Svedberg Open
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Hybrid Stochastic-Dynamic Framework for Latency and Energy-Aware Task Offloading in Edge Computing

Dr. TKS Rathish Babu[1], R. Naveenkumar[2], Dr.R.Shankar[3], Ali Bostani[4], Mrs.E.Pavithra[5], Ponmurugan Panneerselvam[6], Dr .P.DharmendraKumar[7]

[1]Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai - 600 089. Email: tksbabu80@gmail.comhttps://orcid.org/0009-0008-3830-8170

[2]Dept of CSE, School of Engineering and Technology, CGC University Mohali-140307, Punjab India. Email: drnk1983@gmail.com 0000-0001-9033-9400

[3]Assistant Professor, Department of Computer Science (PG), Kristujayanti deemed to be university,Bengaluru – 560077 , India. Email: rshankar@kristujayanti.com

[4]Associate Professor, College of Engineering and Applied Sciences, American University of Kuwait, Salmiya, Kuwait. Email: abostani@auk.edu.kw, 0000-0002-7922-9857

[5]Assistant Professor, Department of Computer Science and Engineering (AIML), Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai. Email: pavithrae@veltech.edu.in,ORCID ID: https://orcid.org/0009-0009-3421-1475?lang=en

[6]Professor& Dean-Doctoral Studies & IPR, Department of Research, Meenakshi Academy of Higher Education and Research, Chennai, tamilnadu. India, Email: ponmurugan@maher.ac.in

[7]Assistant professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India, Email: pdharmendrakumar@kluniversity.in, dharmendra.phd.au@gmail.com, https://orcid.org/0009-0006-5087-9123

Abstract

A system that enables easy, network-on-demand access to a shared and customizable pool of computer resources is cloud computing. Resource provisioning, a key component of Infrastructure as a Service in the cloud, encounters issues with pricing ambiguity, heterogeneity, availability, and demand consideration. To solve the problems, approximation dynamic programming and stochastic programming (SP) are offered, which increase the cloud computing framework's four main uncertainties: price, heterogeneity, demand, and availability. A tree is built to reduce the space in SPDP, a multi-stage paradigm for dynamic programming that takes uncertainty into account in the underlying probability space. The reduction of space successfully promotes exploration, resource over- and under-provisioning, and significant profit-enhancement.The suggested method exemplifies responsive pricing research that is advantageous to both users and suppliers. As compared to other accessible algorithms, the SPDP provides promising results, according to the results analysis.

Keyword: Virtual machine, cloud computing, resource provisioning, stochastic optimization, and dynamic programming.

1. Introduction

Distributed computing gives computational assets high accessibility and sensible costs. Cutting edge versatile applications, for example, facial acknowledgment, picture and video preparing, have expanded the utilization of on-request computational assets. Hence, offloading applications to have them run on cloud servers has developed as an appealing arrangement so as to misuse the limit and effectiveness of distributed computing, just as sparing restricted intensity of cell phones and quicken the speed of executing asset escalated applications. As more applications start to have higher latency requirements, mobile edge computing (MEC) and Customer-premises equipment (CPE) devices are also getting attentions and attractions in the industry. MEC is taking Network Function Virtualization (NFV) to the end clients closer than any time in recent memory [1].

Occasions of edge figuring, including provincial datacenters, cloudlets [2], and mist hubs, convey exceptionally responsive cloud administrations at the system edge. As a key innovation towards 5G, MEC design proposed by ETSI use existing NFV structures broadly received via bearers and sellers [3]. Versatile portable edge applications, including system administrations, are conveyed near the client hardware (UE) with low idleness. Both UE application suppliers and media transmission specialist co-ops (TSPs) can exploit MEC to lessen cost and to alter administrations with spryness dependent on fast changing client requests. As an enhancement to the MEC-empower base stations sent by the specialist organizations (SPs), CPE gadgets, looking for revamping a unique income stream.

One basic issue to determine in offloading applications is the means by which to deal with the security dangers of distributed computing [4] and MEC. Numerous versatile applications use and store individual data identified with banking, wellbeing, business, informing, etc. Delicate data sent to the remote cloud or MEC has a decent opportunity to be presented to either specialist co-ops or noxious clients who approach similar equipment [5]. Conventional approaches to ensure remote execution and information stockpiling incorporate common verification, approval and information encryption for the entire application. On the off chance that each and every piece of the application was shielded by solid and steady security assurances, the framework would be sufficiently protected however the expense would likewise rocket up and not be worthy for business use, in light of the fact that a lot of data must be encoded superfluously. The repetitive encryption and decoding activities hurt both vitality productivity and client experience. This work has to consider the exchange offs among security and ease of use, and to augment the framework security subject to a middle of the road deferral and asset cost.

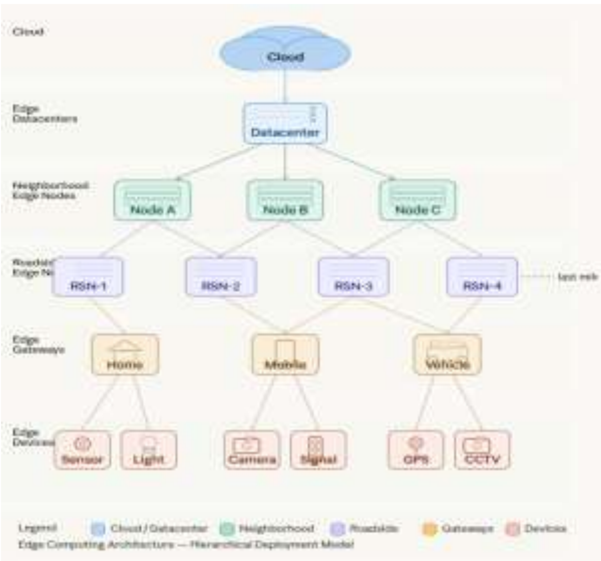

Figure 1: Architecture of Edge Computing

Obviously, the edge processing structure needs a situation administration to progressively check the client needs and the accessible edge has and decide the position or expulsion of edge applications.

2. Related Works

Recent research on Mobile Edge Computing (MEC) task offloading focuses on optimizing latency, energy consumption, and resource allocation using intelligent paradigms such as agent-based models, deep reinforcement learning, and swarm optimization. These approaches address scalability, heterogeneity, and computational constraints, achieving near-optimal performance across diverse edge environments while improving Quality of Experience (QoE). The comprehensive analysis is given in Table 1.

Table 1: Comprehensive Analysis of Literature

Author(s) & Year	Problem Addressed	Methodology / Model	Key Techniques Used	Optimization Objectives	Datasets / Tools	Key Findings / Contributions
Rui Wang et al. (2019) [6]	Inefficient task offloading due to information asymmetry in UAV-supported MEC	Agent-enabled task offloading framework in UAV-supported MEC (UMEC)	Intelligent agent-based decision making	Minimize task execution delay and energy consumption	Simulation-based evaluation	Agent-driven framework significantly reduces latency and energy; improves QoE via optimal offloading strategy
Haifeng Lu et al. (2020) [7]	Scalability issues in large-scale heterogeneous MEC with multiple service nodes	Improved Deep Reinforcement Learning (IDRQN) with LSTM integration	DRL, LSTM, DQN enhancement	Reduce energy consumption, latency, and improve load balancing	iFogSim, Google Cluster Trace	Proposed IDRQN outperforms baseline algorithms in execution time, load balancing, and energy efficiency
Liang Huang et al. (2020) [8]	Joint optimization of task offloading and resource allocation in MEC	DQN-based joint optimization framework	Deep Q-Network (DQN), Mixed Integer Nonlinear Programming (MINLP)	Minimize overall cost (energy, computation, delay)	Numerical simulations	Achieves near-optimal performance for complex NP-hard optimization problems in MEC environments
Kai Lin et al. (2018) [9]	Energy-efficient task offloading in resource-constrained edge environments	Fruit Fly Optimization Algorithm-based Task Offloading (FOTO)	Swarm intelligence (FOA), heuristic optimization	Minimize energy consumption, execution time, and cost	Comparative experimental setup vs CMS-ACO, GA-ACO	FOTO demonstrates superior efficiency over heuristic and ACO-based methods in MEC task scheduling

3. Proposed Methodology

Dynamic programming, which creates the best admission rules, aids in the formulation of stochastic programming. Dynamic programming, which solves control problems dynamically, has grown in popularity over the years. Data centres are made up of a substantial number of nodes, which makes the total system quite massive. Instances in the system can only be solved via dynamic programming if they are small. To address this problem, stochastic programming is combined with a multistage recourse model and approximation dynamic programming. Resource provisioning lowers the cost of VM and network providers. The following is a description of the problem construction:

$$\min_{A_{rl}^{RI} B_{ipk}^{RI}} \sum_{r\in RI}\sum_{p\in CP}\sum_{u\in UI}\sum_{V_i\in VI}\sum_{k\in CI}\sum_{l\in BCI}\left(CT_{rl}^{RI}A_{rul}^{RI} + CT_{ipk}^{RI}B_{uipk}^{RI} + E_{\delta}\left[Q\left(A_{rul}^{RI}, B_{uipk}^{RI}, \omega\right)\right]\right) \text{----} \quad (1)$$

The objective function lowers the cost of resource supply and lowers initial reservation costs. In the next phases of the cloud architecture, the anticipated cost is reduced to a minimum. Table 1 contains the parameters and their descriptions.

Table 2: Mathematical Notations

Symbol	Definition
UI	Group of users in the framework utilization phase, while u ∈ UI signifies the user index
CP	Group of cloud providers in the framework utilization phase, while p ∈ PI signifies the index of VM
VI	Group of VM classes, $V_i \in \{V_1, V_2 \dots V_{end}\}$ signifies the class index of VM
RI	Group of all network routers, while RI={1….RI} signifies index of the router
TS	Provisioning stage set while t∈ TS signifies the index of the time stage
BCI	Reservation contract set for bandwidth while $BCI_1 \in \{BCI_1, BCI_2, \dots, BCI_{last}\}$ signifies the index of the contract
CI	Reservation contract set for VMs, while $CI_1 \in \{CI_1, CI_2 \dots CI_{last}\}$ signify the index of the contract
t_p^p, t_p^s, t_p^{in}	The overall capacity of cloud providers for processing, storage, and bandwidth for internal works
t_r	Routers capacity for bandwidth
D_p^p, D_p^s, D_p^{in}	Processing, storage, and bandwidth for internal works where the resource of demand in the VM classes
D_i^b	Internet for internal works where the resource of demand in the VM classes
$v_{iut}(\omega)$	Count of VMs necessary under a scenario of ω
$CI_{ipk}^{RI}, CI_{rl}^{RI}$	Bandwidth and VMs for fixed first stage cost
$CI_{ipkt}^{res}(\omega)$, $CI_{ipkt}^{util}(\omega)$, $CI_{ipkt}^{od}(\omega)$	Under every stage of the scenario VM cost reservation, on-demand, and utilization
$CI_{rlt}^{res}(\omega)$, $CI_{rlt}^{util}(\omega)$, $CI_{rlt}^{od}(\omega)$	Under every stage of the scenario bandwidth cost reservation, on-demand, and utilization
$CI_{ipkt}^{s}(\omega)$, $CI_{ipkt}^{b}(\omega)$	Outbound and storage bandwidth price for VM provisioning
$A_{ipk}^{RI}, A_{rul}^{RI}$	VM reservation and bandwidth for deterministic decision variable at the first stage
$A_{rult}^{res}(\omega)$, $A_{rult}^{util}(\omega)$, $A_{rult}^{od}(\omega)$	Utilization, reservation, and on-demand allocation for the decision variable
$B_{uipkt}^{res}(\omega)$, $B_{uipkt}^{util}(\omega)$, $B_{uipkt}^{od}(\omega)$	Utilization, VM reservation, and on-demand allocation for the decision variable
δ	All scenario, while ω belongs to δ signifies the index of scenario

$B_{uipk}^{RI} \in N_0, \forall_{util} \in UI, \forall VI_i \in V, \forall_p \in CP, \forall k \in CI$ ---- (2)

$A_{rul}^{RI} \geq 0, \forall_r \in RI, \forall_{util} \in UI, \forall l \in BCI$---- (3)

The above equation ensures that the values in the VM reservation section are integer and the bandwidth of the system is non-negative values. The expected price of the system under the scenario of uncertainty δ is equated as,

$$E_\delta\left[Q\left(A_{rul}^{RI}, B_{uipk}^{RI}, \omega\right)\right]$$

where the provisioning price of the system is reduced by,

$$Q\left(A_{rul}^{RI}, B_{uipk}^{RI}, \omega\right)$$

The six decision variable is used in the uncertainty scenario ω that reduces the cost with the assistance of cost function. The cost function is given below,

$$\begin{aligned}
Q\left(A_{rul}^{RI}, B_{uipk}^{RI}, \omega\right) = min \sum_{r \in RI} \sum_{util \in UI} \sum_{l \in BCI} \sum_{t \in TSl} CT_{rlt}^{res}(\omega) A_{rult}^{res}(\omega) + \sum_{util \in UI} \sum_{V_i \in V} \sum_{p \in BCP} \sum_{k \in CI} CT_{ipkt}^{res}(\omega) B_{uipkt}^{res}(\omega) \\
+ \sum_{r \in RI} \sum_{util \in UI} \sum_{l \in BCI} \sum_{t \in TSk} CT_{rlt}^{res}(\omega) A_{rult}^{res}(\omega) + CT_{rlt}^{od}(\omega) A_{rult}^{od}(\omega) \\
+ \sum_{util \in UI} \sum_{V_i \in V} \sum_{p \in BCP} \sum_{k \in CI} \sum_{t \in TS} ((CT_{ipkt}^{util}(\omega) + CT_{ipkt}^{st}(\omega) D_i^{st} + CT_{ipkt}^{ob}(\omega) D_i^{ob}) B_{uipkt}^{util}(\omega) \\
+ CT_{ipkt}^{od}(\omega) CT_{ipkt}^{st}(\omega) D_i^{st} + CT_{ipkt}^{ob}(\omega) D_i^{ob}) B_{uipkt}^{od}(\omega)
\end{aligned}$$

---- (4)

The above equation's variables stand for the bandwidth and VM allocation for each stage of resource provisioning. The cost of overall provisioning is assessed based on a variety of factors, including providers, routers, customers, contracts, and different time periods, with weights applied based on the appropriate cost of providing. The aim and cost function formulate the optimal stochastic programming.

$B_{uipkt}^{util}(\omega) \leq \sum_{TS \in M_{kt}} B_{uipkt}^{res}(\omega), \forall U \in UI, \forall VI_i \in V, \forall_p \in CP, \forall k \in CI, \forall ts \in TS$---- (5)

$A_{rult}^{util}(\omega) \leq \sum_{TS \in M_{lt}} X_{rult}^{res}(\omega), \forall U \in UI, \forall_r \in RI, \forall l \in LI, \forall ts \in TS$---- (6)

The bandwidth and virtual machine are granted during the use phase, and the value does not go above the resources that can be found in the reserved area. A minimum usage rate is given to the pre-reserved resources.

$A_{rul}^{RI}(\omega) = A_{rult}^{res}(\omega), t = 1, \forall U \in UI, \forall_r \in RI, \forall l \in LI$---- (7)

$B_{uipt}^{RI}(\omega) = B_{uipkt}^{res}(\omega), t = 1, \forall U \in UI, \forall VI_i \in V, \forall_p \in CP, \forall k \in CI$---- (8)

The reservations assigned in the first step are not subject to cost uncertainty, and bookings are made similarly in all situations. Between the network and cloud provider capacity, bandwidth and VM values must reside. The overall amount of each resource used in the system must fall within each provider's prescribed parameters.

$\sum_{u \in UI} \sum_{l \in LI} (A_{rult}^{util}(\omega) + A_{rult}^{od}(\omega)) \leq t_r, \forall_r \in RI, \forall_t \in TS$---- (9)

$\sum_{v_i \in VI} D_i^p (\sum_{u \in UI} \sum_{k \in CI} (B_{uipkt}^{util}(\omega) + B_{uipkt}^{od}(\omega)) \leq t_p^p, \forall_p \in CP, \forall k \in CI$---- (10)

$\sum_{v_i \in VI} D_i^s (\sum_{u \in UI} \sum_{k \in CI} (B_{uipkt}^{util}(\omega) + B_{uipkt}^{od}(\omega)) \leq t_p^s, \forall_p \in CP, \forall k \in C$---- (11)

$\sum_{v_i \in VI} D_i^{in} (\sum_{u \in UI} \sum_{k \in CI} (B_{uipkt}^{util}(\omega) + B_{uipkt}^{od}(\omega)) \leq t_p^{in}, \forall_p \in CP, \forall k \in CI$---- (12)

To satisfy the emerging demand from the user side, the cloud providers' on-demand provisioning and resource usage of VMs from every class is enough.

$B_{uipkt}^{res} \in N_0, \forall U \in UI, \forall VI_i \in V, \forall_p \in CP, \forall k \in CI, \forall_t \in TSk$---- (13)

$B_{uipkt}^{util}, B_{uipkt}^{od} \in N_0, \forall U \in UI, \forall VI_i \in V, \forall_p \in CP, \forall k \in CI, \forall_t \in TS$---- (14)

$A_{uipkt}^{res} \geq 0, \forall U \in \forall U \in UI, \forall_r \in RI, \forall l \in LI, \forall ts \in TSl$---- (15)

$A_{uipkt}^{util}, A_{uipkt}^{od} \geq 0, \forall U \in \forall U \in UI, \forall_r \in RI, \forall l \in LI, \forall ts \in TS$---- (16)

The VM provision is used to acquire the non-negative integers, and the provisioning of router bandwidth is likewise non-negative. The network provides for the supply of enough bandwidth allocation, and the routing is economical. The restriction ensures a certain outcome, and the occurrence of traffic forces the cloud provider to fulfil the client's request while optimising demand. The stochastic approach is used to solve the single-stage issue, and a dynamic programme is added to the multistage approach tree to solve the problem.

The reconstruction of the tree is done in the context of tree creation, and every branch of the tree is connected to the root node and has an even number of levels. Ancestors are also removed from the tree when leaf nodes are removed. As with leaf nodes, antecedent nodes are removed from the upper levels of the tree, and their descendants are assigned to the neighbouring residual scenario as their descendants. By adding the chance of the detached scenario to the likelihood of the nearby residual scenario, probabilities in the scenario are estimated.The resulting tree maintains a strong semantic calculation of the whole scenario of the tree while being a simplified version of the tree that is quicker to traverse. Calculating the value function V (s), which represents the value of being in states, is the foundation of dynamic programming (DP). Bellman's value function is equivalent to the following:

$V(s) = \max_A \{RI(s,a) + \gamma(s,a) \sum_{s'} CF_{s,A}(s')V(s')\}$---- (17)

The stochastic dynamic programming method handles resource provisioning, and the creation of trees minimises space for the multistage issue as well.

4. Result and Discussion

The rapid evolution of mobile computing technologies has exhibited exponential growth, driven by advancements in hardware capabilities, high-speed wireless communication, and the proliferation of intelligent applications. Modern smartphones are no longer limited to basic communication tasks; instead, they support computationally intensive operations such as real-time video processing, augmented reality, image recognition, and natural language processing. These developments have significantly enhanced user experience and application diversity. However, despite improvements in processing power, memory, and connectivity, mobile devices remain inherently constrained by limited battery capacity and energy efficiency, which poses a critical challenge for sustained high-performance execution.

To address this limitation, computation offloading has emerged as an effective paradigm for energy optimization in mobile environments. Computation offloading involves transferring resource-intensive tasks from mobile devices to external resource-rich infrastructures such as cloud servers, edge nodes, or cloudlets. This paradigm not only reduces local computational burden but also minimizes energy consumption and execution latency when appropriately managed. In particular, Mobile Edge Computing (MEC) enables low-latency offloading by bringing computational resources closer to end users, thereby improving responsiveness and Quality of Experience (QoE).

In this study, the primary objective is to minimize energy consumption during continuous execution of computation-intensive applications by leveraging intelligent offloading strategies. A context-aware scheduling model is proposed, which dynamically determines optimal task placement based on parameters such as device state, workload characteristics, network conditions, and energy profiles. The proposed framework adopts a service-oriented architecture to facilitate adaptive and efficient task execution through an optimized task scheduling and offloading algorithm.

Furthermore, a proof-of-concept implementation is developed on an Android platform to validate the effectiveness of the proposed model. Experimental evaluation demonstrates that the framework significantly enhances energy efficiency while maintaining acceptable performance levels. The results indicate that context-aware offloading combined with intelligent scheduling can effectively mitigate rapid battery depletion, making it a promising solution for next-generation mobile computing environments.

Table 3: The scenario of under-provisioning and overprovisioning in SPDP

Cost in $	Oversubscription			On-Demand			Standard Deviation		
Overprovisioning Weight (ω^{+})	λ			λ			Λ		
	0	1	2	0	1	2	0	1	2
0	1000	1500	1550	2800	6500	5000	14000	12400	12300
0.5	800	1000	1400	1500	1400	1350	14900	14300	14200
1	700	950	1300	1400	1300	1300	14800	14300	14200
1.5	600	900	900	1300	1250	1200	14800	14300	14300
2	500	700	800	1200	1200	1100	14800	14800	14300

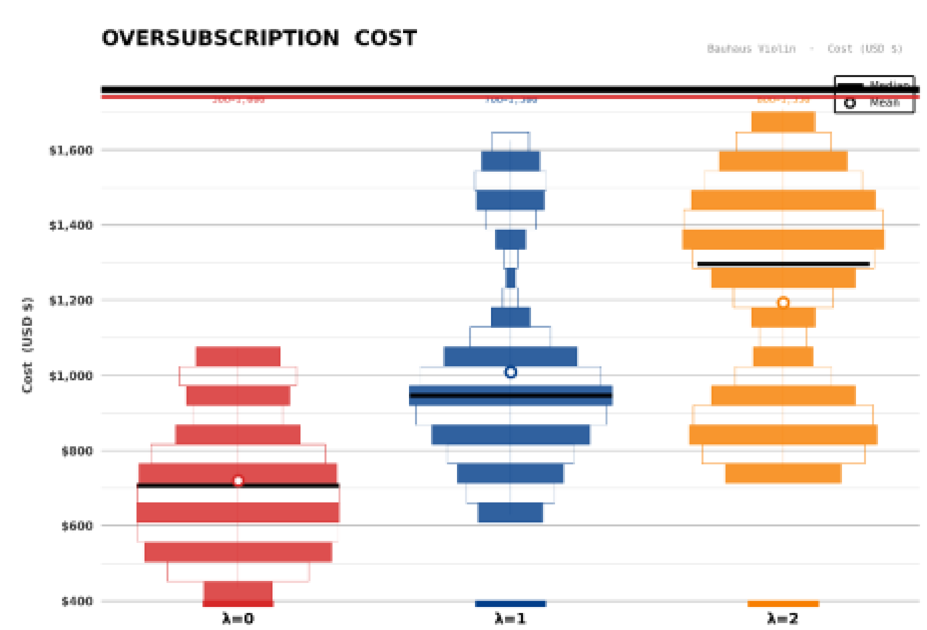

Figure 2: Oversubscribed Cost of SDPD

The violation of overprovisioning in the cost of oversubscription is plotted in Figure 2 and the cost is effectively reduced by applying SPDP. The comparison is carried for various overprovisioning weight.

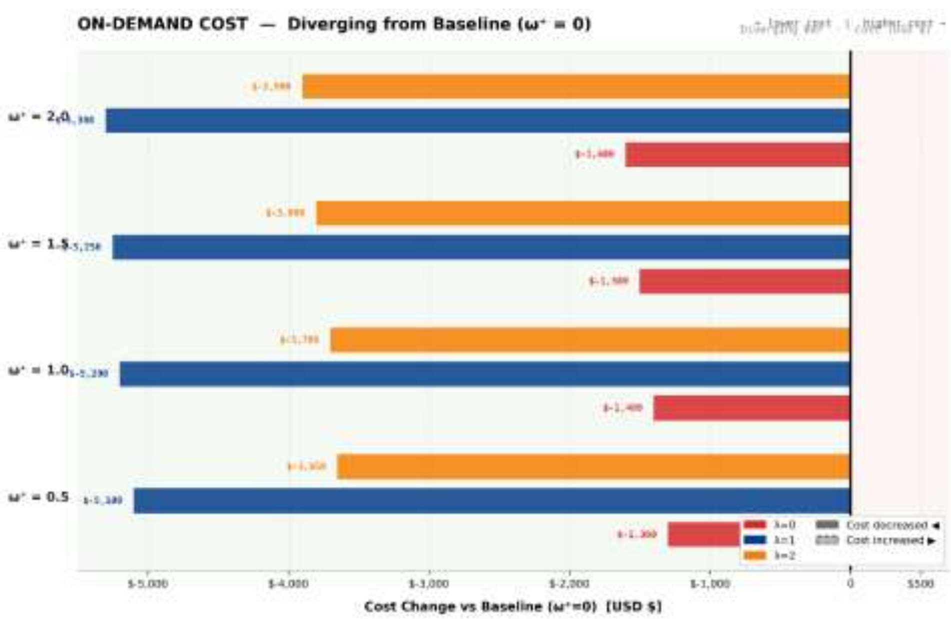

Figure 3: On-Demand Cost of SDPD

The violation of overprovisioning in the cost of On-Demand is plotted in Figure 3 and the cost is effectively reduced by applying SPDP. The comparison is carried for various overprovisioning weight.

Figure 4: Standard Deviation value of Provisioning Cost

The violation of overprovisioning in the standard deviation of cost is plotted in Figure 4 and the cost is effectively reduced by applying SPDP. The comparison is carried for various overprovisioning weight.

5. Conclusion

The SPDP (Stochastic Programming and Approximate Dynamic Programming) method exacerbates the cloud infrastructure's uncertainty problem. The suggested system takes into account variables including cost, heterogeneity, demand, and accessibility. By using SPDP, dynamic programming, which reduces space by using a tree, takes into account uncertainty in intrinsic probability space. The procedure of significantly increasing profitability and efficiently over- and under-provisioning is greatly enriched by the reduction in space. The SPDP model exemplifies the careful examination of prices that is advantageous to cloud service providers and customers. As compared to other accessible algorithms, the SPDP provides promising results, according to the results analysis.The suggested method is used in actual practise, such as a decision support system where decision plans are allocated for efficient provisioning between a user's private data centre and external cloud providers. To increase the accuracy of uncertainty, the variance in the minimization strategy will be used in the future.

Reference

1. W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646.
2. C. You, K. Huang, H. Chae, B.H. Kim, Energy-efficient resource allocation for mobile-edge computation offloading, IEEE Trans. Wireless Commun. 16 (3) (2017) 1397–1411.
3. C. Wang, C. Liang, F.R. Yu, Q. Chen, L. Tang, Computation offloading and resource allocation in wireless cellular networks with mobile edge computing, IEEE Trans. Wireless Commun. 16 (8) (2017) 4924–4938.
4. Changsheng You, Kaibin Huang, Hyukjin Chae, et al., Energy-efficient resource allocation for mobile-edge computation offloading, IEEE Trans. Wireless Commun. 16 (3) (2016) 1397–1411.

5. Wang, C., Liang, C., Yu, F. R., Chen, Q., & Tang, L. (2017). Computation offloading and resource allocation in wireless cellular networks with mobile edge computing. *IEEE Transactions on Wireless Communications*, *16*(8), 4924-4938.
6. Wang, R., & Liu, J. (2017). Adaptive formation control of quadrotor unmanned aerial vehicles with bounded control thrust. *Chinese Journal of Aeronautics*, *30*(2), 807-817.
7. Lu, H., Gu, C., Luo, F., Ding, W., & Liu, X. (2020). Optimization of lightweight task offloading strategy for mobile edge computing based on deep reinforcement learning. *Future Generation Computer Systems*, *102*, 847-861.
8. Huang, L. (2020). Energy harvesting sensor node scheduling in wireless sensor networks. *Encyclopedia of Wireless Networks*, 411-414.
9. Lin, K., Pankaj, S., & Wang, D. (2018). Task offloading and resource allocation for edge-of-things computing on smart healthcare systems. *Computers & Electrical Engineering*, *72*, 348-360.
10. Marc Lefebvre. (2026). Distributed Multi-Agent Control Framework for Autonomous Smart Grid Operation. *National Journal of Intelligent Power Systems and Technology*, 48-55.
11. C.Arun Prasath. (2025). Web-Based Knowledge Services for Distributed E-Learning Environments. *Journal of Information Resources and Knowledge Services*, 1–6. Retrieved from https://www.fsrap.com/index.php/JIRKS/article/view/230
12. Ashu Nayak. (2025). Nanotechnology-Enhanced Eco-Smart Building Materials for Carbon-Neutral and Climate-Resilient Urban Infrastructure. *Journal of Smart Infrastructure and Environmental Sustainability*, *2*(3), 1-8.
13. Shaik Sadulla, & K P Uvarajan. (2025). Environmental DNA and AI-Driven Drug Discovery: Unlocking Nature's Genomic Reservoirs for Novel Therapeutics. Frontiers in Life Sciences Research, 8–13.