EMERGING TECH CONFERENCE – Edge Intelligence

Volume 02, 2023, Page 118 – 124

Proceedings of Emerging Tech Conference:
Edge Intelligence 2023

Intelligence Functions Placement in B5G / 6G wireless networks

Vasiliki Lamprousi[1] Sokratis Barmpounakis[1] Vera Stavroulaki[1] and Panagiotis Demestichas[1]
[1]*WINGS ICT Solutions, Greece*
vlabrousi@wings-ict-solutions.eu, sbarmpounakis@wings-ict-solutions.eu,
veras@wings-ict-solutions.eu, pdemest@wings-ict-solutions.eu

Abstract

The optimal computation, resource, and storage positioning for succeeding the best performance of complex systems, populated by multiple mobile devices with respect to mobile users and data protection, is of great interest in "Beyond the 5th Generation" (B5G) / 6th Generation (6G) networks in academia as well as in industry sectors. In this paper, an intelligence functions placement algorithm is proposed for optimally allocating the functionality to the various network/compute nodes as part of the intelligence distribution decision-making functional component of the smart connectivity platform envisioned in the H2020 project DEDICAT 6G. This algorithm can overcome a possible increase of network latency or a possible unavailability of used edge node and can be applicable to many use cases including the ones having robot edge nodes (warehouse environment). The theoretical estimations show that the proposed algorithm can significantly reduce the power consumption compared to baseline placement algorithms.

1 Introduction

The revolution of Industry 4.0 changed the way companies produce, enhance, and deliver their products by combining physical assets and advanced technologies such as Artificial Intelligence (AI), Internet of Things (IoT), robots, 3D printing, cloud computing, etc. The upcoming revolution of Industry 5.0 aims to achieve better industrial production by enabling effective collaboration and interaction among machines and humans (Adel, 2022). To support these technologies and applications, B5G/6G networks will offer ultra-fast, highly adaptive, and dependable platforms. These networks will also enable sophisticated management algorithms that will optimize various performance metrics, such as latency, energy efficiency, and resource allocation to improve the efficiency and resilience of advanced, industrial processes.

The H2020 project DEDICAT 6G (H2020-ICT-52, 2023) has the overall objective to transform B5G networks into a smart connectivity platform that is highly adaptive, ultra-fast, and dependable/resilient for supporting securely innovative, human-centric applications. Within this project a novel intelligence functions placement algorithm was developed for optimally allocating the functionality to the various network/compute nodes including robotic units (DEDICAT6G, 2021) and is presented in this paper.

The contribution of this paper is as follows: (i) This paper studies the intelligence functions placement problem in a smart warehousing environment where there are multiple robotic units, edge/cloud servers, towards energy efficiency and minimum communication cost (related to transmission delay) among

others. (ii) A novel metaheuristic algorithm is proposed for addressing this problem scalable also in large scale experimentations. (iii) The preliminary results show that the proposed algorithm can obtain an important reduction of the power consumption compared to baseline placement algorithms.

The rest of the paper is structured as follows. Section 2 presents the related work according to the state of the art in the research field. Section 3 provides the description of the intelligence functions placement problem as well as the proposed solution. Section 4 shows the results obtained till now and Section 5 concludes.

2 Related work

The problem of the optimal placement of services, tasks, computation, to the compute nodes of the system of interest has been widely researched in B5G/6G networks. In the literature these problems are solved with various optimization, metaheuristic, and Machine Learning (ML) algorithms. The authors of (L. Yala, 2018) use a genetic metaheuristic algorithm to solve an optimization problem that aims to place Virtual Network Functions (VNFs) for ultra-reliable and Low Latency Communications (uRLLC) services in a way that minimizes latency and maximizes service availability. Their algorithm can find near-optimal solutions faster than an exact algorithm of a Mixed Integer Programming (MIP) solver. The authors of (J. Kim, 2020) present a deep Q-network-based algorithm for placing Cloud-native networking functions (CNFs) on edge clouds in an efficient way, considering the cost of launching and operating CNFs, the back-haul control traffic overhead, and the number of served requests at each time. The simulation results showed that their algorithm can adapt to changes in service demand and reduce the cost per hour.

Moreover, energy efficiency is a key objective for 6G networks, hence there are many placement optimization algorithms proposed to achieve it. The (J. Li, 2023) considers the deployment of AI tasks on edge computing nodes that can access a cloud server, and jointly optimize the resource allocation, offloading decision, computing time, energy consumption and inference accuracy of each node, with the development of an algorithm which breaks down the problem into simpler subproblems based on the alternating direction multiplier method. Similarly, (A. Mesodiakaki, 2022) addresses the joint problem of user association, traffic routing and VNFs placement to maximize the energy efficiency and user acceptance ratio of the mobile network by introducing an energy-efficient real-time heuristic called ONE that uses online convex optimization techniques to solve the problem in a distributed manner. Furthermore, (Taneja A, 2022) proposes an energy-efficient algorithm for virtual machine placement optimization in cloud data centers that reduces the total power consumption and resource wastage by employing a heuristic resource usage factor model with reward and penalty mechanisms. However, there are limited studies that consider additionally the challenges of robot-based compute node and robot related functionality which our study includes.

3 Problem description / Solution approach

A set of n Functional Entities (FEs) e.g., tasks, jobs, services, is assumed for approaching the intelligence functions placement problem, with specific computation and functional requirements. The possible communications/dependencies between FEs are represented by a functional graph (Directed Acyclic Graph-DAG), where each node corresponds to a FE and each edge connects interacting FEs and it is weighted according to the amount of data transferred between FEs. Additionally, each edge has a maximum acceptable transmission delay (threshold).

Also, a set of m Hosting Entities (HEs) e.g., edge nodes, core nodes, robotic units, end user devices, is assumed, having some capabilities. These are the maximum available CPU, and memory resources, the battery level if applicable and the set of FEs that each HE can support (e.g., a robotic unit can support object recognition if camera is available but cannot support grasping an object if robotic arm is not available). Also, a system layout graph is considered, consisting of the available HEs and the communicational channels among them with varied capacity link each.

The aim is to allocate the FEs to HEs by ensuring efficiency of the system with low energy consumption and latency. Therefore, the objective is to find the optimal cost allocation that meets the performance requirements. The Objective Function (OF) which is minimized $OF = w_1 B + w_2 P + w_3 D$, consists of the term $B = \sum_{j=1}^{m} y_j b_j$ which denotes the cost related to the battery level of utilized HEs (takes higher values when battery is low, and close to zero values when it is fully charged or when the HE is not battery-powered). y_j is a binary decision variable that indicates if HE j is utilized or not and b_j is a cost related to the battery level of HE j . The $P = \sum_{j=1}^{m} \left(W_{max}^{j} - W_{idle}^{j}\right).\ Ucpu_j(x_{i,j}) + W_{idle}^{j}$ term denotes the power consumption cost which is modelled based on (Alharbe, Aljohani, & Rakrouki, 2022), where $Ucpu_j(x_{i,j})$ is the CPU utilization rate on the HE j in terms of the decision variable $x_{i,j}$ which indicates if FE i is assigned on HE j. $W_{max}^{j} - W_{idle}^{j}$ are the power consumption when the HE j is fully loaded and idle, respectively. Finally, $D = \sum_{j=1}^{m} \sum_{j\prime=1}^{m} \sum_{i=1}^{n} \sum_{i\prime=1, i\prime\neq i}^{n} z_{i,i\prime,j,j\prime} \frac{k_{i,i\prime}}{cap_{j,j\prime}}$ denotes the cost (transmission delay) imposed by the communication among HEs related to the amount of data transferred ($k_{i,i'}$) between FEs and the maximum capacity link ($cap_{j,j'}$). The binary decision variable $z_{i,i',j,j'}$ shows if HEs j and j' are communicating due to communicating FEs assigned on them. Each term of the OF is normalized and is weighted (w_1, w_2, w_3) depending on the use case.

The constraints of the problem are that each FE can be allocated to only one HE ($\sum_{j=1}^{m} x_{i,j} = 1, \forall\ i = \{1, \ldots n\}$). The available resources (e.g., CPU, memory) of the HEs should be respected ($\sum_{i=1}^{n} x_{i,j} * cpuFE_i \leq cpuFE_j$ and $\sum_{i=1}^{n} [x_{i,j} * memFE_i \leq memHE_j\ \forall\ j = \{1, \ldots m\}$). The functional requirements of the FEs (e.g., camera, wheels) should be respected ($x_{i,j} \leq t_{i,j}, \forall\ i = \{1, \ldots, n\}, j = \{1, \ldots m\}$ where $t_{i,j}$ ≤ is 1 if FE i can be assigned on HE j in terms of functionality types that can be supported by the HE and 0 otherwise). Also, the maximum transmission delay between two interacting FEs should be respected $z_{i,i\prime,j,j\prime} \cdot \frac{k_{i,i\prime}}{cap_{j,j\prime}} \leq l_{i,i\prime}, \forall\ i, i' \in \{1, \ldots, n\}, j, j' \in 1, \ldots, m,$ where $i \neq i', j \neq j',$ where $l_{i,i\prime}$ is the maximum acceptable transmission delay between FEs i and i').

An example of the intelligence functions placement algorithm utilization is shown in Figure 1. Two robots with different functionalities and one server are depicted having some FEs assigned on them. An increased network latency is detected, and the algorithm is triggered. The output of the algorithm is the reallocation of the FEs to the available HEs to overcome this issue.

The intelligence functions placement problem was solved with the development of a metaheuristic genetic algorithm based on the genetic algorithm paradigm. The algorithm initializes a population of *chromosomes*, possible solutions, each consisting of a series of HEs representing the "proposed" HE for each FE (to be placed). Then, the fitness/objective function (OF) is applied to each chromosome obtaining a fitness score each. Over the course of a number of generations defined by the dynamic stopping criterion utilized (for succeeding convergence in a satisfying execution time), a population of chromosomes evolves, and operators like parent selection, crossover, and mutation improve the population's overall fitness. In this metaheuristic tournament selection, one-point crossover and reserve sequence mutation proposed

in (Otman Abdoun, 2012) was utilized among others. Crossover and mutation operators occur with a predefined probability. The crossover and mutation rate (predefined occurrence probability) utilized was 0.8 and 0.15, respectively.

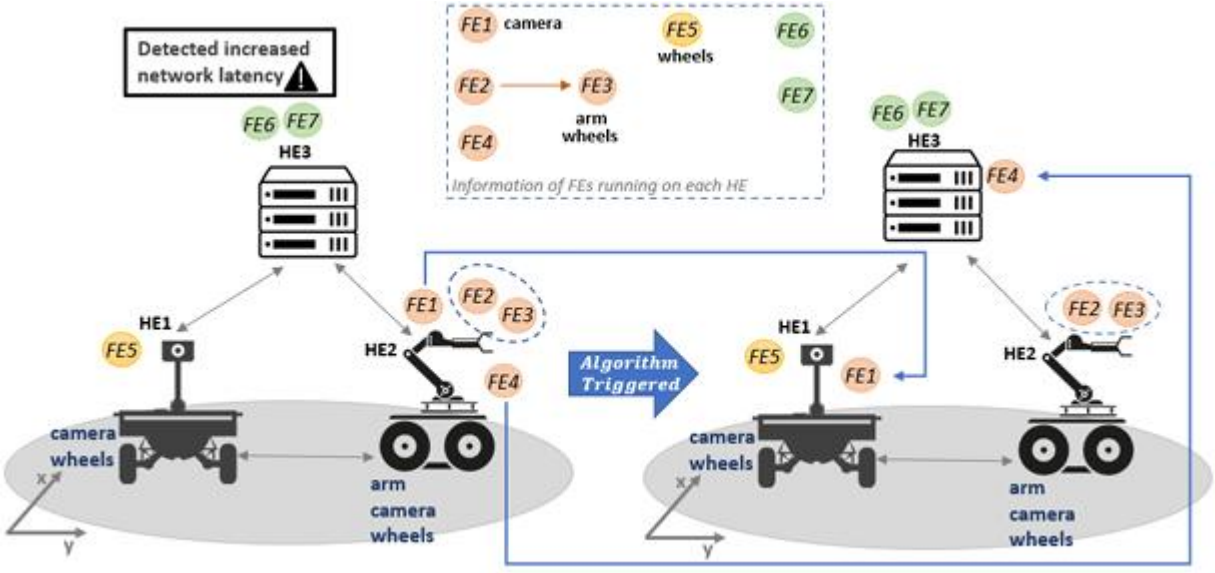

Figure 1: Schematical representation of a case of intelligence functions placement algorithm utilization.

4 Results

A performance testing of the proposed model was performed comparing to a MIP Python solver output

	score		execution time (s)	
# FEs	MIP solver	Proposed algorithm	MIP solver	Proposed algorithm
2	0.1638	0.1638	0.011	0.551
4	0.1697	0.1718	0.030	0.684
8	0.1770	0.1939	1.316	1.371
12	0.1821	0.1822	200	1.494
16	0.1972	0.2081	200	4.675
20	0.2040	0.2278	200	6.128
24	0.2264	0.2457	200	14.675
28	0.2553	0.2935	200	16.624
32	0.2769	0.3101	200	22.307
36	0.3146	0.3317	200	23.730
40	-	0.3517	-	32.850
44	-	0.3787	-	49.427

Table 1: Performance testing (score and execution time) of the proposed intelligence functions placement algorithm compared with the output of the MIP solver with fixed HE-schema and increasing number of FEs.

(Mitchell S, 2011) by measuring the scores (the final minimum obtained value of the objective function OF) and the execution time having a fixed HE-schema of 43 HEs and increasing number of FEs (see Table 1). The levels of the available CPU of the HEs are assumed {2000, 2600, 3000} MIPs, the levels of the available memory are {2048, 4096, 8192} MB, the levels of power consumption when fully loaded and when idle are {260, 360, 460} W and {70, 100, 170} W respectively. The links between HEs have capacity 3.3 − 20 Mbps. The FEs have {500, 750, 1500} MIPS levels of required CPU, {256, 512, 2048} MB levels of required memory, 2-10 MB levels of data transferred and maximum available delay between interacting FEs, 0.2-1.4 s. An execution time limit of 200s was imposed on the MIP Python solver because it was computationally intractable from over 12 FEs. The results showed that the proposed metaheuristic algorithm had close to optimum scores within significantly less time than MIP model, and when exceeding 36 FEs, MIP solver could not find feasible solution within 200s.

Moreover, Figure 2 shows the theoretical measurements of the percentage reduction of power consumption when using the proposed metaheuristic intelligence functions placement algorithm and when using the random feasible placement, which respects the systems constraints, and the round robin placement as baselines. For these measurements the same fixed HE-schema was assumed of 43 HEs as the one assumed for the performance testing. The mean value of ten measurements was obtained in each case. The power consumption was calculated by the P term of the OF described in Section 3. As it is observed in Figure 2, as the number of FEs increases, the power consumption gains increase, until a critical point (25 FEs) and then decreases. The proposed algorithm provides higher power consumption gains when there is sufficient availability of computational resources on the HEs, thus solution space is wider and can easier reach an optimum state. From these estimations is shown that our algorithm can reach up to a 35%-44% reduction of power consumption compared to random feasible placement or round-robin placement, respectively.

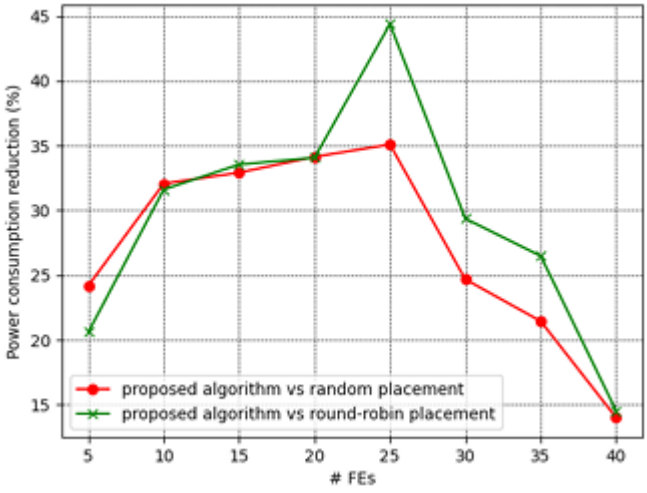

Figure 2. Percentage reduction of power consumption when the intelligence functions placement algorithm is used compared to a random feasible placement and round-robin placement (baselines).

The above scenarios and results were reactive approaches of functions placement. A proactive approach which could predict the future states of the FEs and resources, could identify the possible increase of network latency or possible unavailability of used HE. The prediction of the behavior of services or components can increase efficiency and reduce the operational and maintenance costs of the system by

scheduling any needed placement in advance. The discussed prediction of upcoming critical situations can be succeeded by monitoring data from selected services or components, and train AI/ML models for time series forecasting. The output future state of the system can be used by the developed intelligence functions placement mechanism to determine whether pre-emptive actions are necessary to prevent upcoming critical events. This work is ongoing and will soon generate important results.

5 Conclusion

In this paper, the intelligence functions placement problem for distributing the intelligence/functionality to the various network nodes (edge, core nodes, robotic units etc.) is studied, taking into consideration the transmission delay and power consumption among others. The description of the problem is provided along with the description of the MIP and genetic algorithm implemented for approaching this problem. Additionally, some main results are provided with a discussion on the proactive and dynamical placement of the intelligence functions.

6 Acknowledgement

This work was supported by the European Union H2020 Project DEDICAT 6G under grant no. 101016499. The contents of this publication are the sole responsibility of the authors and do not in any way reflect the views of the EU.

7 References

[1] Abdoun O, Jaafar A, Chakir T (2012). Analyzing the performance of mutation operators to solve the travelling salesman problem. Neural Evolut Comput Int J Emerg Sci 2(1):61–77

[2] Adel A (2022). Future of industry 5.0 in society: human-centric solutions, challenges and prospective research areas. J Cloud Comput; 11:40. https://doi.org/10.1186/s13677-022-00314-5.

[3] L. Yala, P. A. Frangoudis, and A. Ksentini (Dec 2018). Latency and Availability Driven VNF Placement in a MEC-NFV Environment, in 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1–7

[4] J. Kim, J. Lee, T. Kim, and S. Pack (2020). Deep Reinforcement Learning based Cloud-native Network Function Placement in Private 5G Networks, in 2020 IEEE Globecom Workshops (GC Wkshps), Taipei, Taiwan, pp. 1–6

[5] J. Li, F. Lin, L. Yang and D. Huang, (2023 May-June). AI Service Placement for Multi-Access Edge Intelligence Systems in 6G, in IEEE Transactions on Network Science and Engineering, vol. 10, no. 3, pp. 1405-1416, doi: 10.1109/TNSE.2022.3228815

[6] A. Mesodiakaki, M. Gatzianas, G. Kalfas, C. Vagionas, R. Maximidis and N. Pleros, (2022). ONE: Online Energy-efficient User Association, VNF Placement and Traffic Routing in 6G HetNets, 2022 IEEE Globecom Workshops (GC Wkshps), Rio de Janeiro, Brazil, pp. 304-309, doi: 10.1109/GCWkshps56602.2022.10008742

[7] Taneja A, Saluja N, Taneja N, Alqahtani A, Elmagzoub MA, Shaikh A, Koundal D. (2022). Power Optimization Model for Energy Sustainability in 6G Wireless Networks. Sustainability. 14(12):7310. https://doi.org/10.3390/su14127310

[8] Alharbe, N.; Aljohani, A.; Rakrouki, M.A. (2022). A Fuzzy Grouping Genetic Algorithm for Solving a Real-World Virtual Machine Placement Problem in a Healthcare-Cloud. Algorithms, 15, 128. https://doi.org/10.3390/a15040128

[9] Mitchell S, O'Sullivan M, Dunning (2011). Pulp: A linear programming toolkit for python. Accessed May 1, 2013, https://code.google.com/p/pulp-or/

[10] DEDICAT 6G D3.1 First Release of Mechanisms for Dynamic Distribution of Intelligence, Deliverable D3.1, 2021

[11] H2020-ICT-52 project DEDICAT 6G: Dynamic coverage Extension and Distributed Intelligence for human Centric Applications with assured security, privacy, and Trust: from 5G to 6G, Project website available at https://dedicat6g.eu/. Accessed July 11, 2023