SCIENCE CHINA
Information Sciences

• LETTER •

April 2024, Vol. 67, Iss. 4, 149103:1–149103:2
https://doi.org/10.1007/s11432-023-3957-4

Adaptive joint configuration optimization for collaborative inference in edge-cloud systems

Zheming YANG[1,4], Wen JI[1,3*] & Zhi WANG[2,3]

[1]*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190*, China;*
[2]*Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen* 518057*, China;*
[3]*Peng Cheng Laboratory, Shenzhen* 518055*, China;*
[4]*University of Chinese Academy of Sciences, Beijing* 100049*, China*

Received 25 July 2023/Revised 13 November 2023/Accepted 19 February 2024/Published online 27 March 2024

With the rapid development of technologies such as deep learning and the Internet of Things (IoT), the deployment and application of various IoT devices are becoming increasingly widespread, resulting in the inference tasks generated increasing dramatically [1]. Many deep neural network (DNN) models are deployed on servers for performing inference tasks. When dealing with many resource-intensive tasks, it is difficult to satisfy requirements with ultra-low delay if all inference tasks are processed on the cloud. Even though cloud servers have many computation resources, a large amount of data may lead to network congestion due to limited bandwidth resources. To compensate for the above shortcomings, edge computing [2] has a great advantage in saving transmission bandwidth due to being more distributed and closer to the data source. However, the limited computation resources of edge servers can usually only handle some simple inference tasks, and it is difficult to meet the accuracy requirements of complex inference tasks. When faced with intensive diverse inference tasks, traditional edge-only or cloud-only solutions largely ignore the tradeoff between accuracy, delay, and energy consumption, resulting in significant cost waste.

Recent efforts have developed task allocation mechanisms for edge-cloud collaboration [3, 4]. However, different video tasks may require different resolutions and model versions, while requirements such as accuracy and resource consumption also need to be considered. These factors may lead to uncertainty and complexity in edge-cloud collaboration, thus affecting the efficiency and accuracy of task processing. Traditional solutions have difficulty adapting to diverse requirements for video tasks and dynamic changes in resources, making it challenging to identify the optimal tradeoff between multiple metrics.

In this study, we propose an edge-cloud collaborative inference framework for joint configuration optimization, named FlexInfer. The framework can adaptively adjust the data size and model version according to different task requirements, and decide whether to transfer it to the cloud or the edge for inference. Thus, the total delay and total energy consumption of inference tasks are minimized while meeting the accuracy requirements. For the uncertainty in the edge-cloud collaborative process, an adaptive two-stage robust optimization algorithm is proposed to realize the tradeoff between accuracy and cost under resource constraints.

Problem statement. We consider a scenario in which many inference tasks generated by IoT devices need to be processed simultaneously. Accuracy is usually the primary metric for inference tasks. Therefore, we define the cost function as a weighted sum of delay and energy consumption and take accuracy as a constraint on the optimization problem, which can be expressed as

$$\begin{aligned}
\min \quad & \sum_{i=1}^{M} (D_i + \beta E_i) \\
\text{s.t.} \quad & C_1 : f_i(r, v) \geqslant A_i^q, i \in \{1, 2, \ldots, M\}, \\
& C_2 : y_i = \{0, 1\}, i \in \{1, 2, \ldots, M\}, \\
& C_3 : \sum_{n=1}^{N} x_{i,n} = 1, x_{i,n} = \{0, 1\}, r_n \in \mathcal{R}, \\
& C_4 : \sum_{k=1}^{K} x_{i,k} = 1, x_{i,k} = \{0, 1\}, v_k \in \mathcal{V}, \\
& C_5 : \sum_{i=1}^{M} B_i \leqslant B.
\end{aligned} \tag{1}$$

For the above combined optimization problem, the weight parameter β controls the tradeoff between delay and energy consumption. Constraint C_1 ensures that the accuracy requirements of each task can be met. Otherwise, a better parameter configuration will be assigned until the requirements are met. Constraint C_2 ensures that only one resolution can be selected as input for each task. Constraint C_3 ensures that only one model version can be selected for each task. The fourth constraint C_4 ensures that each task must be transferred to the edge server or cloud server. Constraint C_5 imposes a limit on network bandwidth, where the bandwidth shared by all tasks is less than or equal to the available bandwidth.

Framework design. Considering the complex structure of the optimization problem, the original optimization problem with high complexity can be decomposed into a series of subproblems with lower complexity to solve. Specifically,

* Corresponding author (email: jiwen@ict.ac.cn)

© Science China Press 2024

we transform the above problem into a two-stage robust optimization problem based on edge-cloud collaboration. The first and second-stage decision problems are linear optimization models. Let y be the first-stage decision variable and v be the second-stage decision variable. The uncertainty set $\mathcal{U}$ can be a discrete set or a polyhedron. The two-stage robust optimization problem is expressed as

$$\begin{aligned} &\min_{y} \boldsymbol{c}^{\mathrm{T}} y + \max_{u\in\mathcal{U}} \min_{v\in F(y,u)} \boldsymbol{b}^{\mathrm{T}} v \\ &\text{s.t.} \quad C_1, C_2, C_3, C_4, C_5, \end{aligned} \tag{2}$$

where $F(y,u) = \{\boldsymbol{v} \in \boldsymbol{\mathcal{S}_v} : \boldsymbol{G}\boldsymbol{v} \geqslant \boldsymbol{h} - \boldsymbol{Q}y - \boldsymbol{L}u\}$ denotes the second stage problem, and $\boldsymbol{\mathcal{S}}$ is a polyhedron. $\boldsymbol{c}^{\mathrm{T}}$ denotes the inference task cost matrix with different resolutions and servers, $\boldsymbol{G}$ is the total stage coefficient matrix, and $\boldsymbol{b}^{\mathrm{T}}$ is the inference task cost matrix with different models. y is the optimization variable in the first stage and v in the second stage. $\mathcal{U}$ is the uncertainty set of the objective function.

For the above problem model, we can decompose the original problem into two subproblems with mixed integer linear characteristics through the Benders decomposition algorithm. Since the second-stage decision problem is a linear programming (LP) problem for v, it is feasible for any given y and u. Then let π be its dual variable and merge it with the maximization over u. Finally, the subproblem 1 in the Benders-dual method is obtained

$$\begin{aligned} &\mathcal{SP}_1 : \min_{y} \quad \boldsymbol{c}^{\mathrm{T}} y + \eta \\ &\text{s.t.} \quad C_1, C_2, C_3, \eta \geqslant \left(\boldsymbol{h} - \boldsymbol{Q}y - \boldsymbol{L}\boldsymbol{u}_i^*\right)^{\mathrm{T}} \pi_i^*, \ y \in \mathcal{S}_y, \end{aligned} \tag{3}$$

where $\boldsymbol{Q}$ is the first-stage coefficient matrix, $\boldsymbol{L}$ is the second-stage coefficient matrix, and $\boldsymbol{h}$ is the accuracy demand vector. The configuration parameters of the $\mathcal{SP}_1$ contain the decision variable y in the first stage and the auxiliary variable η. η is mainly used to assess the second-stage objective function values.

However, the result of subproblem 1 is only phased. If only the decision variables and constraints in the first stage are considered, the current optimization result can be regarded as a relaxed version of the whole problem. We need to be further optimized in combination with the subproblem of the second stage. Similar to (3), according to the Benders-dual method, the expression of subproblem 2 is as follows:

$$\begin{aligned} &\mathcal{SP}_2 : \max_{u\in\mathcal{U}} \min_{v\in F(y,u)} \boldsymbol{b}^{\mathrm{T}} v \\ &\text{s.t.} \quad C_4, C_5, \boldsymbol{G}v \geqslant \boldsymbol{h} - \boldsymbol{Q}y - \boldsymbol{L}u, \ v \in \mathcal{S}_v. \end{aligned} \tag{4}$$

We can find that the objective functions and constraints of (3) and (4) can be coupled with each other. It can be proven that the original problem is equivalent to the two subproblems [5]. Then, we propose a column-and-constraint generation robust optimization algorithm to solve them. It first dynamically generates the constraints of decision variables in the original space, and then iteratively optimizes the target value according to the uncertainty set $\mathcal{U}$ and the results of subproblem 1. In each iteration, for each constraint of the $\mathcal{SP}_1$, fix the master variables and solve the corresponding $\mathcal{SP}_2$. The optimal auxiliary variable columns are obtained and added to the set of columns. Then, update the objective function and constraints for $\mathcal{SP}_2$, including the newly added auxiliary variable columns, and increase the number of iterations. Finally, the current solution is enhanced by progressively generating columns of auxiliary variables, leading to an approximate solution for the two-stage robust problem.

Performance evaluation. We perform simulation experiments on one cloud server and four edge servers. Five YOLOv5 models of different sizes are deployed on edge servers and cloud servers, respectively. The bandwidths of cloud servers and edge servers are 100 and 50 Mbps. The fluctuating bandwidth is confined within a range of 30%. Notably, diverse inference request rates exert a profound influence on the overall system performance. To emulate real-world conditions, we transmitted the tasks to the server with request rates set at 20, 30, and 40 for inference, respectively. The experimental results are shown in Figure 1. We find that FlexInfer achieves optimal performance at different inference request rates. When the inference request rate increases, the cost of other methods grows rapidly and the advantages of FlexInfer become more apparent. Our method achieves better results under dynamic network conditions. This is due to its adaptive two-stage robust optimization mechanism. On average, FlexInfer can reduce costs by more than 30% compared to other methods. Overall, our method achieves a tradeoff between accuracy and cost under the above evaluations. This not only highlights the efficacy of our approach but also its adaptability in multiple application scenarios.

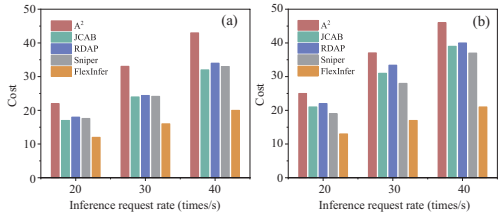

Figure 1 (Color online) Performance comparison of different methods under different inference request rates. (a) Stable bandwidths; (b) fluctuating bandwidths.

Conclusion. In this study, we propose an adaptive edge-cloud collaborative inference framework that can adaptively configure data and model versions according to task requirements, and decide to transfer them to the cloud server or edge server for inference. Considering the complexity of the joint optimization problem, we decompose the original problem into two low-complexity subproblems. We then propose an adaptive two-stage robust optimization algorithm that can optimize the cost of inference tasks under the accuracy constraint. In the future, we plan to study adaptively edge-cloud collaboration strategies based on feature analysis and content preference awareness.

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2023YFB4502805), National Natural Science Foundation of China (Grant No. 62072440), and Beijing Natural Science Foundation (Grant No. L221004).

References

1 Yang Z M, Hu D L, Guo Q, et al. Visual E^{2}C: AI-driven visual end-edge-cloud architecture for 6G in low-carbon smart cities. IEEE Wireless Commun, 2023, 30: 204–210
2 Chen J, Ran X. Deep learning with edge computing: a review. Proc IEEE, 2019, 107: 1655–1674
3 Wang C, Zhang S, Chen Y, et al. Joint configuration adaptation and bandwidth allocation for edge-based realtime video analytics. In: Proceedings of IEEE Conference on Computer Communications, 2020. 257–266
4 Liu S Z, Wang T S, Li J Y, et al. AdaMask: enabling machine-centric video streaming with adaptive frame masking for DNN inference offloading. In: Proceedings of the 30th ACM International Conference on Multimedia, 2022. 3035–3044
5 Zeng B, Zhao L. Solving two-stage robust optimization problems using a column-and-constraint generation method. Oper Res Lett, 2013, 41: 457–461