A Hierarchical Control for Application Placement and Load Distribution in Edge Computing

Adyson M. Maia[a], Dario Vieira[b], Yacine Ghamri-Doudane[a], Christiano Rodrigues[b,c], Marciel B. Pereira[b,c], Miguel F. de Castro[c]

[a]*L3i Lab, La Rochelle University, 23 Av. Albert Einstein, La Rochelle, 17000, Charente-Maritime, France*
[b]*Efrei Research Lab, Université Paris-Panthéon-Assas, 32 avenue de la République, Villejuif, 94800, Val-de-Marne, France*
[c]*GREat Research Lab, Federal University of Ceará – UFC, Campus do Pici, Fortaleza, 60455-760, Ceará, Brazil*

Keywords: Application Placement, Load Distribution, Edge Computing, Distributed Control

1. Hyperparameter Tuning

In this section, we analyze the performance of our Cooperative Hierarchical Distributed Control (Coop-HDC) approach against different hyperparameter values in the scenario with 5 subsystems.

Fig. 1 presents the system performance of our Coop-HDC method by varying the number of generations of the Biased Random-Key with Non-dominated Sorting Genetic Algorithm (BRKNSGA). We can observe that migration cost is the main metric reduced by increasing the number of generations, and its value converges after 50 generations.

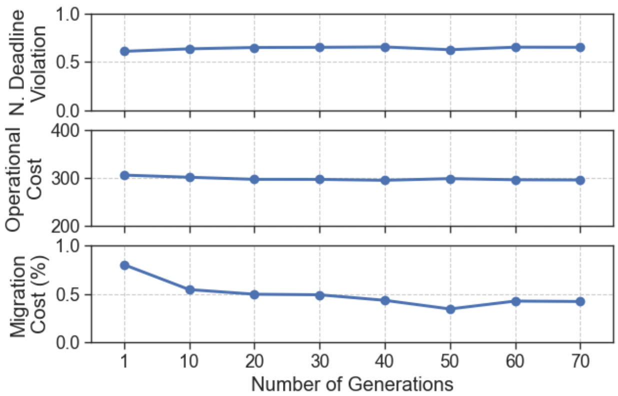

Figure 1: Performance of *Coop-HDC* $it_{\max} = 1$ under different number of generations of BRKNSGA

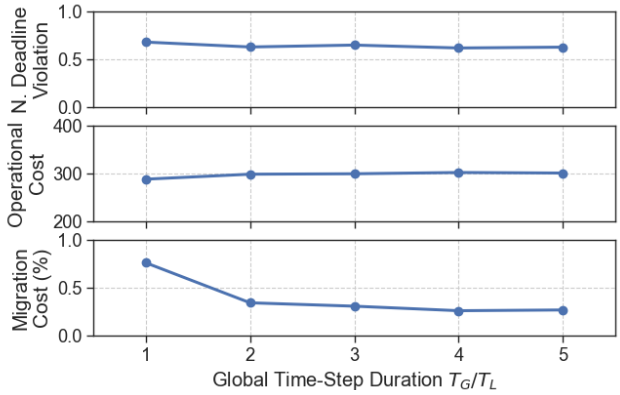

Figure 2: Performance of *Coop-HDC* $it_{\max} = 1$ by varying global time-step duration T_G

Fig. 2 presents the system performance of our Coop-HDC method by increasing the duration of the global time-step regarding the local time-step. That is, $T_G/T_L = 2$ means that the global time step is twice as long as the local time step. In this figure, we can detect that migration cost is the main metric affected by this hyperparameter. Thus, we selected $T_G/T_L = 2$ as the point with the most significant drop in migration cost.

Fig. 3 depicts the system performance of Coop-HDC by varying the length of the local prediction horizon. For $H = 1$, Coop-HDC has the lowest migration and operational costs but the higher deadline violation. Hence, we prioritized the reduction of deadline violation by choosing $H = 2$, which performed well in all three compared metrics.

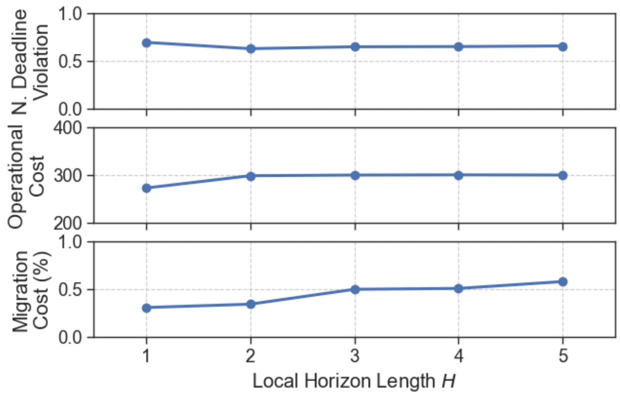

Figure 3: Performance of *Coop-HDC* $it_{\max} = 1$ by varying local horizon length H

Fig. 4 shows the performance of Coop-HDC by increasing the maximum number of $it_{\max}$ of Iterative Cooperation Strategy algorithm. We can note that migration cost augments by increasing the number of iterations. The deadline violation also increased slightly with more iterations. Thus, $it_{\max} = 1$ or $it_{\max} = 2$ have the better results.

2. Extra Evaluations

In Fig. 5, we analyze the average performance of our proposal over time. Coop-HDC presents a stable performance over time regarding deadline violation and operational cost but with more variations variation than the *Centralized* method. Regarding migration cost, Coop-HDC has more frequent changes but is

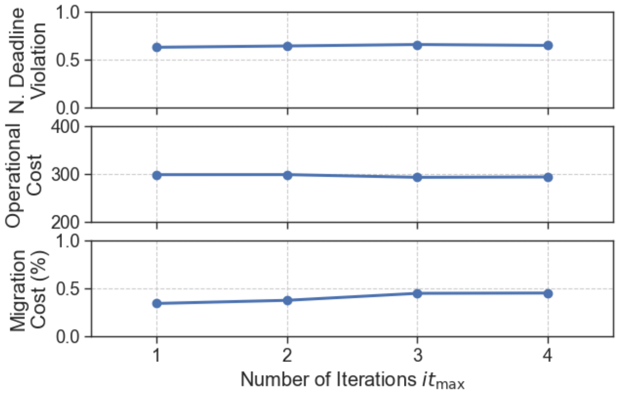

Figure 4: Performance of *Coop-HDC* under different number of iterations $it_{\max}$

still under control as it follows the results from the *Centralized* method.

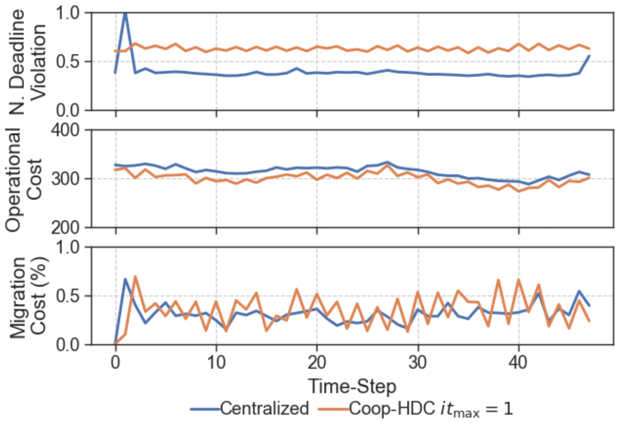

Figure 5: System performance over time.

3. Related Work

Table 1 presents a comparison of the discussed related work.

References

[1] O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, P. Leitner, Optimized iot service placement in the fog, Service Oriented Computing and Applications 11 (4) (2017) 427–443. doi:10.1007/s11761-017-0219-8.

[2] O. Ascigil, T. K. Phan, A. G. Tasiopoulos, V. Sourlas, I. Psaras, G. Pavlou, On uncoordinated service placement in edge-clouds, in: 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 2017, pp. 41–48. doi:10.1109/CloudCom.2017.46.

[3] L. Ferrucci, M. Mordacchini, M. Coppola, E. Carlini, H. Kavalionak, P. Dazzi, Latency preserving self-optimizing placement at the edge, in: Proceedings of the 1st Workshop on Flexible Resource and Application Management on the Edge, FRAME '21, Association for Computing Machinery, New York, NY, USA, 2020, p. 3–8. doi:10.1145/3452369.3463815.

[4] L. Chen, C. Shen, P. Zhou, J. Xu, Collaborative service placement for edge computing in dense small cell networks, IEEE Transactions on Mobile Computing 20 (2) (2021) 377–390. doi:10.1109/TMC.2019.2945956.

[5] H. R. Chi, R. Silva, D. Santos, J. Quevedo, D. Corujo, O. Abboud, A. Radwan, A. Hecker, R. L. Aguiar, Multi-criteria dynamic service migration for ultra-large-scale edge computing networks, IEEE Transactions on Industrial Informatics 19 (11) (2023) 11115–11127. doi:10.1109/TII.2023.3244321.

[6] E. Baccarelli, M. Scarpiniti, A. Momenzadeh, Ecomobifog–design and dynamic optimization of a 5g mobile-fog-cloud multi-tier ecosystem for the real-time distributed execution of stream applications, IEEE Access 7 (2019) 55565–55608. doi:10.1109/ACCESS.2019.2913564.

[7] M. Abbasi, E. Mohammadi-Pasand, M. Khosravi, Intelligent workload allocation in iot–fog–cloud architecture towards mobile edge computing, Computer Communications 169 (2021) 71–80. doi:https://doi.org/10.1016/j.comcom.2021.01.022.
URL https://www.sciencedirect.com/science/article/pii/S0140366421000438

[8] E. Baccarelli, M. Scarpiniti, A. Momenzadeh, S. S. Ahrabi, Learning-in-the-fog (lifo): Deep learning meets fog computing for the minimum-energy distributed early-exit of inference in delay-critical iot realms, IEEE Access 9 (2021) 25716–25757. doi:10.1109/ACCESS.2021.3058021.

Table 1: Related Work Comparison

Work	Architecture	Control Layer	Opt. Method	Coordination	Scalability	Dynamic Adaptation	Key Innovation
Skarlat et al. [1]	Hierarchical Edge	Local and global opt.	Heuristic-based Service Placement (HSP)	Local Coordination, Global Constraints (LCGC)	Scalable to large Internet of Things (IoT) deployments	Adaptable to varying load	Optimized IoT Service Placement (OISP)
Ascigil et al. [2]	Uncoordinated Edge-Clouds (UEC)	Independent Edge Computing (EC) nodes	Application Ranking Policies (ARP)	No coord.	Limited by lack of coordination	Static policies	Decentralized Service Placement (DSP)
Ferrucci et al. [3]	Autonomous Edge Nodes (AEN)	Autonomous EC nodes	Self-optimizing Placement (SOP)	Autonomous Decision-Making (ADM)	Scalable through Self-Optimization (SO)		
Chen et al. [4]	Dense Small Cell Networks (DSCN)	Collaborative Base Stations (BSs)	Parallel Gibbs Sampling (PGS)	Collaborative Placement Decisions (CPD)	Effective in dense deployments	Adaptive to Network Conditions (ANC)	Collaborative placement in DSCN
Chi et al. [5]	Ultra-Large-Scale Edge Networks (ULSEN)	Decentralized with central coord.	Multi-Criteria Decision Making (MCDM)	Central Guidance with Local Actions (CGLA)	Designed for ULSEN	Dynamic Service Migration (DSM)	MCDM
Baccarelli et al. [6]	Multi-tier Mobile-Fog-Cloud (MFC)	Virtualized Fog/Cloud Servers (VFS)	Gradient-based adaptive iterations, Genetic Algorithm (GA)	Centralized with Task Allocation and Scheduling (TAS)	Suitable for MFC ecosystems	Adaptive to application throughput and scheduling disciplines	Joint resource and task allocation
Abbasi et al. [7]	IoT-Fog-Cloud (IFC)	Edge Devices (ED), cloud	GA	Decentralized with Intelligent Allocation (IA)	Improved Quality of Service (QoS) with reduced power consumption	Addressing QoS and Security Limitations (SL)	Intelligent workload allocation with QoS enhancement
Baccarelli et al. [8]	Multi-tier Fog Nodes (FN)	Fog nodes with Conditional Deep Neural Networks (DNNs)	Primal-dual gradient-based iterations	Adaptive Resource Reconfiguration (ARR)	Self-detection and reaction to env. changes	Real-time adaptation to env. changes	Conditional DNNs with early exits for efficient inference
This Work	**Two-layer Hierarchical Control (HC)**	**Upper Global Controller (GC), lower Local Controllers (LCs)**	**GA, Limited Look-ahead Control (LLC)**	**Among LCs**	**Designed for large-scale EC systems**	**Proactive Control Strategy (PCS)**	**HC, PCS**