A Multi-Objective Approach for Optimizing Content Delivery Network System Configuration

Hoang-Loc La*†, Thanh Le Hai Hoang*† and Nam Thoai*†
*High Performance Computing Laboratory, Advanced Institute of Interdisciplinary Science and Technology,
Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT),
268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
†Vietnam National University Ho Chi Minh City,
Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam
Email: {loclh249, thanhhoang, namthoai}@hcmut.edu.vn

***Abstract*—Optimizing the Content Delivery Network system configuration has been addressed as an interesting problem for the system owners. They want to minimize the investment cost while guaranteeing their system's quality. Several works have resolved this problem as a single-objective optimization (SOO) problem with heuristic methods. These approaches usually aggregate the objectives into a scalar function and resolve the problem with SOO algorithms. A typical drawback of these approaches is that they cannot capture the trade-off between the objectives, which usually leads to a sub-optimal solution. To overcome this drawback, this paper considers the problem as a discrete multi-objective problem and resolves it with meta-heuristic techniques, namely Bayesian optimization (BO) and evolutionary methods. More importantly, we also propose an empirical method to improve the convergence speed of the standard BO methods in discrete space. Our experiments show that our proposed method can dramatically improve the rate of convergence. Moreover, we apply our method to a real CDN system and compare our solution with the system's current solution. Our experimental results show that our proposed solution can save about 39% of the current cost with the same internal traffic.**

***Index Terms*—Content Delivery Network, Bayesian Optimization, Genetic Optimization, Multi Objective Optimization**

I. Introduction

Content Delivery Networks (CDNs) have gained a lot of attention from academia and enterprise in recent years. The rapidly increasing usage of Over-The-Top (OTT), Video-on-Demand (VoD), and media streaming services put pressure on the conventional hosting schema, in which a content server or data center handles all user requests. On the other hand, CDNs offload traffic from content servers by responding to end-user requests in place of the content servers and in closer physical and network proximity to the end-users. Therefore, CDN systems can help the content providers to guarantee their service's latency and quality.

However, the trade-off between cost and service quality has been addressed as a big challenge for CDN providers. They want to reduce the investment cost but still guarantee or maximize the service quality at the same time. This paper describes a general form of CDN configuration optimization problem. We consider the configuration problem as a discrete multi-objective optimization (MOO) problem. The problem contains two conflicting objectives and takes discrete variables as input. After that, we also consider a practical instance of our general problem and resolve it by applying MOO algorithms. Furthermore, our quality function is based on simulation environments, which can be very time-consuming to evaluate. The BO approach is a well-known method for optimizing expensive black-box functions. For these reasons, we choose Bayesian methods as our approach. In this paper, we adopt the multi-objective Bayesian optimization (MOBO) approach, which is originally designed for continuous space, to optimize more effectively in discrete space.

Corresponding authors: Hoang-Loc La, Nam Thoai.

The remaining sections are organized as follows. Section II describes related works. Section III reminds background about the considered MOBO algorithms. Section IV describes the general problem and a practical use case. Section V describes our proposed method to resolve the problem. The next section is our experimental results. Finally, section VII consists of concluding remarks and our future work.

II. Related Work

There were several Bayesian-based MOO methods, namely ParEGO [1], USeMO [2], DGEMO [3], TSEMO [4], etc. The above BO approaches use Gaussian Process (GP) as their surrogate model. The original design of GP model is applied to continuous space. The issues of applying single-objective BO methods in discrete space were described and analyzed in [5]. To adapt these algorithms for discrete problems, a *basic* approach is that resolving the problems similar to continuous problems, but the result of the evaluation functions is rounded to the closest integer. A typical drawback of this approach is that GP can ignore behaviors of the actual function. Merchán et al. [5] proposed the kernel-based approach to overcome this issue. Especially, they applied a *transformation* function to round the input of the Matérn kernel function. However, they only applied and experimented with these methods in single-objective problems. In another approach, Luong et al. [6] proposed a hyper-parameter tuning strategy to improve the trade-off between exploration and exploitation when applying BO algorithms in discrete SOO problems. This paper proposes an empirical method that motivated by Luong's method to improve the convergence speed of MOO algorithms.

III. Background

In this paper, we will consider two state-of-the-art MOBO methods, namely TSEMO and USeMO. Assume we have a k-objective optimization problem. In general, the methods are iterative and based on four major steps:

- Step 1: The statistical models M_1, M_2,..., M_k for each of the k objective functions are trained with the historical data from the initial samples and the past iterations. These models are built based on GP. Moreover, we choose Matérn function [7] as the GP's kernel. The hyper-parameters of Gaussian Process is optimized by log marginal likelihood method [8].
- Step 2: Instead of solving the expensive original MOO problem, the methods will build a cheap MOO problem and apply an MOEA to find the *Pareto* set [2]. The cheap MOO problem corresponds to $AF(M_1)$, $AF(M_2)$,..., $AF(M_k)$ objective functions. We denote $AF(.)$ as any acquisition function of the standard single-objective BO algorithm. In this paper, we evaluate USeMO with Upper Confidence Bound (UCB) [9] and Expected Improvement (EI) [10] as the acquisition functions. Both TSEMO and USeMO use NSGA-II to optimize the cheap problem.
- Step 3: After solving the cheap problem, a candidate set x_s is chosen from the *Pareto* set by heuristic strategies [2], [4].
- Step 4: The training set of the GP model is updated by adding the new candidate set x_s. The process is back to step 1.

IV. Problem Description

A. General problem

We consider the CDN configuration optimization problem as a discrete multi-objective optimization problem. Our target is to minimize the investment cost f_{cost} and maximize the system quality function $f_{quality}$ at the same time. These functions take X as an input vector. The input X of the problem can be CDN configurations as: memory sizes of caching servers, the bandwidth of network links, etc. These configuration parameters are usually integer-based or categorical variables. Therefore, we assume that X is discrete. $f_{quality}$ can be any quality metrics as QoS, internal traffic, average latency, jitter, etc. f_{cost} is used to measure the investment cost for the CDN system. The next section will be a practical instance of the general problem.

B. Memory Allocation for Surrogate Nodes

We consider a CDN system that contains N surrogate nodes. The problem goal is to find an optimal memory size configuration for all surrogate nodes, which minimizes the hardware cost and reduces the internal traffic $f_{traffic}$ simultaneously. The problem takes $x_i, \forall i \in \{0, ..., N\}$ as input variables. In this problem, minimizing $f_{traffic}$ is equivalent to maximizing $f_{quality}$. The f_{cost} function is simply the total memory of system, $f_{cost} = \sum x_i$. To simplify the problem, all surrogate nodes use Least-Recently-Unit (LRU) algorithm as their caching eviction method. Moreover, resolving this problem by using a brute-force approach is infeasible.

V. Solutions

The original Bayesian algorithms resolve optimization problems in continuous space. This paper adapts a batch version of TSEMO and USeMO to resolve the problem [3]. To initialize the population in the discrete space, we use a modified version of Latin Hypercube Sampling (LHS) [11].

Luong et al. [6] proposed a method to help the standard BO methods avoid the repetition of observations in discrete space. Their method try to balance between exploration and exploitation by turning the parameters of acquisition function and the kernel function. A drawback of this method is that it requires resolving an optimization problem to find an optimal set of factors, which is cumbersome and very expensive in MOO problems. Tuning all of these factors can be a complex optimization problem and consume a lot of time.

In this paper, we consider UCB acquisition function. We denote $\mu(x)$ and $\sigma(x)$, respectively, as mean and variance of the surrogate model's posterior distribution. The exploration factor is denoted as β. The UCB acquisition function is formalized as follow:

$$\alpha^{UCB}(x) = \mu(x) + \sqrt{\beta} \times \sigma(x)$$

To reduce the fine-tuning time, we only tune the β factor. We consider a MOBO algorithm is stuck in a local optimum when the Pareto fronts of two adjacent iterations are has the same pattern and very close to each other. Moreover, we use the bidirectional Hausdorff distance to evaluate the similarity between Pareto fronts of adjacent iterations [12]. Formally, we determine the algorithm get stuck in a local optimum when the distance between two corresponding Pareto fronts is less than ξ_{pf}. The β factor can be determined by resolving the following optimization problem:

$$\begin{aligned} \beta_t^* &= \ argmin(g(\beta_t)), \beta_t \in [0, 1] \\ g(\beta_t) &= \ (\beta_t - \beta_t^0) - d(PF_t, PF_t^0) \end{aligned} \tag{1}$$

PF_t and PF_t^0 are Pareto fronts, which are suggested by the adjusted β_t and the original β_t^0, respectively. Remarkably, instead of computing PF_t based on the black-box function, which is very expensive, we use the trained surrogate model to evaluate this value. There are two objectives for the above problem as:

- Maximize the difference between Pareto fronts generated by two adjacent iterations.
- Minimize the adjustment of the β factor.

We use L-BFGS-B [13] optimizer to resolve this problem. **Algorithm 1** describes the details of our proposed method. Remarkably, our proposed idea can be extrapolated to other acquisition functions.

VI. Experiments

A. Environment

We consider three topologies of CDN system to evaluate the convergence of MOBO algorithms with different sizes of the problem:

- A small size topology of the Vietnam system, which has 5 variables that corresponding to 5 caching nodes. [14]
- A medium-size topology of a France telecommunication company, which has 13 variables. [15]
- A big-size topology of a Japan telecommunication company, which has 55 variables. [16]

Figure 1 illustrates the topology of the CDN networks. This section considers two types of datasets, such as a real dataset and a simulated dataset. We run ten executions for each experiment with 30 iterations and a batch size of 10.

Algorithm 1: The proposed method for tuning β factor

Input: Initial data $D_0 = (x_0, y_0)$;
for $t \leftarrow 0$ **to** n **do**
  /* k is the size of D_t */
  β_t^0 is computed as suggested in [9] ;
  Compute PF_t^0 by using the surrogate model ;
  if $\{Distance(PF_t, PF_t^0) < \xi_{pf}\}$ **then**
    Find the optimal β^* using (1), with $0 < \beta < 1$;
    Select the next sample x_{t+1} with β^* ;
  else
    Select the next sample x_{t+1} with β_t^0 ;
  Compute y_{t+1} obtained by the black-box function;
  Augment $D_{t+1} = \{D_t, (x_{t+1}, y_{t+1})\}$ and update the surrogate model.

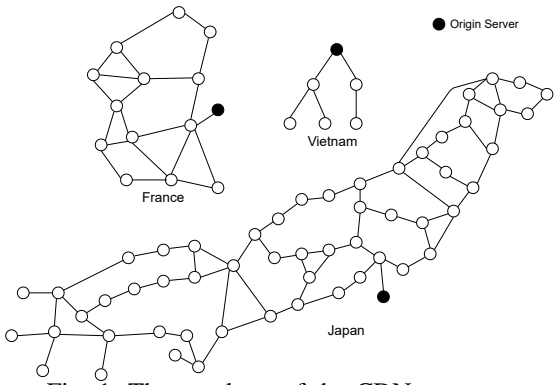

Fig. 1: The topology of the CDN systems.

Regarding to the proposed algorithm, we configure $\xi_{pf} = 0.001$. The NSGA-II algorithm uses uniform mutation operator and single-point crossover operator with a probability of 0.9. We run this evolutionary algorithm with 100 initial samples and 30 generations.

B. Results & Discussion

1) Experiments with simulated data

TABLE I: ENVIRONMENT SETUP FOR THE SIMULATED DATA EXPERIMENTS

Number of contents	500
Warm up	1000 requests
Evaluation	1000 requests
Range of caching memory	[50, 450]
Content popularity	$Gamma(k = 0.475, \theta = 170.6067)$

Table I shows the setup for the simulated data experiments. Figure 2 depicts the evolution of the average hypervolume value and the Pareto front of the last iteration computed by each algorithm in three types of CDN networks. At first glance, our proposed method outperforms other algorithms in all three topologies.

Furthermore, USeMO-UCB is better than USeMO-EI in all of the experiments. This observation is reasonable because we use the standard EI function, which is widely known to be too greedy. In another way, the UCB acquisition function contains the β exploration factor, which will be tuned during the optimization process. This mechanism will help UCB to balance exploration and exploitation better than the standard EI.

2) Experiments with the real data

This section leverages the real trace log from the Vietnamese CDN system to compare our proposed algorithm and the current real system configuration. Particularly, we only get the system trace log in the peak hour of a day to run experiments. Importantly, the search range of input variables is from 25% to 400% of the real memory size. Running optimization on this workload can consume a lot of time. Therefore, we only consider original USeMO-UCB algorithm, its *transformation* version and our proposed method.

Figure 3 illustrates the Pareto front of three versions of USeMO-UCB algorithm at the last iteration. Especially, our proposed method still outperforms the *basic* and *transformation* versions of the USeMO-UCB algorithm. More importantly, the Pareto points of all the considered algorithms are dominated the current solution of the real system. Therefore, we can use our proposed solutions to optimize the current system. Figure 4 depicts the trade-off between the cost-saving and the relative traffic of our solution. In particular, our solution can save nearly 39% of the current cost with the same internal traffic of the system.

VII. Conclusions and Future Work

In this work, we have modeled the general framework of the CDN configuration optimization problem. The framework can be applied to several types of configuration problems: caching replicas, topology optimization, bandwidth allocations, etc. An instance of the general problem is described and resolved by the MOBO algorithms. Moreover, we also propose a tuning method to adapt the exploration factor of the acquisition function. The experiments show that our method can improve the convergence of MOBO algorithms. In the future, we will apply the general framework with more sensitive quality metrics like latency, jitter, etc. Although, reliable emulators or test-beds can evaluate these metrics, these tools usually consume a significant amount of time for each evaluation. Therefore, a lightweight and more effective optimization algorithm is vital to optimizing based on these tools.

Acknowledgment

This research was conducted within the project of Emulation of the color-based caching scheme in Telco-CDNs with Mininet using real data sponsored by TIS (IT Holding Group).

We acknowledge the support of time and facilities from High Performance Computing Laboratory, Ho Chi Minh City University of Technology, VNU-HCM for this study.

Thanh Hoang Le Hai was funded by Vingroup Joint Stock Company and supported by the Domestic Master/Ph.D. Scholarship Programme of Vingroup Innovation Foundation (VINIF), Vingroup Big Data Institute (VINBIGDATA), code VINIF.2020.ThS.24.

References

[1] J. Knowles, "Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.

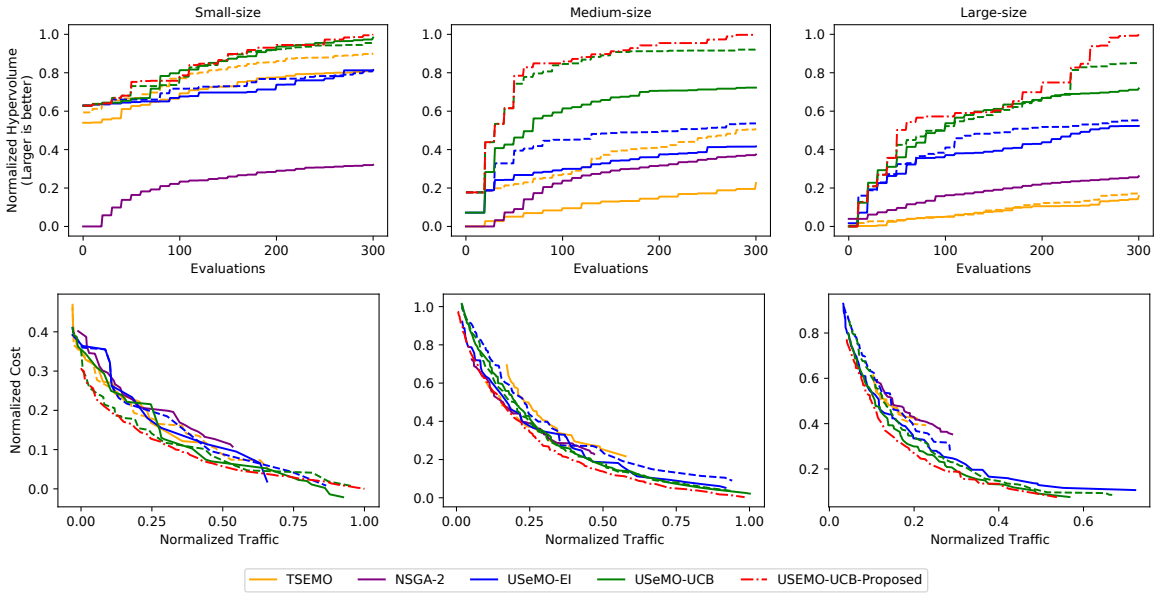

Fig. 2: *Top:* Normalized Hypervolume Indicator between algorithms in 3 types of CDN networks. *Bottom:* Pareto fronts of algorithms at the last iteration in 3 types of CDN network. The solid and dashed lines of each color, respectively, are the *basic* and the *transformation* version of the same algorithm. The red dash-dot lines are our proposed method.

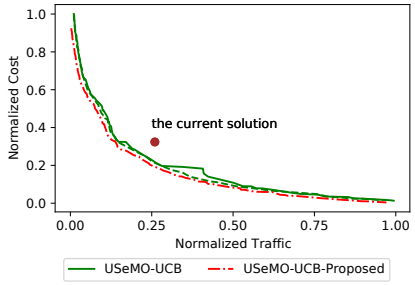

Fig. 3: The Pareto fronts of three versions of USeMO-UCB at the last iteration. The circle point is the current solution of the real system.

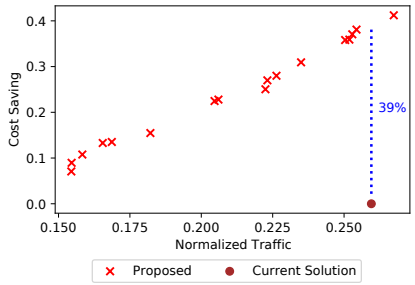

Fig. 4: The trade-off between cost saving and relative traffic of our proposed solution.

[2] S. Belakaria, A. Deshwal, N. Kannappan Jayakodi, and J. Doppa, "Uncertainty-aware search framework for multi-objective bayesian optimization," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 10 044–10 052, 04 2020.

[3] M. Konakovic-Lukovic, Y. Tian, and W. Matusik, "Diversity-guided multi-objective bayesian optimization with batch evaluations," in *NeurIPS*, 2020.

[4] E. Bradford, A. Schweidtmann, and A. Lapkin, "Efficient multiobjective optimization employing gaussian processes, spectral sampling and a genetic algorithm," *Journal of Global Optimization*, vol. 71, 06 2018.

[5] E. Garrido-Merchán and D. Hernández-Lobato, "Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes," 05 2018.

[6] P. Luong, S. Gupta, D. Nguyen, S. Rana, and S. Venkatesh, *Bayesian Optimization with Discrete Variables*, 12 2019, pp. 473–484.

[7] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[8] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning.*, ser. Adaptive computation and machine learning. MIT Press, 2006.

[9] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," 07 2010, pp. 1015–1022.

[10] J. Mockus, "On bayesian methods for seeking the extremum," in *Proceedings of the IFIP Technical Conference*. Berlin, Heidelberg: Springer-Verlag, 1974, p. 400–404.

[11] C. Maschio and D. Schiozer, "Probabilistic history matching using discrete latin hypercube sampling and nonparametric density estimation," *Journal of Petroleum Science and Engineering*, vol. 147, 05 2016.

[12] R. Rockafellar and R. J.-B. Wets, *Variational Analysis*. Heidelberg, Berlin, New York: Springer Verlag, 1998.

[13] D. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1-3, pp. 503–528, Aug. 1989, copyright: Copyright 2007 Elsevier B.V., All rights reserved.

[14] H.-L. La, A.-T. N. Tran, Q.-T. Le, M. Yoshimi, T. Nakajima, and N. Thoai, "A use case of content delivery network raw log file analysis," in *2020 International Conference on Advanced Computing and Applications (ACOMP)*, 2020, pp. 71–78.

[15] Z. Li and G. Simon, "In a telco-cdn, pushing content makes sense," *Network and Service Management, IEEE Transactions on*, vol. 10, pp. 300–311, 09 2013.

[16] T. Nakajima, M. Yoshimi, C. Wu, and T. Yoshinaga, "A light-weight content distribution scheme for cooperative caching in telco-cdns," in *2016 Fourth International Symposium on Computing and Networking (CANDAR)*. IEEE, 2016, pp. 126–132.