INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2025, VOL. 71, NO. 4, PP. 1–7
Manuscript received August 19, 2025; revised October 2025. doi: 10.24425/ijet.2025.155482

Multi-criteria MILP model for geo-distributed cloud configuration with advanced client preference integration

Izabela Zoltowska, and Kacper Kozerski

***Abstract*—This paper addresses the problem of selecting a cloud infrastructure configuration for a geo-distributed enterprise. It extends the well-known virtual machine (VM) placement problem to consider multiple datacenters so they can serve a distribution of end-users in their geographic locations in an optimal way in terms of low end-user latency, and acceptable costs. We approach this problem by formulating a multi-criteria mixed integer linear program (MILP) that integrates an aspiration/reservation-based modeling of the client's preferences. A newly proposed model supports the selection of virtual instances across cloud regions, ensuring flexible trade-offs among QoS objectives: total infrastructure cost, user distance, and edge-to-central latency. Case study results based on Google datacenters in Europe demonstrate the flexibility of our method in providing Pareto-optimal solutions aligned with varied preferences. The approach contributes to the growing preference-aware cloud resource allocation field and offers a scalable solution to the service composition problem in heterogeneous cloud environments.**

***Keywords*—cloud computing; IaaS; QoS; Virtual Machine Placement; Aspiration/Reservation Reference Point Method**

I. Introduction

CLOUD configuration refers to the process of selecting and organizing on-demand computing resources such as hardware, operating systems, storages, networks, databases, etc., in a network-based system to meet specific user or application requirements [1]. It is enabled through the Infrastructure-as-a-Service (IaaS) cloud paradigm.

According to the review paper [2], the main challenges of that process are how to compose complex cloud infrastructures that satisfy diverse and often conflicting requirements, including non-functional Quality of Service (QoS) related criteria such as costs, availability, etc. This led researchers to incorporate the QoS parameters when considering the distributed application structure and deployment, leading to the so-called QoS-aware cloud service composition problem [3].

When services start growing in size, they might want to cater to a larger and more geographically distributed clientele. One example of such a geo-distributed enterprise operation is the provisioning of distributed resources such as virtual machines (VM) [4]. However, in recent years, the connection between QoS and the aforementioned spatial distribution was made apparent, i.e., placement of VMs far from the data center leading to latency issues [5], [6]. High access latency is an important issue in geo-distributed cloud environments, undertaken by cloud providers to prevent decreasing revenue caused by degradation of user satisfaction. The significance of geo-based allocation was further demonstrated by inclusion in the list of cloud architecture archetypes, e.g., DNS Load Balancing with Geo-Mapping Multi-Region archetype [7].

Izabela Zoltowska, and Kacper Kozerski are with Faculty of Electronics and Information Technology, Warsaw University of Technology, Warsaw, Poland (e-mail: izabela.zoltowska, kacper.kozerski.stud@pw.edu.pl).

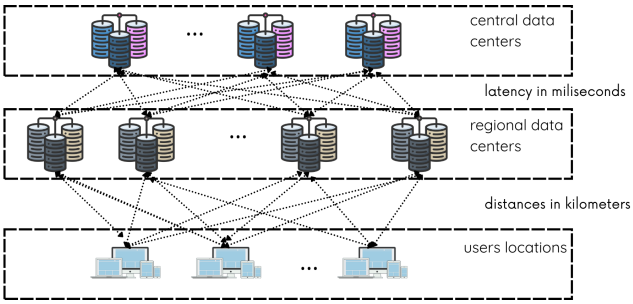

Fig. 1. Considered cloud architecture with three layers, where the top layer is a central, high-performance data center, and the lower layers are edge servers and users.

The issues stemming from geolocation are further exacerbated when we consider a tree-like cloud architecture with three layers, where the top layer is a central, high-performance data center, and the lower layers are edge servers and users, as illustrated in Figure 1. Here, there are 2 dimensions of spatial distribution influence on the whole system – the distances between clients and regional data centers, and between central and regional centers. Building the architecture in the cloud in such a case involves purchasing resources in the specified locations. It can be a challenging task so enterprises often decide to rely on auto-scaling solutions offered by cloud providers, which can lead to unnecessarily high costs [8]. On the other hand, multi-region deployments within a cloud provider's network configured by the clients themselves (or by the broker representing the client) allows for substantial benefits.

Challenges associated with configuring cloud architectures are often approached from the cloud providers point of view, where focus is on the energy usage minimization, profit maximization, resources utilization, etc. as reviewed in [9].

Multi-criteria optimization is a natural way of handling the VM placement problem, where Mixed Integer Programming (MILP) model is formulated to provide optimal solution with exact algorithms. According to review paper [6], the problem is often reduced to a single objective with constraints imposed on other, less important criteria. However, to achieve trade-off solutions all criteria should be optimized simultaneously and in such a case most existing models use weighted sum approach [6]. Such a simple approach has several drawbacks, as discussed later. Moreover, current models give little consideration to user-oriented goals. When they do, these goals are often simplified into fixed performance constraints, such as Service Level Agreements (SLAs), see e.g, [10]. Obviously, the goals of service provider and client are often conflicting, e.g., allocation of the edge units closer to clients might lower the latency, but also lower the utilization rates, which might in turn raise the provider's costs. One of the approaches to reconcile the conflicts is to incorporate the client's preferences into the model, as suggested in the survey [11]. Previous papers aiming in incorporating SLAs considered on-demand user requests from individual location, see e.g. [12] where multi-objective evolutionary NSGA-II algorithm is used to develop method to suggest the best datacenter, based on the user's request and SLAs.

The most common approaches to express preference models ultimately lead also to weight creation and optimization using the weighted sum [13]. This formulation, while easy to analyze at first, suffers from certain issues, like the inability to explore the whole Pareto front or the tendency to select more extreme solutions. Other common problem is choosing the proper weight vector, especially difficult when objectives represent different physical units. Therefore, application of alternative methods might be recommended. One such method is the Reference Point Method, which expresses the preference using aspiration/reservation-based modeling [14], enabling flexible trade-offs between objectives. This approach enables interactive exploration of Pareto non-dominated solutions, as successfully demonstrated in problems from energy markets domain, see e.g. [15], [16].

The goal of this paper is to provide an optimization model that supports the decision of selecting the distributed resource allocation plan, along with a flexible and easy-to-use preference modeling. To our best knowledge, previous works do not incorporate geo-distributed clients' preferences toward QoS in cloud allocation models. The resource planning and provisioning configuration problem is considered for the reservation model, which establishes base infrastructure of continuously working VMs that may be further supplemented with additional on-demand, auto-scaling tools [17].

We strive to achieve it by formulating it as a multi-criteria MILP model that integrates the reference point-based preference modeling to select trade-offs among QoS objectives: total infrastructure cost, user-to-distance, and edge-to-central latency. The modeled situation is the aforementioned tree architecture, where the top layer is a central, high-performance data center, and the lower layers are edge servers and users, as illustrated in Figure 1. This can be seen as a version of DNS Load Balancing with Geo-Mapping Multi-Region archetype [7] where additional high-performance central unit, is used, for instance, for processing data collected from all regions.

This contributes directly to the research directions highlighted in the survey by Alashaikh et al. [11], addressing the integration of client preferences in virtual machine placement within edge-central cloud architectures. Our method enables the selection of infrastructure configurations that are Pareto-optimal and sensitive to diverse, incommensurable criteria such as cost, latency, and geographic distance. This approach not only aligns with the survey's call for richer preference modeling but also introduces a practical decision-support tool that directly attaches the client's preferred values for QoS criteria, thus advancing the state-of-the-art in preference-aware Virtual Machines placement.

II. System model and problem formulation

A. Problem formulation

We consider a static decision-making problem faced by an enterprise client seeking the IaaS service to deploy a globally accessible distributed web application. With an estimated number of client's end users W_u located in different geographic regions $u \in U$, the requirement is to reserve a possibly low-cost infrastructure consisting of basic resources $r \in \mathcal{R}$ (CPU, memory, storage, etc.), guaranteeing a possibly low latency. That is why, when considering the set of possible VM instance allocation regions, two layers should be established:

- **Edge layer:** A subset of regional edge components $e \in R^e$ that handle latency-sensitive application modules, such as user interfaces or real-time data processing. Resources located at these regions are limited, and specific configurations of VMs are available.
- **Central layer:** A single data center node responsible for aggregating and processing data from all edge locations. This component can be assigned to one of the central nodes $c \in R^c$. It demands high computational capacity and low-latency connections to the edge layer.

Client requirements. The client aims to rent cloud infrastructure from a provider, selecting specific VM types and deployment regions in the considered period of reservation. The client should specify the hardware configuration of requested VMs, such as required processor speed, memory size, disk space, etc. [6]. The VMs will be created accordingly, and run in chosen regions on the cloud provider's infrastructure. Thus, data that are the basis for the decision include the following: distances $D_{u,e}$ in kilometers from each end user region u to each acceptable edge location e, and minimum aggregated resource type r requirement $P_r^{\min}$ per number of users.

Cloud specification. The resources that constitute IaaS are described by the set of VM types $t \in T^e$ available at location e, where each type has its own capacity P_{tr} regarding a specific quantity of resources r. Each edge location has a maximum number $V_{te}^{\max}$ of VMs of type t. Additionally, the latency in milliseconds L_{ec} between each edge region e and its connected central nodes c is provided. All considered deployment options c for central unit fulfill the client's requirements. Costs of renting each instance type t in edge region e are given as C_{te}; they scale linearly with usage time. Cost F_c of renting

the central unit in location c is fixed. All costs are expressed in \$/h.

The solution to this problem is to select an optimal configuration set of instance types and deployment regions such that:

- All resource requirements and capacities constraints are satisfied;
- The total cost of leasing instances is minimized;
- Maximum latency between edge nodes and the central node is minimized;
- Maximum distance between end users and their assigned edge nodes is minimized.

Client preferences over these diversified, competing objectives are integrated into a multi-criteria integer programming model using an advanced aspiration/reservation-based scalarization technique [14]. In this approach the client provides preferences toward criteria by declaring reservation and aspiration levels, which are provided in exact quantities of criteria – reservation is the minimum acceptable performance of consecutive criteria, while aspiration states their preferable values. It is intuitive and straightforward way, in contrast to weight parameters used e.g. in [10], where a machine learning model is used to determine normalized weights used further in the multi-criteria model. Moreover, the aspiration/reservation-based framework guarantees to achieve Pareto solution most close to user's preferences, while weighted sum may be unable to achieve some efficient solutions [13]. The approach transforms the multicriteria problem to the maximization of the minimum partial achievement problem [14], [16].

B. Model formulation

The problem is formulated as a multi-criteria mixed integer linear programming (MILP) model that integrates an aspiration/reservation-based modeling of client preferences.

Formally, the notation is stated as follows:

Sets:

$r \in \mathcal{R}$ – types of resources, i.e., CPU, memory, storage
$e \in R^e$ – possible edge allocation regions,
$t \in T^e$ – types of instances (virtual machines) available in region e,
$c \in R^c$ – possible central unit allocation regions,
$u \in U$ – set of end users located in different geographical regions.

Parameters:

C_{te} – cost of renting an instance of type t in edge region e,
F_c – cost of renting the central unit in region c,
D_{ue} – distance in kilometers between user u and edge region e,
L_{ec} – latency in milliseconds between edge regions e and central region c,
P_{tr} – quantity of resources of type r available in VM of type t,
W_u – estimated number of users in region u,
$P_r^{\min}$ – minimum quantity of resource type r required per user supported in edge instances,
$V_{te}^{\max}$ – maximum number of VM of type t available in edge region e.

Decision Variables:

x_{te} – number of edge instances of type $t \in T$ allocated in region e,
y_c – binary variable indicating whether a central unit is located in region c,
v_e – binary variable indicating whether edge region e is used for connection with thecentral region
z_{ue} – binary variable indicating whether end users in region u are connected to the edge e.

Criteria-related Variables and Parameters:

q_1 – minimized cost of utilized infrastructure,
q_2 – minimized largest distance between end users and edge infrastructure,
q_3 – minimized largest latency between edge node and central unit,
a_i – partial achievement measure of the criterion q_i, $i = \{1, 2, 3\}$, with respect to the corresponding aspiration and reservation levels (q_i^a and q_i^r, respectively); free variable,
$\underline{a}$ – the worst partial achievement among all a_i; free variable,
q_i^a – aspiration target value of criterion q_i, $i = \{1, 2, 3\}$; parameter representing client's preference,
q_i^r – reservation target value of criterion q_i, $i = \{1, 2, 3\}$; parameter representing client's preference.

The optimization model consists of the following objective functions and constraints:

$$q_1 = \sum_{e \in R^e} \sum_{t \in T^e} C_{te} x_{te} + \sum_{c \in R^c} F_c y_c, \tag{1}$$

$$q_2 \geq D_{ue} z_{ue}, \qquad \forall u \in U, e \in R^e, \tag{2}$$

$$q_3 \geq L_{ec}(-1 + v_e + y_c), \qquad \forall e \in R^e, c \in R^c, \tag{3}$$

$$a_i \leq \gamma(q_i^r - q_i)/(q_i^r - q_i^a), \qquad \forall i = 1, \ldots, 3, \tag{4}$$

$$a_i \leq (q_i^r - q_i)/(q_i^r - q_i^a), \qquad \forall i = 1, \ldots, 3, \tag{5}$$

$$a_i \leq \alpha(q_i^a - q_i)/(q_i^r - q_i^a) + 1, \qquad \forall i = 1, \ldots, 3, \tag{6}$$

$$a_i \geq \underline{a}, \qquad \forall i = 1, \ldots, 3, \tag{7}$$

$$\sum_{t \in T^e} P_{tr} x_{te} \geq P_r^{\min} \sum_{u \in U} z_{ue} W_u, \qquad \forall e \in R^e, r \in \mathcal{R}, \tag{8}$$

$$\sum_{e \in R^e} z_{ue} = 1, \qquad \forall u \in U, \tag{9}$$

$$\sum_{u \in U} z_{ue} \leq |U| v_e, \qquad \forall e \in R^e, \tag{10}$$

$$\sum_{c \in R^c} y_c = 1, \tag{11}$$

$$x_{te} \leq V_{te}^{\max} v_e, \qquad \forall e \in R^e, t \in T^e. \tag{12}$$

Constraints (1)-(3) determine values of consecutive optimization criteria: total cost, maximum distance from edge, and maximum latency between edge nodes and central unit, respectively. All three criteria are minimized by maximizing the achievement functions that show normalized satisfaction of the client when the value q_i of specific i criterium reaches a level below aspiration, i.e., when $q_i \leq q_i^a$, or above reservation,

i.e., when $q_i \geq q_i^r$. Specifically, the linear constraints (4)-(6) are stated, where α and γ are parameters $0 < \alpha < 1 < \gamma$ representing additional satisfaction/dissatisfaction. Finally, the constraint (7) determines the worst achievement $\underline{a}$, maximized in the following objective:

$$\text{objective function} = \text{lex}\max\Big(\underline{a}, \quad \sum_i a_i\Big) \tag{13}$$

and its practical, approximate formulation:

$$\text{objective function} = \max \underline{a} + \varepsilon \sum_i a_i \tag{14}$$

The second term in (13) is used for regularization, introduced to guarantee the solution efficiency in a non-unique optimal solution. Such an approach to multi-criteria optimization was introduced in [14], and applied in several areas, see e.g., [15]. The so-called reference-point method offers significant advantages in multi-criteria optimization, particularly over the weighted sum approach, as it guarantees Pareto-optimal solutions and is inherently well-suited for handling incomparable criteria expressed in different physical units – such as cost, geographic distance, and latency in our case – without requiring artificial normalization, see [11].

Following constraints are directly related to infrastructure placement decisions. Constraints (8) ensure the required capacity of resource types (i.e., computational and memory resources) at the edge are fulfilled. Constraint (9) ensures each user location is connected to exactly one edge region. Constraint (10) guarantees that any edge region serving users is also connected to the central unit. Constraint (11) ensures the selection of exactly one central region.

Constraint (12) ensures that edge instances are only allocated in regions connected to user locations without exceeding their capacities.

III. Case Study

In this section, we show effectiveness of our approach on a case study assuming geographically dispersed enterprise that considers provisioning VMs from Google Cloud. We analyze three problem instances by varying the user preferences. Different solutions obtained allow to compare trade-offs and performance gains. The newly proposed model was implemented in the AMPL software and solved using CPLEX 22.1.1.

Five user locations were assumed – Athens, Lisbon, Oslo, Rome, and Warsaw – with appropriate predicted user counts, which can be found in Table I. It was assumed that each user requires at a minimum 1 vCPU and 3GiB of RAM.

TABLE I
ESTIMATED USER COUNTS AT GIVEN LOCATIONS

User location	User count
Athens	50
Lisbon	100
Oslo	30
Rome	10
Warsaw	100

Data on latency was found on the website [18]. As the latency datasheet didn't include information about all of the available European servers, only the locations in the Netherlands, Belgium, Frankfurt, London, and Zurich were selected for possible central and edge sites. The exact data on latencies can be found in Table II. For simplicity, it was assumed that the same locations would have 0 latency. The distances between locations are in the Table III (where only a country was specified, the capital city was assumed). The technical parameters and prices were scraped from the Google Cloud Platform website [19]. For the central server, the strongest C4 high-CPU machine was required. In the case of edge locations, the 3 weakest C4 standard machine types were selected - the appropriate prices and parameters can be found in Table IV, Table V. As the case study was small, upper limits on resources weren't taken into account.

TABLE II
LATENCIES BETWEEN SITES [MS]

Edge \ Central	Belgium	London	Frankfurt	Netherlands	Zurich
Belgium	0.000	6.966	7.417	7.098	16.461
London	6.843	0.000	13.851	10.566	19.911
Frankfurt	7.422	13.926	0.000	7.471	10.182
Netherlands	7.209	10.66	7.568	0.000	14.024
Zurich	16.252	19.941	7.832	14.071	0.000

TABLE III
ROUNDED DISTANCES BETWEEN SITES [KM]

User \ Edge	Belgium	London	Frankfurt	Netherlands	Zurich
Athens	2092	2395	1803	2168	1620
Lisbon	1715	1588	1893	1867	1725
Oslo	1086	1154	1099	913	1403
Rome	1175	1436	960	1299	685
Warsaw	1161	1450	891	1095	1044

TABLE IV
PRICING OF TECHNOLOGIES AT SITES [$/H]

Site \ Tech.	Central	vm1	vm2	vm3
Netherlands	8.57324160	0.10170930	0.20755350	0.41510700
Belgium	8.99047373	0.10665916	0.21765446	0.43530892
Frankfurt	9.63469056	0.11430188	0.23325060	0.23325060
London	9.30809088	0.11042724	0.22534380	0.45068760
Zurich	11.42200627	0.13550584	0.27652054	0.55304108

We present results assuming three different preference cases:

TABLE V
TECHNOLOGY SPECIFICATIONS

Technology	CPU [no]	RAM [GiB]
central	192	384
vm1	2	7
vm2	4	15
vm3	8	30

- Case 1 – most emphasis is put on distances from edge instances expressed in quite tight reservation and aspiration points;
- Case 2 – low-cost solution is preferred by providing aspiration at zero cost;
- Case 3 – lowest possible latency is required, expressed as aspiration at zero latency.

Results, depending on different aspirations and reservations in each case, are shown in Tables VI and VII.

In the first case, we set a tight aspiration for maximal distance and left other criteria within satisfiable bounds. Visualization of the relation between aspiration/reservation values and obtained solution of criteria is provided in Figures 2. To facilitate demonstration we reduced the criterion space to two dimensions, examining relations between each pair of criteria. One can observe that despite completely different measures of criteria, the solution was correctly determined. In the second case, we enforced a very strict, unsatisfiable goal for both distance and cost, but since the reservation of cost was also unsatisfiable, the optimization process tried its best to achieve as low cost as possible. This confirms that the model is flexible and robust – even if the reservation point is not achievable, i.e. the cost in Case 2, still the best possible solution is found. In the third case, we set more liberal goals for both costs and distances, but tried to reach 0 latency, which, under our assumption, is possible. By examining only the criteria, depending on the specific criteria pair, different solution can be considered as most balanced, trade-off proposition.

Visualizations of resulting cloud configuration are shown in Figure 3. We can observe that shifting our focus to different criteria changed the structure of the solution significantly. Solution to Case 1 resulted in a very spread-out localization of the edge stations. Case 2 resulted in allocating edge instance in place with lowest cost – Frankfurt, and the cheapest central node possible – in Netherlands. Finally, solution to Case 3 resulted in clustering all of the edges and central in the same location – Frankfurt. These results, supported by specific values of criteria allow to make most informed decision.

TABLE VI
RESULTING CRITERIA VALUES q_i OBTAINED UNDER DIFFERENT ASPIRATION q_i^a AND RESERVATION q_i^r PREFERENCES IN DIFFERENT CASES. THE VALUES WERE ROUNDED TO 2 DIGITS.

Case	q_1^r	q_1^a [$/h]	q_1	q_2^r	q_2^a [km]	q_2	q_3^r	q_3^a [ms]	q_3
1	100	20	22.51	3000	1000	1715.35	20	1	7.83
2	10	0	17.03	3000	200	1893.48	20	15	7.47
3	40	20	18.09	3000	1500	1893.48	20	0	0.00

TABLE VII
RESULTING ALLOCATIONS OF CENTRAL AND EDGE COMPUTING UNITS, ALONGSIDE VM COUNTS, OBTAINED IN THE CONSIDERED CASES OF DIFFERENT PREFERENCES.

Case	Central unit	Edge units
1	Frankfurt	Belgium (50 vm1)
		Frankfurt (18 vm1)
		Zurich (25 vm1)
2	Netherlands	Frankfurt (37 vm3)
3	Frankfurt	Frankfurt (37 vm3)

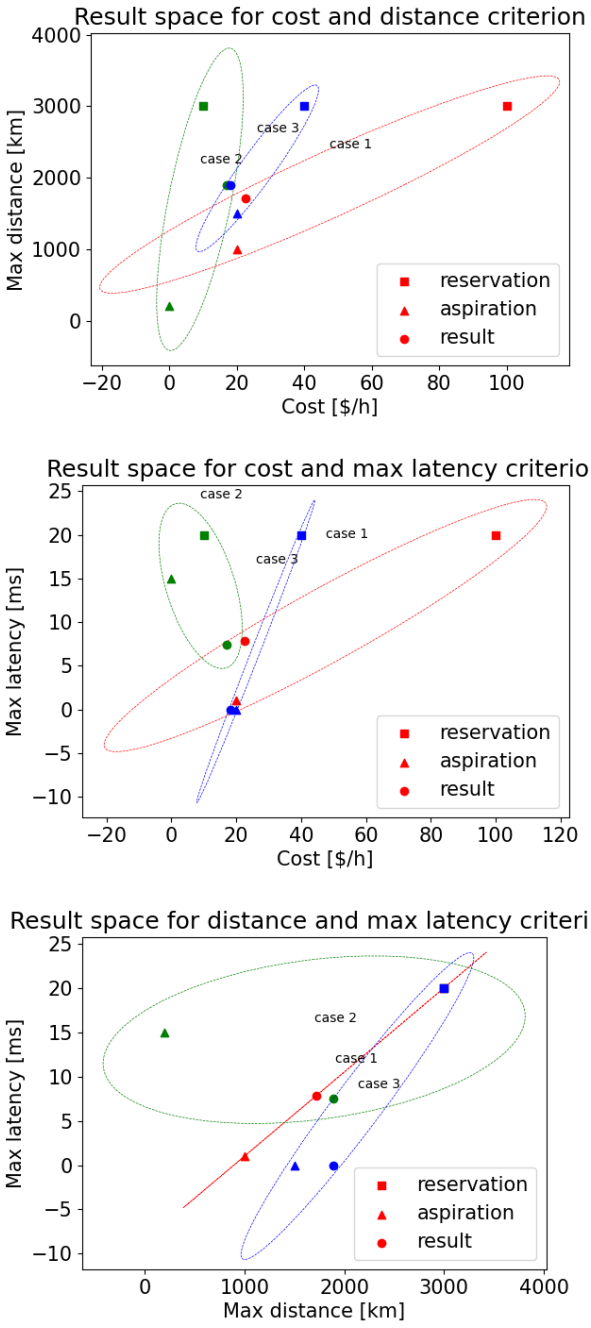

Fig. 2. Pareto solutions obtained with three different cases of preferences toward criteria, illustrated for each pair of criteria. For clarity, entries from each case are enclosed in ellipsoids.

IV. Conclusions

In this paper, we propose a multi-criteria MILP model to support the cloud infrastructure configuration choice of the enterprise client. The model takes the client's preferences toward cost and QoS and determines The required number of VMs of specific providers' types in selected regions to connect with high-performing central unit machines, to meet the geo-distributed demands.

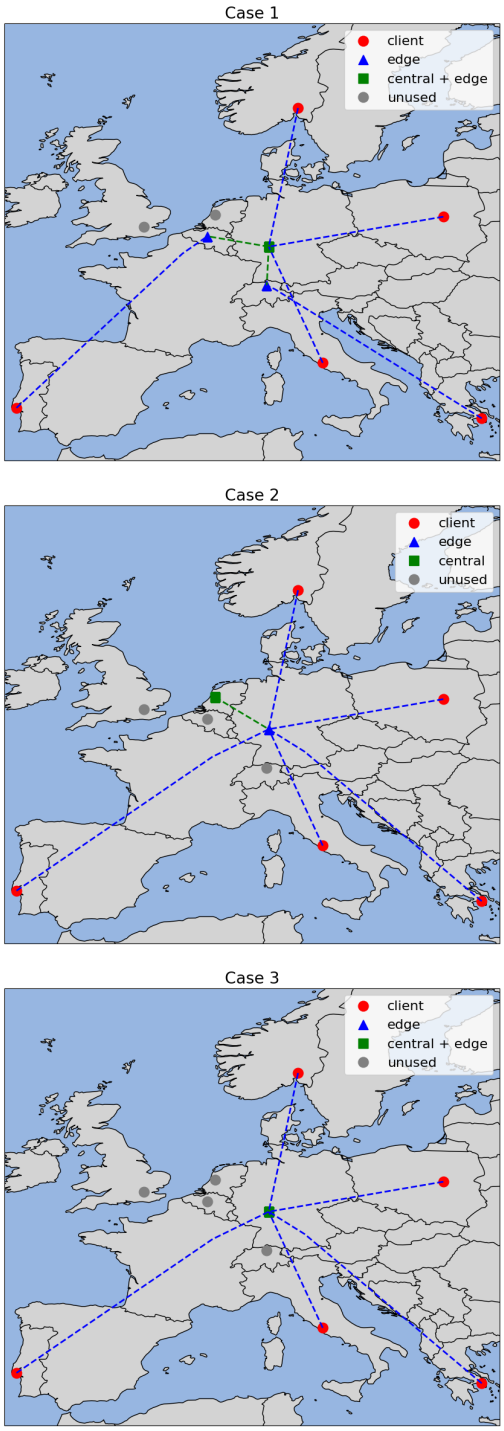

Fig. 3. Resulting edge and central locations for VMs obtained with three cases of preferences toward criteria.

The case study results demonstrate that expressing preferences in the applied Reference Point Method is straightforward and can significantly affect the resulting configuration of the edge-central infrastructure. The observed flexibility confirms the model's ability to adapt to diverse client preferences and trade-offs.

In future work, we intend to evaluate the approach on a real-world instance. The most promising area is to explore how this approach could be integrated into a dynamic Infrastructure as Code (IaC) framework to handle uncertain demand. We expect that our optimization model could be used to support continuous management of geo-distributed cloud infrastructure.

REFERENCES

[1] R. Buyya, C. Vecchiola, and S. T. Selvi, *Mastering cloud computing: foundations and applications programming*. Newnes, 2013. [Online]. Available: https://dl.acm.org/doi/book/10.5555/2531413

[2] V. Hayyolalam and A. A. P. Kazem, "A systematic literature review on qos-aware service composition and selection in cloud environment," *Journal of Network and Computer Applications*, vol. 110, pp. 52–74, 2018. [Online]. Available: https://doi.org/10.1016/j.jnca.2018.03.003

[3] A. Jula, E. Sundararajan, and Z. Othman, "Cloud computing service composition: A systematic literature review," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3809–3824, 2014. [Online]. Available: https://doi.org/10.1016/j.eswa.2013.12.017

[4] O. Afolalu, "Enterprise Networking Optimization : A Review of Challenges, Solutions, and Technological Interventions," *Future Internet*, vol. 17, no. 133, pp. 1–21, 2025. [Online]. Available: https://doi.org/10.3390/fi17040133

[5] M. Malekimajd, A. Movaghar, and S. Hosseinimotlagh, "Minimizing latency in geo-distributed clouds," *The Journal of Supercomputing*, vol. 71, no. 12, pp. 4423–4445, 2015. [Online]. Available: https://doi.org/10.1007/s11227-015-1538-1

[6] H. Talebian, A. Gani, M. Sookhak, A. A. Abdelatif, A. Yousafzai, A. V. Vasilakos, and F. R. Yu, "Optimizing virtual machine placement in IaaS data centers: taxonomy, review and open issues," *Cluster Computing*, vol. 23, pp. 837–878, 2020. [Online]. Available: https://doi.org/10.1007/s10586-019-02954-w

[7] A. Berenberg and B. Calder, "Deployment archetypes for cloud applications," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1–48, 2022. [Online]. Available: https://doi.org/10.1145/3498336

[8] M. Ciavotta, G. P. Gibilisco, D. Ardagna, E. D. Nitto, M. Lattuada, and M. A. A. da Silva, "Architectural design of cloud applications: A performance-aware cost minimization approach," *IEEE Transactions on Cloud Computing*, vol. 10, no. 3, pp. 1571–1591, 2022. [Online]. Available: https://doi.org/10.1109/TCC.2020.3015703

[9] W. Attaoui and E. Sabir, "Multi-criteria virtual machine placement in cloud computing environments: a literature review," in *2024 International Conference on Ubiquitous Networking (UNet)*, vol. 10. IEEE, 2024, pp. 1–11. [Online]. Available: https://doi.org/10.1109/UNet62310.2024.10794708

[10] S. Rawas, A. Zekri, and A. El-Zaart, "LECC: Location, energy, carbon and cost-aware VM placement model in geo-distributed DCs," *Sustainable Computing: Informatics and Systems*, vol. 33, p. 100649, 2022. [Online]. Available: https://doi.org/10.1016/j.suscom.2021.100649

[11] A. Alashaikh, E. Alanazi, and A. Al-Fuqaha, "A survey on the use of preferences for virtual machine placement in cloud data centers," *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–39, 2021. [Online]. Available: https://doi.org/10.1145/3450517

[12] H. Ziafat and S. M. Babamir, "A method for the optimum selection of datacenters in geographically distributed clouds," *The Journal of Supercomputing*, vol. 73, no. 9, pp. 4042–4081, 2017. [Online]. Available: https://doi.org/10.1007/s11227-017-1999-5

[13] M. Whaiduzzaman, A. Gani, N. B. Anuar, M. Shiraz, M. N. Haque, and I. T. Haque, "Cloud Service Selection Using Multicriteria Decision Analysis," *The Scientific World Journal*, vol. 2014, no. 1, p. 459375, 2014, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2014/459375. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1155/2014/459375

[14] A. P. Wierzbicki, *Reference Point Approaches*. Boston, MA: Springer US, 1999, pp. 237–275. [Online]. Available: https://doi.org/10.1007/978-1-4615-5025-9_9

[15] M. Kaleta, W. Ogryczak, E. Toczyłowski, and I. Zoltowska, "On Multiple Criteria Decision Support for Suppliers on the Competitive Electric Power Market," *Annals of Operations Research*, vol. 121, no. 1-4, pp. 79–104, 2003. [Online]. Available: https://doi.org/10.1023/A:1023351118725

[16] I. Zoltowska, "Risk preferences of ev fleet aggregators in day-ahead market bidding: Mean-cvar linear programming model," *Energies*, vol. 18, no. 1, p. 93, 2024. [Online]. Available: https://doi.org/10.3390/en18010093

[17] K. Sumalatha and M. S. Anbarasi, "A review on various optimization techniques of resource provisioning in cloud computing." *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 9, no. 1, 2019. [Online]. Available: http://doi.org/10.11591/ijece.v9i1.pp629-634

[18] C. Kumar, "How much is Google Cloud Latency (GCP) between Regions? — geekflare.com," https://geekflare.com/cloud/google-cloud-latency/, [Accessed 18-04-2025].

[19] Google, "Gcp pricing," https://cloud.google.com/compute/all-pricing?hl=pl#section-1, [Accessed 18-04-2025].