
2. Avoids unnecessary activation of power-intensive servers.
3. The proposed GA-based multi-objective scheduling approach improves resource utilization.
4. The proposed methods provide a scalable path toward more energy-efficient data center operations.

References

1. H. Ma, and J. Fang, *Proc. of International Conference on Big Data and Intelligent Algorithms*, 2021.
2. P. Haskul, *Seminar report*, 2025.
3. Z. Li, *Journal of Grid Computing*, 2022.
4. C. Panggabean, et. al., *ArXiv*, 2025.
5. I.P. Oladoja, et.al., *International Journal of Computer Applications*, 2021.