Federated Scheduling of Fog-native Applications over Multi-Domain Edge-to-Cloud Ecosystem

Mays AL-Naday
University of Essex
Colchester, UK
mfhaln@essex.ac.uk

Vasileios Karagiannis
Austrian Institute of Technology
Vienna, Austria
vasileios.karagiannis@ait.ac.at

Tom De Block
AIOTI
Brussels, Belgium
Tom@viniblock.com

Bruno Volckaert
Ghent University - imec
Gent, Belgium
bruno.volckaert@UGent.be

***Abstract*—Fog computing is emerging as geo-distributed and connected edge-to-cloud ecosystems, spanning multiple domains operated by different entities. Consequently, fog-compatible applications need to support distributed operations and decentralized management. This promoted the adoption of the microservices architecture, to facilitate application modularity and autonomy. Transitioning to fog-native applications, i.e., running distributed microservice workflows over multiple domains, is a challenging endeavor. On one hand, distributing workflows require awareness of the intents and dependencies of microservices, as this may impact the supply of data and the perceived Quality of Service (QoS). On the other hand, the variant capacities and energy supply, coupled with limited information-sharing across fog autonomies, hinders the prospect of end-to-end optimization. To tackle such problems, we propose a novel federate optimisation algorithm for multi-domain scheduling of fog-native microservice workflows. The algorithm incorporates workflow intents in decision-making by combining Bender's decomposition with Alternating Direction Method of Multipliers (ADMM) to provide optimized workflow placement, mapping, routing and admission. The performance of the algorithm is evaluated analytically and compared to state-of-the-art intent-based ADMM (iADMM). The results show performance trade-offs with the proposed iBADMM (direct), with the latter improving the fraction of workflow greenness by $\approx 15\%$.**

***Index Terms*—fog-computing, fog-native, microservices, workflow placement, optimisation**

I. Introduction

The proliferation of cloud services has prompted the cloud-native paradigm, referring to the development of applications for intrinsic operation on distributed cloud resources. This has demanded greater modularity, and in turn incentivised adoption of the microservices architecture in developing applications. So far, cloud-native applications have run within the boundaries of central clouds. Meanwhile, edge-to-cloud autonomous and connected systems are emerging as a geo-dispersed and decentralised ecosystem, spanning multiple domains of different actors in the field [1]–[3]. The coexistence of multiple autonomies coupled with variant constraints on their resources increases the likelihood of application *dispersion*. That is to distribute application's components over different fog clusters and interconnect them to compose the application. Provisioning dispersed, yet intent-compliant, applications with state-of-the-art solutions is not straightforward.

Specifically, variation of actors' intents, infrastructure capacities, and energy supply introduce non-trivial trade-offs. Actors' intents may sometimes conflict. For instance data-sharing intents of end-users may conflict with operational intents of cloud providers. Added to that, capacity constraints increase the likelihood of congestion, which ultimately impact the perceived QoS. The structure of energy supply in terms of cost and greenness may vary the operational costs and introduce a trade-off with environmental intents of any of the actors. Current indicators [4], [5], show that smaller data centers have higher energy prices and worse power usage efficiency (PUE). This not only translates into higher operational costs, but further means poorer utilisation of green energy. Notably, the variation in supply of green energy to data centers remains unclear. Current indicators show the purchase of green energy to correlate with the size of cloud providers, with ultra-large providers having the biggest share [6], [7].

The above creates different attraction forces, which impact compliance with actors' intents while meeting QoS thresholds and optimizing system operations. For geo-distributed applications, such forces can interplay with each other and may create a ripple effect that impacts the end-to-end application performance. Furthermore, to deliver "need-to-run-close-to-user" applications, there is a need for efficient utilization of the sparse edge. This includes conserving small, edge clusters for such applications while pushing counterparts to cheaper clusters, within QoS bounds. Hence, to facilitate an intent-based application scheduling, taking into account the limited sharing of information across autonomies, there is a need for an optimisation solution that: 1) has knowledge of actors' intents; and 2) can act on the underlying heterogeneity given limited information sharing among actors.

This work proposes a novel federated optimization algorithm for scheduling geo-distributed fog-native applications, referred to as *microservice workflows*, in a multi-domain edge-to-cloud ecosystem. For simplicity, we refer to this ecosystem as the "fog ecosystem". The proposed algorithm jointly solves the problem of workflow placement, mapping, routing and admission. It does so by combining Bender's decomposition [8] with ADMM [9], to decompose the problem and distribute the solving tasks to the corresponding actors. Consensus is reached through iteration that reduces the gap between partial solutions to an acceptable level. The algorithm is compared with our earlier state-of-the-art algorithm

Mays AL-Naday, Vasileios Karagiannis, Tom De Block, Bruno Volckaert, "Federated Scheduling of Fog-native Applications over Multi-Domain Edge-to-Cloud Ecosystem", 2023 International Conference on Network and Service Management (CNSM), IEEE, 2023, DOI:

iADMM from [10]. The proposed algorithm is referred to as intent-based Bender's decomposition with ADMM with direct paths or iBADMM (direct) as the ADMM component differs from [10] by incorporating end-user intents in data upload, as well as solving the routing problem for direct cluster-to-cluster paths. Accordingly, the contributions of this work are three-fold: 1) model the multi-domain ecosystem describing different actors with examples of their intents; 2) formulate the problem of intent-based workflow placement, mapping, routing and admission; and 3) propose the federated optimization algorithm iBADMM (direct) to jointly solve the problem and evaluate its performance, compared to state-of-the-art.

The remainder of this paper is structured as follows: Section II reviews state-of-the-art related work, while Section III describes the multi-domain ecosystem. Section IV models the ecosystem, including interactions among actors, and formulates the problem of intent-based workflow placement, mapping, routing and admission. Section V describes the proposed iBADMM (direct) algorithm, while Section VI evaluates the performance of the algorithm. Finally, Section VII draws the conclusions and proposes future work.

II. Related work

Edge-to-cloud ecosystems have been studied extensively with sample summarizing efforts that include [1], [2]. Several studies focus on the orchestration of scientific workflows in fog computing The work of [11] proposes a solution that combines Discrete Moth-Flame Optimization with Differential Evolution (DMFO-DE) to reduce energy consumption for task execution in scientific workflows. Other approaches to workflow scheduling include: an agent-based approach optimized through a genetic algorithm [12], and swarm-based methods such as the hybrid fireworks algorithm of [13]. While these efforts address relevant challenges, they assume global knowledge of the state, limiting their applicability to centrally managed ecosystems.

The work of [14] proposes an ADMM-based distributed algorithm that decomposes the service mapping and routing problems to node-level sub-problems. Although the solution assumes global knowledge of actors; it provides a baseline to newer approaches. The work of [15] proposed a Bender decomposition approach for solving the problem of cloudlet placement and task allocation. Their approach allows for decoupling the placement problem from the task allocation counterpart. However, they still solve each problem globally. The work of [16] proposes a distributed algorithm for solving the problem of graph embedding in Service Function Chaining (SFC), based on a combination of Bender decomposition and ADMM. While their solution lacks consideration of latency and has different cost factors, it has strong potential in enabling multi-domain workflow orchestration.

Orthogonally, increasing efforts are focusing on the role of green energy in operating cloud infrastructure and services. The work of [17], reviews studies of energy usage in the cloud, in combination with an overview of use of green energy in fog computing for 6G enabled IoT. The work of [18] introduces a Mixed Integer Linear Programming (MILP) model to reduce the carbon footprint in cloud architectures. Finally, our earlier work in [10] proposes iADMM, a decentralised algorithm for intent-based mapping and admission based on ADMM. iADMM solves parts of the problem tackled in this work, while tolerating path routing sub-optimality and ignoring the intents of end-users. Overall, the reviewed efforts offer different perspectives of confronting the addressed problem. Nevertheless, while their contributions provide a strong foundation for the work at hand, they fall short in either enabling autonomous decision-making or incorporating intent awareness of different actors.

III. A multi-domain ecosystem

A. The ecosystem, actors and information exposure

A metropolitan fog ecosystem is illustrated in Figure 1. A metropolitan fog ecosystem (illustrated in Figure 1) may follow the OpenFog reference architecture of [3] and similar definitions [19]. It is comprised in the minimum of: access gateways (nodes), connecting end-users to the ecosystem; a set of geo-distributed and autonomous edge-cloud data centres and/or clusters (*fog nodes*) of variant capacities; application(s) offered as services over said clouds; and a programmable network that connects all aforementioned parts. Within such an ecosystem, multiple actors of different intents may co-exist and interact: application provider(s), clouds providers(s), network operator(s) and end-user(s). State information sharing is foreseen to be limited by virtue of competitiveness across the different actors in the ecosystem. Edge-cloud clusters and network elements are managed by their respective Management and Orchestration systems (MANOs).

Application (app) providers are assumed to deploy workflows over one or multiple fog nodes, with each microservice having at least one Point of Presence (mPoP). App providers have limited knowledge of the ecosystem. They do not know of the capacities of fog nodes, nor the connectivity and distance to end-users. To optimize deployment decisions, app providers are advised of the cost of microservice deployment and scaling on each fog node. Further, limited, information may be provided to assist in meeting the intents of app providers. We consider *workflow greenness* - i.e. the fraction of workload per workflow served by green energy - as one of such intents. Hence, cloud providers inform their app counterparts of the fraction of green energy supply per node.

Similarly, cloud providers are aware of limited network information including: cost of sending data traffic; end-to-end network latency on any path between a pair of fog nodes or between an access gateway and a fog node; and some of the intents of app providers such as workflow resource requirements and QoS thresholds. The network operator is assumed here to be responsible for access gateways, in addition to the programmable network. The network operator knows the cost of processing users' requests in any of the fog nodes along with the processing latency, but they do not know the infrastructure capacities of fog nodes.

TABLE I
SUMMARY OF NOTATIONS

Notation	Definition
$\mathcal{N}$	Edge-to-cloud clusters (i.e. fog nodes)
$\mathcal{U}, \mathcal{P}$	Access gateways (nodes), network paths
$\mathcal{W}$	workflow graph
$\mathcal{S}^w, \mathcal{R}^w$	microservices, dependencies in $\mathcal{W}$
$\mathcal{M}_n$	other cluster (fog nodes) than n
c_n, μ_n, e_n, η_n	CPU capacity & speed, energy cost & green fraction
b_p, l_p, h_p	bandwidth, length & hop count of path p
c^s, q^s, r^s, τ^w	task size, input, output data size, response time
$\mathcal{F}$	intent utility function
$\phi^s_n, \theta^s_n, \theta^s_{un}, \theta^s_{nm}$	costs: placement, mapping, upload, routing
δ, ω	workload & traffic demand
α, γ, β	variables: placement, mapping, routing/admission
ζ, ρ, ϵ	Lagrangian multiplier, penalty parameter, error gaps
ψ, Υ	routing dual variable, linear problem cost

IV. THE ECOSYSTEM MODEL

a) ***Edge-to-cloud nodes****:* are modeled as a set of fog nodes, $\mathcal{N}$. Each $n \in \mathcal{N}$ is connected to other nodes $\mathcal{M}_n = \mathcal{N} \backslash n$ by a set of paths $\mathcal{P}_{\mathcal{M}_n}$. nodes are small-sized at the edge (nano clusters), and expands when moving nearer to the network core. For simplicity without loss of generality, we assume each data centre to be comprised of a single cluster. Hence, paths between clusters traverse outside data centre boundaries over external network infrastructure. Similar to [10], each $n \in \mathcal{N}$ is defined by the tupple $\langle c_n, \mu_n, e_n, \eta_n \rangle$, where: c_n is the computing capacity, in number of CPU cores; μ_n the average CPU speed, in GHz; e_n the energy price at the node site, in Penny per milliCPU (mCPU); and η_n the ratio of green energy supply. The cloud provider manging the node is aware of all tupple information. Whilst, application providers and the network operator may only be aware of η_n along with the processing latency and cost per microservice.

b) ***Access Gateways and Network Paths****:* gateways are modeled as a set of nodes, $\mathcal{U}$. Each $u \in \mathcal{U}$, from one side, connects a group of end-users. From the other side, u connects to $\mathcal{N}$ fog nodes. An access gateway acts on the intents of the network operator and end-users, when sharing user-generated data and/or accessing fog services. Access and fog nodes are connected by a set of network paths $\mathcal{P} = \{\mathcal{P}_{\mathcal{U}}, \mathcal{P}_{\mathcal{N}}\}$, where $\mathcal{P}_{\mathcal{U}}$ is the set of access-cloud paths and $\mathcal{P}_{\mathcal{N}}$ is the set of inter-cluster (i.e. inter-fog-nodes) counterparts. Each $p \in \mathcal{P}$ is defined by the tupple $\langle b_p, l_p, h_p \rangle$, where: b_p is the bandwidth capacity of p, in Gbps; l_p is the metric distance of p, in km; and h_p is the hop count. While the operator knows all path metrics, cloud providers may only know the unitary cost of traffic and end-to-end network latency.

c) ***A microservice workflow****:* is modeled as a service graph $\mathcal{W}(\mathcal{S}^w, \mathcal{R}^w)$, whereby $\mathcal{S}^w$ is the set of microservices in the workflow and $\mathcal{R}^w$ is the set of directional inter-microservice dependencies. A dependency $r_{s,t}$ between (s, t) microservices indicates t reliance on s. A gateway u directs users' requests to the first (ingress) microservice in $\mathcal{W}$, while expects a response back from the last (egress) microservice in the workflow. u may further supply user-generated data as input to any $s \in \mathcal{S}^w$. App providers may offer multiple workflows in the ecosystem, with a set of intents, specified at microservice and/or workflow levels or as a general policy.

d) ***Intents****:* are modeled as utility functions. Each actor in the ecosystem defines a set of intents, which may (or not) conflict with those of their peers. We consider *workflow greenness* as a business intent for app providers as well as other actors. We define $\mathcal{F}(\eta_n, \mathcal{W})$ as the utility function corresponding to workflow greenness, given the fraction of green energy supply to n. App providers may further define non-functional intents, such as resource requirements and QoS thresholds. Similar to [10], we assume resource requirements are defined per s by $\langle c^s, q^s, r^s \rangle$, whereby: c^s mCPU is the average task size; q^s Mb/s is user-generated input data; and r^s Mb/s is the average size of the result of running s jobs. QoS is defined by $\langle \tau^w \rangle$, the response time tolerance threshold per workflow. End-user intents are represented by their access gateway. Such intents may reflect data privacy, the cost of data upload and risk to QoS due to network variant conditions. Here, we focus on cost of data sharing and risk to QoS, assuming traffic from an access gateway to a fog node incurs network cost, borne by end-users.

Cloud providers and the network operator may define operational intents, reflected in the cost of scaling a workflow deployment. Three costs are incurred in such an ecosystem: the *placement* cost of a microservice s at a fog node n, ϕ^s_n; the *admission* cost of requests for s at fog node n, θ^s_n; and, the *transmission* cost of data traffic. The latter is modeled as two-part cost of: uploading user-generated data on access-fog paths $\mathcal{P}_{\mathcal{U}}$, θ^s_{un}; and routing the output of s to its dependents $\mathcal{S}^{w,s}$ on inter-cluster paths $\mathcal{P}_{\mathcal{N}}$, θ^s_{nm}. More details on the cost and utility functions are provided on GitHub [1].

e) ***Workload, traffic and decision variables****:* Each $u \in \mathcal{U}$ receives requests for $\mathcal{W}$ at a rate of λ^w_u. Note that the request rate for $\mathcal{W}$ is equivalent to the request rate for each $s \in \mathcal{S}^w$, hence $\lambda^w_u \equiv \lambda^s_u$. λ^s_n is the aggregate request rate received by n, for all $\mathcal{U}$ and λ^s_{nm} is the fraction of s requests served by n and the response is directed to m. Accordingly, workload per s can be defined as:

$$\delta^s_u = \lambda^s_u c^s, \ \delta^s = \sum_{u \in \mathcal{U}} \delta^s_u \tag{1}$$

whereas upload and inter-cluster traffic per s can be formulated as:

$$\omega^{s,q}_u = \lambda^s_u q^s, \ \omega^{s,r}_n = \sum_{m \in \mathcal{N}} \lambda^s_{nm} r^s, \forall n, m \in \mathcal{N} \tag{2}$$

To deploy a workflow in the fog ecosystem, optimal decisions (illustrated in Figure 1) need to be made on: 1) the ***placement*** of each $s \in \mathcal{S}^w$ in one or multiple fog nodes; 2) the ***mapping*** of requests and user data $\forall s \in \mathcal{S}^w$ from u to n; 3) ***routing*** output of s to its dependents, hosted on the same or peer fog nodes; and accordingly, 4) ***admission*** of workload for s at n. The workflow orchestrator makes the placement decision centrally, while access nodes make the mapping decision in a decentralised fashion. Similarly, cloud

[1] https://github.com/mfhaln/iBADMM

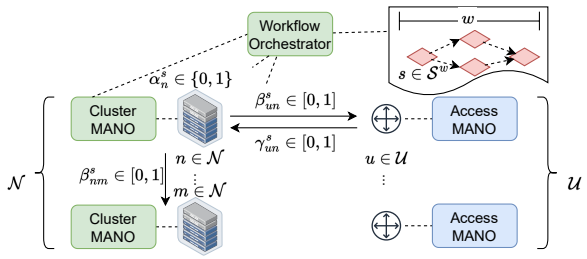


Fig. 1. Decision variables in the ecosystem

MANOs decide on admission and inter-cluster routing in a decentralised manner. The placement decision is defined as a binary variable:

$$\alpha^s_n = \begin{cases} 1 & \text{if } s \in \mathcal{S}^w \text{ is deployed at } n \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

While the mapping and admission decisions are defined as linear variables, γ^s_{un}, $\beta^s_{un} \in [0,1]$, representing the fraction of λ^s_u to be mapped to n and admitted by n respectively. The total fraction admitted by n for s is $\beta^s_n = \sum_{u \in \mathcal{U}} \beta^s_{un}$. The inter-microservice routing decision is defined as β^s_{nm}, the fraction of total requests for s admitted by n and results are routed to dependents, hosted at m. Note that when s and its dependents are deployed in the same node, $m = n$. The admission and routing decisions are tied by the flow conservation constraint, in which input to s must equal its output:

$$\sum_{u \in \mathcal{U}} \beta^s_{un} = \sum_{m \in \mathcal{N}} \beta^s_{nm}, \forall n \in \mathcal{N} \tag{4}$$

A. Problem formulation

Now, the problem of intent-based joint scheduling of workflows can be formulated as a constrained optimisation of the decision variables, given the cost and utility functions:

$$\min_{\alpha,\beta,\gamma} \sum_{s \in \mathcal{S}^w} \sum_{n \in \mathcal{N}} \Big(\phi^s_n \alpha^s_n + \sum_{u \in \mathcal{U}} (\delta^s_u \theta^s_n + \omega^{s,q}_u \theta^s_{un}) \gamma^s_{un} + \sum_{m \in \mathcal{N}} \omega^{s,r} \theta^s_{nm} \beta^s_{nm} \Big) \tag{5}$$

Subject to the flow conservation constraint of (4), and:

$$C_\tau: \quad \sum_{s \in i} \tau^s_n + \tau^s_{nm} \leqslant \tau^{\varnothing(w)}, \tau^{\varnothing(w)} \leqslant \tau^w \ \ \forall i \in \mathcal{I}^w \tag{6}$$

$$C_c: \quad \sum_{s \in \mathcal{S}^w} \alpha^s_n \sum_{m \in \mathcal{N}} \delta^s \beta^s_{nm} \leqslant c_n, \ \ \forall n, m \in \mathcal{N} \tag{7}$$

$$C_b: \quad \sum_{s \in \mathcal{S}^w} \omega^{s,q}_u \gamma^s_{un} \leqslant b_{p_{un}}, \sum_{s \in \mathcal{S}^w} \omega^{s,r}_n \beta^s_{nm} \leqslant b_{p_{nm}} \tag{8}$$

$$\forall \quad u \in \mathcal{U}, n, m \in \mathcal{N}, p \in \mathcal{P}$$

$$C_\gamma: \quad \sum_{n \in \mathcal{N}} \gamma^s_{un} = 1, \ \ \forall s \in \mathcal{S}^w, u \in \mathcal{U} \tag{9}$$

$$C_\beta: \quad \sum_{m \in \mathcal{N}} \beta^s_{nm} = 1, \ \ \forall s \in \mathcal{S}^w, n, m \in \mathcal{N} \tag{10}$$

$$C_e: \quad \beta^s_{un} = \gamma^s_{un}, \ \ \beta^{s,t}_{nm} = \gamma^{s,t}_{nm} \tag{11}$$

$$\forall \quad s, t \in \mathcal{S}^w, r_{st} \in \mathcal{R}^w, u \in \mathcal{U}, n, m \in \mathcal{N}$$

$$C_v: \quad \alpha^s_n, \beta^s_{un}, \beta^s_{nm}, \gamma^s_{un} \geqslant 0 \tag{12}$$

Where: C_τ ensures that latency on the diameter of the workflow graph, $\tau^{\varnothing(w)}$, does not exceed the maximum response time tolerance. C_c ensures that CPU allocation does not exceed capacity, at any fog node. While, C_b ensures the upload and inter-cluster traffic do not exceed the bandwidth capacity of their paths. C_γ ensures all demand of an access node is allocated, while C_β ensures that all results of s are routed to its dependents, or back to the end user if s is the egress microservice. C_e and C_v ensure: a) consensus is reached on the fraction of demand mapped from u to n and admitted for processing by n; and b) all decision variables are positive.

The problem is NP-hard with added non-trivial challenges [20]. First, problem terms have different dimensions: $\gamma \in \mathbb{R}^{\mathcal{U} \times \mathcal{N}}$, while $\beta \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$. This does not lend to straightforward mapping between γ and β, to satisfy the consensus constraint C_e. Second, the latency constraint C_τ couples all the constraints; thereby, hinders the separation of concern of the different parts of the problem. To address the coupling of C_τ we adopt the constraint decomposition technique from [10], defining the latency tolerance per microservice as $\tau^s = \chi^s \tau^w$ where $\chi^s \in [0,1]$ is derived from the weight of s relative to the heaviest chain in the workflow. This allows to transform C_τ into: $C^{s,\gamma}_\tau$, constraint on the end-to-end latency of sending q^s from u to n; $C^s_{nm,\tau}$ constraint on end-to-end latency of routing s results from n to its peer, m; and $C^s_{n,\tau}$ constraint on processing latency of s, respectively. While $C^{s,\gamma}_\tau$ is defined in [10], here we define:

$$C^s_{nm,\tau}: \quad \tau^s_n + \frac{h_p r^s}{b_p - \beta^{s,t}_{nm} \omega^{s,r}} \leqslant \chi^s \tau^w \tag{13}$$

$$C^s_{n,\tau}: \quad \frac{c^s c_n}{c_n \mu_n - c^s \sum_{u \in \mathcal{U}} \delta^s_u \beta^s_{un}} + \tau^s_{nm} \leqslant \chi^s \tau^w \tag{14}$$

where $\beta^{s,t}_{nm}$ and β^s_{un} are transformations of β^s_{nm}, described in Section V-A as we address the dimensionality challenge. b_p and h_p are the bandwidth and hop count on $p_{nm} \in \mathcal{P}_\mathcal{N}$.

Now, aside from the large problem space of $|\mathcal{W}| \times |\mathcal{S}^w| \times |\mathcal{U}| \times |\mathcal{N}|^2$ that renders global solving methods highly expensive, none of the actors in the ecosystem possesses all the information needed to solve the problem. This means global solving approaches are infeasible. Hence, we propose a federated optimization algorithm to solve the problem. Notably, we solve the routing problem first then use the solution to derive the admission counterpart.

V. Federated Optimisation Algorithm

To solve (5) in a federated manner, we propose an optimisation algorithm combining Bender's decomposition with ADMM. First we address the dimensionality challenge by applying aggregation and rationing techniques to transform variables when used in different dimensions, then we decompose the problem into three-parts, solve them individually and reconcile the solutions iteratively until convergence is reached.

A. Addressing the dimensionality challenge

To decompose the linear terms of (5), there is a need to derive $\gamma^{s,t}_{nm}$ from γ^s_{un} and β^s_{un} from β^s_{nm}. $\gamma^{s,t}_{nm}$ is used

to solve for $\beta^{s,t}_{nm}$, deriving consensus on the volume of r^s routed from n to t, hosted at m. $\beta^{s,t}_{nm}$ is then used to calculate β^s_{nm}, the fraction of total volume of r^s routed to m. Equivelently, β^s_{un} is used to solve γ^s_{un}, to reach consensus on the volume of requests mapped by u to n, and admitted by n. ***Demand aggregation***: to obtain $\gamma^{s,t}_{nm}$, we first calculated $\gamma^s_n = \sum_{u\in\mathcal{U}} \gamma^s_{un}$ and $\gamma^t_m = \sum_{u\in\mathcal{U}} \gamma^t_{um}$, the aggregate demand for s,t, mapped to n,m respectively. Then, we define $\gamma^{s,t}_{nm} = \min(\gamma^s_{un}, \gamma^t_{um}), r_{s,t} \in \mathcal{R}^w$. This restricts $\beta^{s,t}_{nm}$ to not exceed the mapping fraction on either n or m. Then, we calculate $\beta^s_{nm} = \max(\{\beta^{s,t}_{nm} \mid t \in \mathcal{S}^{w,s}\})$, because when multiple dependents from $\mathcal{S}^{w,s}$ are deployed at m, n still routes one flow of r^s assuming that m deploys a proxy that rations input to each dependent, given respective allocation decisions. Hence β^s_{nm} should accommodate the dependent of the largest allocation at m. ***Response rationing***: upon obtaining a solution for $\{\beta^s_{nm} | \forall m \in \mathcal{N}\}$, the fraction of aggregate demand for s that n is willing to route to each m, β^s_{un} is derived by rationing $\beta^s_n = \sum_{m\in\mathcal{N}} \beta^s_{nm}$ by the ratio of access-node demand $\{\gamma^s_{un}\}$ to the aggregate demand γ^s_n. This is justified as it obeys the constraint of (4) and does not violate any other constraint. This is described as:

$$\beta^s_{un} = \beta^s_n \frac{\{\gamma^s_{un} \mid u \in \mathcal{U}\}}{\gamma^s_n} \tag{15}$$

Note that (15) constitutes deriving the per access node *admission* decision from the aggregate *routing* counterpart.

B. Two-stage problem decomposition and solution

First, we decompose the problem using principles of Bender's decomposition into an integer (*placement*) and a linear part. The latter is then decomposed into two linear problems, *mapping* and *routing*. All three are then solved in two-stage iterations: 1) an inner loop, $k \in 1, 2, \ldots$, that solves the linear sub-problems using intent-based ADMM for direct inter-cluster routing paths - iADMM (direct), given a feasible solution of the integer problem; and, 2) an outer loop, $j \in 1, 2, \ldots$ that solves the integer problem given an optimal solution of the linear sub-problems. The routing solution obtained in the inner loop is further utilized to derive the *admission* solution.

1) Solving the linear sub-problems: given β^s_{un} and an initial solution for the placement problem, $\hat{\alpha}$, the augmented Lagrangian of the mapping problem is formulated as:

$$\begin{aligned}\mathcal{L}(\gamma,\beta,\lambda) = \sum_{s\in\mathcal{S}^w}\sum_{n\in\mathcal{N}} \hat{\alpha^s_n} \sum_{u\in\mathcal{U}} (\delta^s_u\theta^s_n + \omega^{s,q}_u\theta^s_{un})\gamma^s_{un} + \\ \zeta^s_{un}(\gamma^s_{un} - \beta^s_{un}) + \frac{\rho}{2}(\gamma^s_{un} - \beta^s_{un})^2\end{aligned} \tag{16}$$

Where $\rho > 0$ is the penalty parameter of $\mathcal{L}$ and ζ^s_{un} is the Lagrange multiplier of the difference between the mapping and admission solutions. Notice that the penalty term renders the augmented Lagrangian strictly convex, irrespective of the original form of the problem. The mapping problem can be further reduced and posed as a per-access-node per-microservice set of sub-problems in the form:

$$\min_{\gamma^s_{un}} \ \hat{\alpha^s_n} \sum_{u\in\mathcal{U}} \gamma^s_{un}\Big(\delta^s_u\theta^s_n + \omega^{s,q}_u\theta^s_{un} + \zeta^s_{un} + \frac{\rho}{2}(\gamma^s_{un} - \beta^s_{un})^2\Big) \tag{17}$$

Subject to: $C^{s,\gamma}_\tau$ referenced in Section IV-A from [10], C_γ, C_e and C_v. Since the network operator can ensure sufficient bandwidth and include the cost of reliable data upload within θ^s_{un}, we omit C_b. Now, γ^s_{nm} is derived from γ^s_{un}.

Given γ^s_{nm} and $\hat{\alpha}$, the Lagrangian of the routing problem is defined as:

$$\begin{aligned}\mathcal{L}(\beta,\gamma,\lambda) = \sum_{n\in\mathcal{N}} \hat{\alpha^s_n} \sum_{m\in\mathcal{N}} \sum_{t\in\mathcal{S}^{w,s}} \hat{\alpha^t_m}\big(\omega^{s,r}\theta^s_{nm}\beta^{s,t}_{nm} + \\ \zeta^{s,t}_{nm}(\beta^{s,t}_{nm} - \gamma^{s,t}_{nm}) + \frac{\rho}{2}(\beta^{s,t}_{nm} - \gamma^{s,t}_{nm})^2\big)\end{aligned} \tag{18}$$

where α^t_m is the current placement solution of the dependent $t \in \mathcal{S}^{w,s}$ on m. The problem can be simplified and posed as a per-microservice per-fog-node set of sub-problems:

$$\min_{\beta^{s,t}_{nm}} \hat{\alpha^s_n} \sum_{t\in\mathcal{S}^{w,s}} \hat{\alpha^t_m} \sum_{m\in\mathcal{N}} \beta^{s,t}_{nm}\big(\omega^{s,r}\theta^s_{nm} - \zeta^{s,t}_{nm} + \frac{\rho}{2}(\beta^{s,t}_{nm} - \gamma^{s,t}_{nm})^2\big) \tag{19}$$

The problem of (19) is subject to: $C^s_{nm,\tau}$, $C^s_{n,\tau}$, C_s , C_β, C_e and C_v. To solve (17) and (19), we adapt a novel version of iADMM proposed in [10] and name it iADMM (direct), corresponding to direct inter-cluster routing. At iteration k of the inner loop, each access node u solves its instances of the mapping problem first to obtain $\gamma^{s,k+1}_{un}$ and share it with the respective n. The latter utilises $\{\gamma^{s,k+1}_{un} | u \in \mathcal{U}\}$ to calculate $\gamma^{s,t,k+1}_{nm}$, use it to solve n instances of the routing problem and obtain $\beta^{s,t,k+1}_{nm}$. This is then used to calculate $\beta^{s,k+1}_{nm}$, $\beta^{s,k+1}_n$ and $\beta^{s,k+1}_{un}$, as described earlier in Section V-A. Finally, $\gamma^{s,k+1}_{un}$ and $\beta^{s,k+1}_{un}$ are used to calculate the dual variable $\zeta^{s,k+1}_{un}$, while $\beta^{s,t,k+1}_{nm}$ and $\gamma^{s,t,k+1}_{nm}$ are used calculate $\zeta^{s,t,k+1}_{nm}$. The iterations continue until the residual parameters are below the primary and dual error gaps, $\epsilon_{pri}, \epsilon_{dual}$. iADMM (direct) is described in Algorithm (1).

Algorithm 1 Intent-based ADMM (direct): inner algorithm

1: Given: $\hat{\alpha}^j, \epsilon_{pri}, \epsilon_{dual}, \rho$
2: Initialize: $k \leftarrow 0$, $\gamma^s_{un}, \beta^s_{un}, \zeta^s_{un} \leftarrow \{0\}$
3: Each $n \in \mathcal{N}$ calculates τ^s_n and advertises it with θ^s_n to $\mathcal{U}$.
4: A designated network MANO calculates τ^s_{nm} and publish it with θ^s_{nm} to each $n \in \mathcal{N}$.
5: Each $u \in \mathcal{U}$ calculates τ^s_{un} and publish it with θ^s_{un} to $\mathcal{N}$.
6: **while** $s_{pri} > \epsilon_{pri}$ OR $s_{dual} > \epsilon_{dual}$ **do**
7: Each u solves (16) and publish the $\gamma^{s,k+1}_{un}$ solution to n
8: Each n: 1) calculates $\gamma^{s,t,k+1}_{nm}$ and uses it to solve (18), obtaining $\beta^{s,t,k+1}_{nm}$; 2) uses $\beta^{s,t,k+1}_{nm}$ to calculate $\beta^{s,k+1}_{nm}, \beta^{s,k+1}_n$ and $\beta^{s,k+1}_{un}$, and obtains the routing dual variable ψ^s_n; 3) calculates the dual variables $\zeta^{s,k+1}_{un}, \zeta^{s,t,k+1}_{nm}$; and 4) shares $\beta^{s,k+1}_{un}$ and $\zeta^{s,k+1}_{un}$ with u
9: $\mathcal{N}$ and $\mathcal{U}$ nodes update their respective τ^s_n, τ^s_{un} and τ^s_{nm}.
10: $\mathcal{U}$ nodes calculate s_{pri} and s_{dual}
11: $k \leftarrow k + 1$
12: **end while**
13: Each $n \in \mathcal{N}$ advertises the optimal linear solution $\hat{\beta^s_{nm}}$ and the routing dual $\hat{\psi^s_n}$ to the workflow orchestrator.

2) Solving the integer master problem: Following Bender's decomposition, the master problem is defined for the integer term of the original problem given the cost of the linear sub-problem:

$$\min_{\alpha^s} \sum_{n \in \mathcal{N}} \phi_n^s \mathcal{F}(\eta_n, \mathcal{W}) \alpha_n^s + \hat{\Upsilon}^s, \ \forall s \in \mathcal{S}^w \tag{20}$$

Subject to:

$$C_p: \quad 1 \leqslant \sum_{n \in \mathcal{N}} \alpha_n^s \leqslant |\mathcal{N}|, \ \forall s \in \mathcal{S}^w \tag{21}$$

$$C_\Upsilon^s: \quad \hat{\Upsilon}^s \geqslant \Upsilon_{init}^s \tag{22}$$

Where Υ_{init}^s is the lower bound on the cost of the linear problem, defined by the conditions of the ecosystem. C_p is the placement constraint, ensuring that s is deployed at least in one fog node (cluster), while C_Υ^s guarantees that the cost of the linear problem is higher than the lower bound on cost. At any iteration j of the outer loop, the workflow orchestrator uses $\hat{\beta_{nm}^s}$ and the routing duality variable ψ_n^s to solve (20), obtaining $\alpha_n^{s,j}$ and the placement dual variable $\Pi^{s,j}$. The latter represents the optimal cost of the linear problem, which determines the lower bound (LB) on the master problem. Meanwhile, $\hat{\beta_{nm}^s}$ is used to calculate the upper bound (UB) on the problem. Iterations of the outer loop continue until the gap between UB and LB is reduced to acceptable margin. We refer to the combination of Bender's outer Algorithm (2) with iADMM (direct) as iBADMM (direct).

Algorithm 2 Intent-based BD: the outer algorithm

1: Given: ϵ_{plc}
2: Initialize: $j \leftarrow 0, UB \leftarrow +\infty, LB \leftarrow 0, \alpha_n^{s,j} \leftarrow \{1\}$
3: **while** $UB - LB \geqslant \epsilon_{plc}$ **do**
4: Solve the sub-problems using $\alpha_n^{s,j}$ in Algorithm (1) and publish $\hat{\beta_{nm}^s}, \psi_n^s$ to the workflow orchestrator
5: Workflow orchestrator solves the master problem of (20) to obtain $\alpha_n^{s,j+1}$ and $\Pi^{s,j+1}$
6: Calculate UB and LB for the iteration
7: $j \leftarrow j + 1$
8: **end while**
9: Workflow orchestrator publishes the converged solution $\hat{\alpha_n^{s,j+1}}$ to fog MANOs, which publish it to access nodes.

VI. Evaluation

This section evaluates the proposed iBADMM (direct) algorithm, compared to state-of-the-art iADMM [10]. For placement with iADMM, we utilise Bender's algorithm proposed above and refer to the combination as iBADMM. iBADMM (direct) differs from iBADMM in: a) iBADMM does not incorporate the cost of data transmission on upload paths; and b) iBADMM does not solve the routing problem for direct inter-cluster paths. Instead, it solves for anchored paths by access nodes. The evaluation is based on analytical modelling and extensive simulations, using simmer Discrete Event simulator and the Abilene topology (11 nodes, 28 links) [21]. The topology is selected for providing a realistic setup for our experiments [22]. The network topology provides the paths used to connect fog and access nodes. The key performance indicators (KPIs) analyzed are: ***Latency Residual Budget (LRB)***, an indicator of compliance with the response time; ***Workload Greenness***, an indicator of the fraction of workload served by green energy; ***mPoP replicas***, indicating the number of fog nodes having deployed a specific microservice; and, ***Workflow Dispersion***, indicating the spread of a workflow deployment over fog nodes. The evaluation settings are summarised on GitHub aforementioned repository.

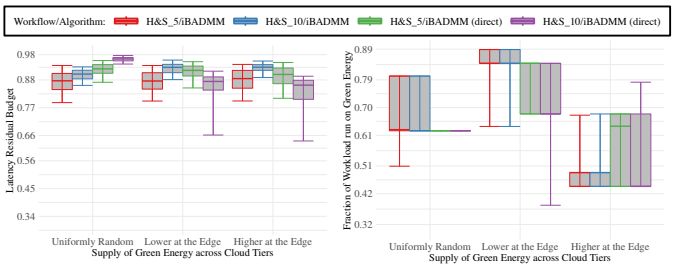
Performance is evaluated for two Hub and Spoke (H&S) workflow settings: 24 workflows, each the size of 5 microservices; and 10 workflows, each the size of 10 microservices. The first resembles large number of workflows of small dependencies, while the latter reflects a smaller set of workflows with larger dependencies. The distribution of requests generates realistic workloads, similar to that observed in [23].

Two control scenarios have been investigated: varying the supply of green energy between the edge and the cloud; and, varying the non-functional intents of workflows. The latter is achieved by varying the resource requirements (i.e. task and input/output data) per microservice and response time criticality of each workflow. Varying green energy supply highlights the trade-offs between cost and utility functions, while varying the workflow structure illustrates performance variation given application complexity.

*a) **Green Energy Variation**:* Figure 2 shows the results when varying the supply of green energy across fog nodes. Figure 2a shows the LRB of H&S workflows to be lower for iBADMM (direct) as opposite to iBADMM, with larger workflows of 10 microservices having the lowest LRB. The effect is higher when the supply of green energy is larger at the edge. This is because of the upload cost in iBADMM (direct), which results in pulling microservices towards the edge to minimize it. When the green supply is larger at the edge, the attraction force is higher. Because, the app provider too prefers placement at the edge. However, given the limited resources of the edge, the response time is higher.

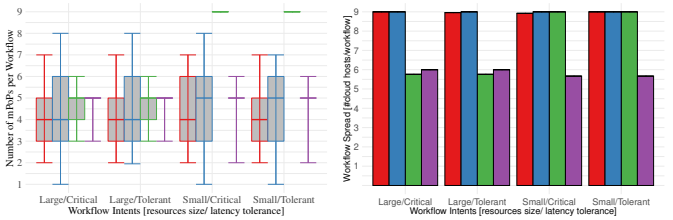
Figure 2b shows when green supply is higher in the cloud, workflow greenness is $\approx 0.70 - 0.89$. In contrast, when supply is higher at the edge, the fraction is lower $\approx 0.45 - 0.65$, with iBADMM (direct) having an improvement over iADMM by $\approx 15 - 17\%$. This is because of the direct inter-cluster routing of iBADMM (direct), causing mid-to-large fog nodes to favour placing all microservices of a workflow in one cluster and routing traffic internally. This is further confirmed by the results of mPoPs replicas and workflow dispersion; shown on the GitHub repository instead of here due to space limitation.

*b) **Workflow Variation**:* Figure 3 shows the results when varying the resource requirements per microservice in a workflow. The green supply here is lower at the edge compared to the cloud. The LRB and workflow greeness results are shown on Github instead, due to space limitation. Figure 3a shows that iBADMM (direct) replicates smaller microservices over larger number of nodes, 9 in the case of H&S_5 and an average of 5 in case of H&S_10 workflows. This is because



(a) LRB (b) Workflow greenness

Fig. 2. Performance Indicators between iBADMM and iBADMM (direct), given variation of Green Energy supply.



(a) mPoPs (b) Workflow spread

Fig. 3. Performance Indicators between iBADMM and iBADMM (direct), given variation of Workflow non-functional intents.

smaller workflows have less likelihood of violating any of the constraints given the low number of dependencies and low resource requirements. However, workflows of higher dependency have higher variation of microservice requirements. This results in tighter constraints that limit the number of mPoPs. Figure 3b confirms this showing workflow distributability of iADMM (direct) for process intensive and/or large workflows to be smaller in the range of $\approx 5-6$ nodes, as opposite to iADMM distributing workflows over all nodes.

VII. Conclusions

This work proposed a novel intent-based federated optimization algorithm for scheduling fog-native microservice workflows, over multi-domain edge-to-cloud ecosystem. This included solving the problem of workflow: placement, mapping, routing and admission. The algorithm, iBADMM (direct), is based on a combination of Bender's decomposition and Alternating Direction Method of Multipliers. It reduces information sharing among actors to the operational cost, with no requirement to exchange business-sensitive counterparts. This allows for flexibility and scalability in workflow scheduling, while preserving the autonomy of actors. The performance of the algorithm has been evaluated analytically, and compared to state-of-the-art iBADMM algorithm. Evaluation results illustrated the superior performance of the proposed algorithm and revealed non-trivial trade-offs between workflow intents and the heterogeneity of green energy in the ecosystem. This included the interplay between green energy supply and diverse energy prices between edge and cloud, with more expensive green edge energy reducing the gains of proximity.

References

[1] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*. New York, NY, USA: Association for Computing Machinery, 2015, p. 37–42.

[2] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018.

[3] O. C. A. W. Group, "Openfog reference architecture for fog computing," Fremont, CA, USA, Tech. Rep. OPFRA001.020817, Feb 2017.

[4] A. M, B. P, and C. L, "Trends in data centre energy consumption under the european code of conduct for data centre energy efficiency," *ENERGIES*, vol. 10, no. 10, p. 1470, 2017.

[5] Y. Sverdlik, "What is the Data Center Cost of 1kW of IT Capacity?" https://www.datacenterknowledge.com/archives/2016/08/23/what-is-the-data-center-cost-of-1kw-of-it-capacity, Aug 2016, [Online; accessed 12-March-2022].

[6] SKhynix, "Beyond Clean Energy: Technology Must Help Address Data Center Challenges," https://news.skhynix.com/hed-beyond-clean-energy-technology-must-help-address-data-center-challenges/, Dec 2021, [Online; accessed 08-July-2022].

[7] G. Riley, "Green Data Centres and Solar Power," https://blog.spiritenergy.co.uk/commercial/green-data-centres-solar-power, Feb 2020, [Online; accessed 08-July-2022].

[8] A. Conejo *et al.*, *Decomposition Techniques in Mathematical Programming: Engineering and Science Applications*. Springer Berlin Heidelberg, 2006.

[9] S. Boyd *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, p. 1–122, jan 2011. [Online]. Available: https://doi.org/10.1561/2200000016

[10] M. AL-Naday, T. Goethals, and B. Volckaert, "Intent-based decentralized orchestration for green energy-aware provisioning of fog-native workflows," in *2022 18th International Conference on Network and Service Management (CNSM)*, 2022, pp. 184–190.

[11] O. H. Ahmed *et al.*, "Using differential evolution and moth–flame optimization for scientific workflow scheduling in fog computing," *Applied Soft Computing*, vol. 112, p. 107744, 2021.

[12] M. Mokni *et al.*, "Cooperative agents-based approach for workflow scheduling on fog-cloud computing," *Journal of Ambient Intelligence and Humanized Computing*, apr 2021.

[13] A. M. Yadav, K. N. Tripathi, and S. C. Sharma, "A bi-objective task scheduling approach in fog computing using hybrid fireworks algorithm," *The Journal of Supercomputing*, vol. 78, no. 3, pp. 4236–4260, aug 2021.

[14] H. Xu and B. Li, "Joint request mapping and response routing for geo-distributed cloud services," in *2013 Proceedings IEEE INFOCOM*, Apr 2013, pp. 854–862.

[15] S. Yang *et al.*, "Cloudlet placement and task allocation in mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5853–5863, 2019.

[16] Y. Yu *et al.*, "Network function virtualization resource allocation based on joint benders decomposition and admm," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1706–1718, 2020.

[17] U. M. Malik *et al.*, "Energy efficient fog computing for 6g enabled massive iot: Recent trends and future opportunities," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[18] M. Aldossary and H. A. Alharbi, "Towards a green approach for minimizing carbon emissions in fog-cloud architecture," *IEEE Access*, vol. 9, pp. 131 720–131 732, 2021.

[19] T. A. for IoT and E. C. Innovation, "Aioti strategic research and innovation agenda," 2023. [Online]. Available: https://aioti.eu/wp-content/uploads/2023/01/AIOTI-SRIA-Final.pdf

[20] H. K. Apat, R. Nayak, and B. Sahoo, "A comprehensive review on internet of things application placement in fog computing environment," *Internet of Things*, p. 100866, 2023.

[21] S. Knight *et al.*, "The Internet Topology Zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, Oct 2011.

[22] J. L. Herrera *et al.*, "Qos-aware fog node placement for intensive iot applications in sdn-fog scenarios," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13 725–13 739, 2022.

[23] B. Liu, Y. Lin, and Y. Chen, "Quantitative workload analysis and prediction using google cluster traces," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2016, pp. 935–940.