Received May 26, 2021, accepted June 7, 2021, date of publication June 23, 2021, date of current version July 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3091693

Joint Optimization for Bandwidth Utilization and Delay Based on Particle Swarm Optimization

XU LIU[1], ZHENHAO LI[2], PENG XU[2], AND JIALING LI[2]

[1]China Academy of Industrial Internet, Beijing 100102, China
[2]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Peng Xu (xupeng@bupt.edu.cn)

This work was supported by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61921003.

ABSTRACT One of the most significant problems in large-scale applications is how to allocate requests from end-users to datacenters. Most existing studies focus only on high bandwidth utilization of datacenters or low request delay of end-users, lacking the consideration of both at the same time. To achieve the joint optimization for bandwidth utilization of datacenter providers and delay of end-users, this paper proposes a novel Particle Swarm Optimization (PSO) based request allocation algorithm, based on the construction of a mathematical model for joint optimization. The algorithm adapts the coding scheme and redefines the operator to make PSO suitable for discrete optimization problems and use fitness function to get a global optimal solution. Through extensive simulations, it is demonstrated that the PSO based request allocation algorithm proposed in this study can improve the bandwidth utilization and keep the average request delay within acceptable limits or even better.

INDEX TERMS Bandwidth utilization, joint optimization, large-scale applications, particle swarm optimization, request allocation, request delay.

I. INTRODUCTION

Recently, large-scale applications have gained significant popularity due to the increasing number of users. In large-scale applications, such as social network, education network and web search, they are always deployed geographically distributed datacenters for both reliability and better performance. Under the scenario that multiple datacenters jointly provide services, one of the most significant challenges is how to efficiently allocate different users' requests to different datacenters in order to improve bandwidth utilization or reduce request delay. This study refers to this kind of problem as the request allocation problem.

Request allocation has obtained a lot of concentrations from the research groups in the last few years. There are two factors needing to be concentrated on in this problem. On the one hand, request allocation aims to improve bandwidth resource utilization of datacenters. Different datacenters may reach their peak workload at different times. Some datacenters may reach peak workload while others have low bandwidth utilization at a moment. In this way, overloaded datacenters are more vulnerable to failures and datacenters with low bandwidth utilization are a waste of energy and investment. On the other hand, large-scale applications provide service for many users so that it should reduce the request delay. With the common localized allocated strategy, users' requests are always sent to closet datacenter in order to decrease the transport delay. However, it causes high delay when datacenters reach peak workload.

Most of prior studies have been done and they focused on improving bandwidth resource utilization of datacenters like using store and forward schemes [1] avoiding peak time of target datacenter and using forwarding tree cohorts by fair sharing strategy [2]. Other studies have focused on reducing end-user request delay, such as using Heterogeneous Fair Resource Allocation and Scheduling (HFRAS) algorithms to achieve delay reduction [3]. Or the combination of workload allocation among different servers and delay-base workload allocation (DBWA) algorithm is used to achieve the goal of delay reduction [4]. Few approaches consider both improving bandwidth utilization and reducing request delay. This is due to the fact that low delay and high bandwidth utilization are the opposite of each other. Users close to the datacenters experience low delay. Once the bandwidth

The associate editor coordinating the review of this manuscript and approving it for publication was Muhamamd Aleem.

utilization increases, users far away from the data centers are likely to suffer high latency. Therefore, in this study, the aim is to find a better balance between the two mentioned above. Requests' different transport delay between end-user and datacenter, combined with the heterogeneity of physical resources, makes it hard to take into account both request delay and bandwidth utilization. Classic scheduling algorithms like "greedy algorithm (GA) [5]" and "locality algorithm (LA) [6]" are also not suitable to solve this problem due to that only a single factor can be considered. In fact, considering the NP-hard feature of this problem and the practical online request allocation context, it faces a dilemma of getting a sub-optimal solution when combining bandwidth utilization for datacenter providers and request delay for end-users. To address this problem, meta-heuristics has been widely used in resource allocation problems due to its smart iteration process. Request allocation is a kind of resource allocation problem. Common meta-heuristics include Ant Colony Optimization [7], Genetic Algorithm [8], Particle Swarm Optimization Algorithm [9], Simulated Annealing Algorithm [10], etc. Among many heuristics, this paper finally selects particle swarm optimization for the advantages below.

- PSO has strong robustness and thereby it can be widely used in many fields, such as artificial neural network training [11] and virtual machine allocation [12].
- PSO has faster execution speed and higher efficiency of problem-solving. Based on this feature, [13] proposes a triple archives PSO (TAPSO) and [14] proposes an eXpanded PSO (XPSO).
- For all particles in PSO, they can search solutions concurrently, which can be executed in parallel. Reference [15] proposes a PSO variant based on multiple adaptative strategies (MAPSO) algorithm based on this feature.
- PSO can be easy to implement and there are few parameters to be adjusted.

This paper studies the request allocation between end-users and provider, aiming at increasing bandwidth utilization while reducing users' request delay.

However, there are many challenges to be addressed when leveraging the particle swarm optimization technique in request allocation:

- How to make joint optimization for bandwidth utilization and request delay.
- How to make particle swarm optimization suitable for discrete optimization problem.
- How to determine the value of each parameter of particle swarm optimization when updating speed of particle.

To address the first challenge, this study first constructs a mathematical model to describe the problem where each feasible solution is represented by two matrixes. One represents bandwidth utilization of datacenters. The other represents delay of end-users. Then this study combines the requirement of high bandwidth utilization and low request delay to quantitatively evaluate each solution.

To address the second challenge, this study adjusts the coding scheme and redefine the operator in particle swarm optimization. In this way, the PSO is suitable for discrete optimization problem.

To address the third challenge, this study determines the value of each parameter when updating speed of particle by using fitness function which considers bandwidth utilization and request delay. And it determines the value of fitness function using global best solution, local best solution and current solution.

Next, the key contributions to this paper as follows:

- This paper proposes a request allocation algorithm based on particle swarm optimization to achieve joint optimization of bandwidth utilization and request delay.
- In the algorithm, this paper redefines coding schemas, parameters and operators to make PSO suitable for discrete optimization problem. This paper introduces the fitness function which considers bandwidth utilization and request delay to determine the value of parameters in PSO when updating speed of particle.
- Finally, an extensive comparison of the algorithm proposed in this paper, the LA and the GA is presented to demonstrate that the algorithm proposed in this paper can achieve joint optimization of bandwidth utilization and average request delay.

The outline of this paper is listed as follows. Section II describes and models the problem, puts forward a method to evaluate each solution of this problem. And Section III presents our algorithm based on particle swarm optimization. The experiment setup, results and analysis are presented in Section IV. Section V reviews and discusses the related works. In the end, Section VI makes a conclusion of this paper.

II. PROBLEM DEFINITION

A. PROBLEM DESCRIPTION

Traditionally, request allocation is handled by deploying mapping nodes, which are typically HTTP or authoritative ingress proxies that route user requests from a given locale to the appropriate datacenter, or DNS servers that resolve the same name of Web sites. In real application, with the success of Software Defined Network (SDN), requests can be easily handled by the SDN controller. The information of each datacenter in the network can be gathered by the SDN controller. In this way, massive requests come to the controller, the controller can make a more reasonable request allocation strategy by using global information. In real system architecture, multiple SDN controllers act as gateways to manage multiple data centers simultaneously to achieve load balance. This study simplifies this practical system model and constructs the multiple datacenters model based on the SDN controller just like the model in the study [16]. The whole model consists of two parts. One is a set $D = 1, 2, \ldots, d$ of datacenters which are distributed in different regions for better reliability, the other is a set $U = 1, 2, \ldots, u$ of end-users. C_{D_j} is the fixed bandwidth capacity of datacenter

TABLE 1. Notations.

Notation	Description
D	the set of datacenters
U	the set of end-users
D_j	the jth datacenter, j=1,2,...
N_k	the number of requests of kth end-user, k=1,2,...
U_k	the kth end-user, k=1,2,...
U_{k_i}	the ith request in kth end-user, i=1,2,....N
C_{D_j}	the bandwidth capacity of datacenter j
B_{D_j}	the amount of bandwidth occupied by a request
$S_{U_{k_i}D_j}$	whether end-user k requests i to j
P_{D_j}	the bandwidth utilization of j
T_{U_k}	the average latency experienced by user k
θ	the random waiting time in queue
$L_{U_kD_j}$	the transport delay between datacenter j and end-user k
$\varphi_{U_kD_j}$	the process delay in datacenter j
$\sigma_{U_kD_j}$	the sum delay for a request

j and B_{D_j} is the amount of bandwidth occupied by a request. This paper assumes each end-user can only send one request at a given moment. Then it needs to find a mapping from end-users requests to datacenters. Let $S_{U_kD_j}$ indicate whether end-user k sends a request to datacenter j.

$$S_{U_kD_j} = \begin{cases} 1, & if\ k\ request\ to\ j \\ 0, & otherwise. \end{cases} \tag{1}$$

B. PERFORMANCE METRICS

The multiple datacenters model contains two roles: the provider and end-users. In order to maximize profits, provider tends to maximize the bandwidth utilization and users want to get low latency experience for each request. This study needs to evaluate the quality of each request allocation solution, with the aim of achieving a balance between two metrics. The following discussed two metrics in detail and applied the Nash bargaining solution [17] to achieve the goal.

1) HIGH BANDWIDTH UTILIZATION

Since the provider contains M datacenters, the goal for this study is to find a strategy which can maximize the bandwidth utilization. Under normal conditions, the massive requests from users always cause the different workload among datacenters. An efficient request allocation can shift some workload from overloaded datacenter to datacenter with low bandwidth utilization. In this way, this study can maximize the bandwidth utilization. To achieve the goal, this study proposes the definition of bandwidth utilization to evaluate each datacenter workload as follows:

$$P_{D_j} = \frac{\sum_{k=1}^{U}\sum_{i=1}^{N_k} S_{U_{k_i}D_j}B_{D_j}}{C_{D_j}}, \tag{2}$$

where $S_{U_kD_j}$ represents the whether end-user k requests datacenter j, B_{D_j} represents the amount of bandwidth occupied by a request and C_{D_j} represents the bandwidth capacity of datacenter j. To evaluate the whole bandwidth utilization, this study can relate the entire multiple datacenters model to a bargaining market. Each datacenter is viewed as a player and bandwidth utilization is the only metric for the player. In this Nash bargaining game, high bandwidth utilization can be achieved at all datacenters as follows:

$$aB_{max} = \prod_{j=1}^{D} P_{D_j}. \tag{3}$$

2) LOW DELAY

Delay is the most important metric for end-users. In order to retain users, many large companies make lots of efforts to reduce the delay time. Amazon reports that every 100ms delay in page load time decreases sales by 1 percent [18]. In this paper, the delay contains three parts. The first part is the transport delay between end-user and datacenter. The second part is the process time of each request. The third part is a random waiting time in queue inside the datacenter. Therefore, the definition of delay for each request as follows:

$$\sigma_{U_kD_j} = L_{U_kD_j} + \varphi_{U_kD_j} + \theta, \tag{4}$$

where $L(U_kD_j)$ represents the transport delay and $\varphi(U_kD_j)$ represents the process time. In real world, applications can only process several requests at a given moment. Then the next request must wait in the queue. To simulate such an environment, it assumes that waiting time satisfies the uniform distribution. Therefore, it is easy to express the interval using: $[u_j*(1-\delta), u_j*(1+\delta)]$ where u_j denotes the average process time for every request. δ denotes the standard deviation in uniform distribution. Therefore, this study sets θ as mean value. This study proposes the definition of delay for end-user as follows:

$$T_{U_k} = \frac{\sum_{k=1}^{U}\sum_{i=1}^{N_k} S_{U_{k_i}D_j}\sigma_{U_kD_j}}{N_k}, \tag{5}$$

where T_{U_k} represents the average latency experienced by user k. Just as the method evaluating bandwidth utilization, it can also view end-users as players and low delay is the only metric for the player. In this Nash bargaining game, it can achieve low delay at all end-users as follows:

$$aD_{min} = \prod_{k=1}^{U} T_{U_k}. \tag{6}$$

Given the above models, it can combine the aB_{max} and aD_{min} as follows:

$$aJ_{max} = aB_{max} + \frac{1}{aD_{min}}. \tag{7}$$

According to the definition of aJ_{max}, a larger aJ_{max} means a better feasible request allocation solution. This study would like to get a solution with maximized aJ_{max}.

C. EVALUATION METRICS

To better evaluate the efficiency of Particle swarm optimization (PSO), Greedy algorithm (GA) and Locality algorithm (LA) of improving bandwidth utilization and reducing

request delay, this report proposes the following formula for the overall evaluation of the algorithms:

$$f(x, y) = \alpha|\frac{x - x_{min}}{x_{max} - x_{min}}| + \beta|\frac{\frac{1}{y} - y_{min}}{y_{max} - y_{min}}| \quad (8)$$

where x represents bandwidth utilization, and y represents delay at the same moment. α and β denote weights, which are taken as 0.5 and 0.5, respectively, in this formula depending on the actual situation. α and β may be assigned different weights in other cases, and this paper will not discuss these cases in detail here due to space limitations. x_{min} and x_{max} are the minimum and maximum value of x. y_{min} and y_{max} are the minimum and maximum value of y.

III. METHOD

A. REQUEST ALLOCATION BASED ON PARTICLE SWAM OPTIMIZATION

PSO is a stochastic based meta heuristic computational method inspired by the social behavior in animals, such as fish schooling or bird flocking [19]. The PSO algorithm is proposed by Dr. Eberhart and Dr. Kennedy, from the behavior of bird predation. Initially, all particles are randomly initialized in search space and a single particle represents a feasible solution in specific problems. In the algorithm, particle contains two properties, namely particle position and particle velocity. Particle position represents the current feasible solution and velocity is used to update position in order to find the optimal solution. In PSO, particle needs to store global best solution and local best solution. Particle velocity is updated by recorded solutions as follows:

$$V_{id}(t+1)=V_{id}(t)+C_1r_{1d}(P_{id}-X_{id})+C_2r_{2d}(P_{gd}-X_{id}), \quad (9)$$

where $C1$ and $C2$ are two positive numbers called acceleration coefficients and r_{1d} and r_{2d} are two uniformly distributed numbers in interval [0, 1]. P_{id} and P_{gd} denotes the local Best and global Best solution of a particle. When velocity has been updated, position of a particle can also be calculated as follows:

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1), \quad (10)$$

where $X_{id}(t+1)$ is the next position of a particle.

This study uses PSO to solve the request allocation problem. As mentioned above, PSO has many advantages in solving NP-hard problem such as easy to implement and few parameters need to be adjusted, however there are some challenges needing to be addressed.

- To make PSO suitable for discrete optimization problem, this study redefines the coding schema, parameter and operator in PSO.
- To determine the value of each parameter of PSO when updating, this study introduces the fitness function which considers bandwidth utilization and request delay.

Then aJ_{max} is used to measure each solution in order to get local best and global best solution.

1) CODING SCHEMA

PSO is suitable for continuous problems. In order to make PSO suitable for discrete optimization problem, this study redefines the coding schema. A two-dimensional encoding schema is quite useful for solving the request allocation problem and this study uses two kinds of encoding schema called datacenter schema which is shown in Fig. 1 and end-user schema which is shown in Fig. 2. The first dimension of datacenter schema is an n-bit vector $(x_1, x_2, \ldots, x_n)$ where n is the number of datacenters and the second dimension contains the set of requests which come from different end-users using x_n^t to represent. In the first dimension, every bit is associated with a datacenter. If the bit is $\iota^{\circledR}1\iota^{-}$ it means that the datacenter has processed requests. If the bit is $\iota^{\circledR}0\iota^{-}$ then it means that the datacenter has never received any requests. Besides this, this study converts datacenter schema to end-user schema. The first dimension of end-user schema is an n-bit vector where n is the number of end-users and the second dimension represents the relationship between datacenters and end-users.

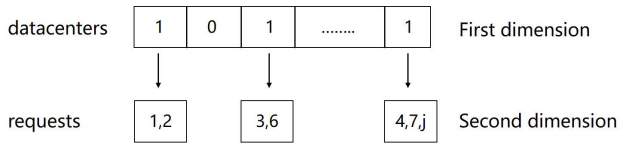

FIGURE 1. The datacenter schema.

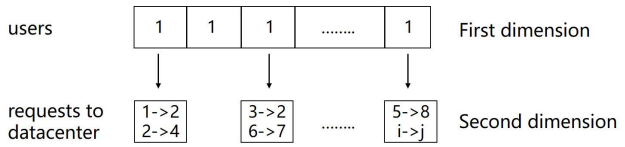

FIGURE 2. The user schema.

2) PARAMETER AND OPERATOR IN PSO

Parameters in PSO contain position and velocity and operators contain subtraction, addition and multiplication. The details are presented as below:

Position: According to the datacenter encoding schema, this study uses a bit vector $X^t = (X_1^t, X_2^t, \ldots, X_n^t)$ to represent the particle's position. n represents the number of datacenters and x_n^t represents a single datacenter which processes requests from different end-users. X^t represents a feasible solution for request allocation problem.

Velocity: This study uses a bit vector $V^t = (V_1^t, V_2^t, \ldots, V_n^t)$ to represent the particle's velocity. n represents the number of datacenters and V_n^t indicates whether the request on the corresponding datacenter should be redistributed. If V_n^t is $\iota^{\circledR}0\iota^{-}$, then requests on the corresponding datacenter should be redistributed. If V_n^t is $\iota^{\circledR}1\iota^{-}$, then there will be no change.

Subtraction Operator: The subtraction operator is used to calculate the difference between the two feasible request allocation solutions and this study uses symbol $\ominus$ to represent the subtraction operator. As mentioned above, the value of X_n^t

is '1' or '0'. Therefore, given two positions X_i^t and X_j^t, if the value of bits in vector are same, then subtraction result will be 1, otherwise 0. For example (1,0,1,0) $\ominus$ (1,1,0,0) = (1,0,0,1).

Addition Operator: The addition operator is used to update the particleι^-s velocity and this study uses symbol $\oplus$ to represent the addition operator. There are three factors in addition operator which are current velocity, local best position and global best position. If there are n datacenters, then $(P_1V_1 \oplus P_2V_2 \oplus \ldots P_nV_n)$ denotes that a particle updates its velocity by using V_1 with a probability P_1, using V_n with a probability P_n and so on. For example, 0.2(1,0,1,0) $\oplus$ 0.8(1,0,0,1) = (1,0,?,?). If the value of bits is same, then the addition result is not changed. Otherwise, for the third bit, the addition result has possibility 0.2 to be 1 and 0.8 to be 0. For the fourth bit, the addition result has possibility 0.2 to be 0 and 0.8 to be 1. Thus, parameters shown in equation 9 would be replaced by the possibility P_1, P_2 and P_3 and the value of possibility can be calculated by fitness function. Therefore, in addition operator, the uncertain bits value can be calculated as below:

$$S_{U_kD_j} = \begin{cases} q1, & if\ rand \le P_1 \\ q2, & if\ P_1 < rand \le P_2 \\ q3, & if\ P_1 + P_2 < rand \le 1, \end{cases} \tag{11}$$

where *rand* is a random number generated in range 0.0 to 1.0. $q1$, $q2$ and $q3$ are one of the bits' values in corresponding particle's position.

Multiplication Operator: The multiplication operator is used to update the particle's next position and this study uses symbol $\otimes$ to represent the multiplication operator. $X_i^t \otimes V_j^{t+1}$, represents the current position update based on the velocity vector V_j^{t+1}. If the bit's value in V_j^{t+1} is 1, then the corresponding bit in X_i^t would not be changed. Otherwise, the requests processed by the corresponding datacenter need to be allocated again. In this way, it can get a better solution.

3) UPDATING STRATEGY

In our PSO based request allocation algorithm, it updates particle's position and velocity as below:

$$V_i^{t+1} = P_1V_i^t \oplus P_2(X_{lb} \ominus X_i^t) \oplus P_3(X_{gb} \ominus X_i^t) \tag{12}$$

$$X_i^{t+1} = X_i^t \otimes V_i^{t+1} \tag{13}$$

As mentioned above, P_1, P_2 and P_3 are obtained by fitness function. The definition of the fitness function as below:

$$f_{c_j} = \frac{\sum_{i=1}^{n} X_{ij}B_i}{T_{c_j}} + \frac{M_j}{\sum_{i=1}^{m} X_{ij}D_i} \tag{14}$$

where X_{ij} denotes whether the i^{th} request has been processed by j^{th} datacenter and T_{c_j} denotes the bandwidth threshold of j^{th} datacenter. B_i represents the amount of bandwidth a request occupies and D_i represents the delay time of the i^{th} request. The fitness function of particle's position considers bandwidth utilization of datacenters and delay time of end-users. Similarly, the fitness function of local best position and global best position can be calculated as below:

$$f_{lb_j} = \frac{\sum_{i=1}^{n} X_{ij}B_i}{T_{lb_j}} + \frac{M_j}{\sum_{i=1}^{m} X_{ij}D_i} \tag{15}$$

$$f_{gb_j} = \frac{\sum_{i=1}^{n} X_{ij}B_i}{T_{gb_j}} + \frac{M_j}{\sum_{i=1}^{m} X_{ij}D_i} \tag{16}$$

Then the value of possibility can be written as:

$$P_1 = \frac{f_{c_j}}{f_{c_j} + f_{lb_j} + f_{gb_j}} \tag{17}$$

$$P_2 = \frac{f_{lb_j}}{f_{c_j} + f_{lb_j} + f_{gb_j}} \tag{18}$$

$$P_3 = \frac{f_{gb_j}}{f_{c_j} + f_{lb_j} + f_{gb_j}} \tag{19}$$

In addition operator, if it always selects maximum P, it will result in that the algorithm may fall into local optima quickly. To address this issue, this study introduces a rand number to get optimal solution. As the addition operator mentioned, the uncertain bits' value can be calculated using equation 11.

In this way, this study can update particle's velocity in each iterator. And as the multiplication operator mentioned, if the bit value of particle's velocity V_i^{t+1} is '0', the corresponding bit value of particle's position needs to be reconsidered. The following strategy is used to reallocate requests processed by the corresponding data center. First step is to generate a random number in range 0 to 1.

If the *rand* <= P1, there will not be changed.

If the P1 < *rand* <= P2+P1, then the corresponding request allocation method adopts the local-best position. For example, the *jth* datacenter processes *n* requests from the *ith* user. Then the request allocation in current solution should be replaced by local-best solution.

If the P1+P2 < *rand* <= 1, then the corresponding request allocation method adopts the global-best position. After updating current position in this way, there must be a situation where the same request is processed by multiple data centers simultaneously. Therefore, it is necessary to find these requests and reallocate them in locality algorithm. The request allocation algorithm based on particle swam optimization can be described in Algorithm 1.

At the beginning of the algorithm, it needs to initialize m users, n requests, d datacenters and p particles. And using greedy strategy to initialize the position and using random strategy to initialize velocity of a particle. Each particle contains a copy of information of all users and datacenters. The particle swam optimization consists of multiple iterations. In each iteration, each particle updates particle's position and velocity to get local best solution. When an iteration is completed, it will updates the global best solution. The algorithm ends with the final solution when all iterations are completed.

From the above analysis, it can be obtained that the time complexity of the PSO algorithm is proportional to the number of particles and the number of iterations. In addition, the time complexity of the three algorithms, GA, LA and

Algorithm 1 Request Algorithm Based on Particle Swam Optimization

1: Input m users, n requests, d datacenters and p particles.
2: Get initial current position, local-best position, global-best position, and velocity of a particle.
3: **for** each iteration I **do**
4: **for** each particle $particle_k$ **do**
5: Calculate $V^{t+1}_{particle_k}$ according to addition operator.
6: Calculate $X^{t+1}_{particle_k}$ according to multiplication operator.
7: Calculate aJ_{max} according to the allocation of $particle_k$.
8: Update local-best position for $particle_k$.
9: **end for**
10: Update the global-best position.
11: **end for**
12: Output the global-best position for request allocation and its aJ_{max}.

PSO, is comparable and all are related to the number of particles and the number of iterations.

IV. EXPERIMENT

A. EXPERIMENTAL SETUP

There are several parameters in the algorithm. In order to simulate real scenes, this study uses Wikipedia request traces [20] to represent the request number, and uses requests from 04:10 AM, September 19, 2007 GMT to 04:11 AM, September 20, 2007 GMT with line record format $< number, timestamp, url >$. WikiPedia can only provide relatively old traces, but old traces which contain peak and minimum values work well in this experiment. If WikiPedia updates traces, it can update the experimental data. In each hour, this study computes the average number of requests per minute for our experiments. Then, it gets the request distribution as shown below in Fig. 3. The chart shows the changing trend of request numbers, showing the peak and idle periods of the service respectively, which is in good agreement with the actual situation.

In this experiment, there are 5 datacenters and 500 end-users which send requests at the same minute. In real scenes, every request costs almost the same bandwidth, therefore each datacenter contains 6000 units of bandwidth and every request costs only 1 unit. The service time is different, because the hardware resources of each datacenter are different. Thus, the service time of datacenter is randomly generated from 30 to 50ms in this experiment. As mentioned in Section II, the assumption of this experiment is that waiting time satisfies the uniform distribution, therefore the average waiting time is equal to datacenter's service time. And it allocates the random number of requests to users and ensures that all requests are allocated. The transport delay which from end-user to datacenter is randomly generated from 0 to 50ms. In this way, this study can simulate the different distances between users and datacenters. This study tests the iterative

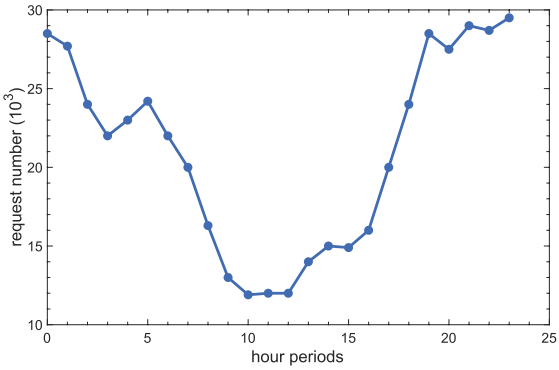

FIGURE 3. The request distribution.

process of PSO algorithm and compare our algorithm with GA and LA. There are few parameters that need to be adjusted in PSO algorithm.

In GA, this study calculates the minimum request delay for each user and datacenter, and allocates each request to a datacenter that can provide the minimum request delay. In LA, this study needs allocate each request to the closest datacenter.

As mentioned in Section II, the multiple datacenters model contains two roles: the provider and end-users. To maximize profits, provider tends to maximize the bandwidth utilization and users want to get low delay experience for each request. In the experiment of this study, it needs to evaluate the quality of PSO, GA and LA, with the aim of achieving a balance between two metrics. For the bandwidth utilization, the goal is to get the maximum value. To achieve this goal, this study proposes the definition of bandwidth utilization to evaluate each datacenter workload. And to evaluate the whole bandwidth utilization, this study relates the entire multiple datacenters model to a bargaining market to get aB_{max}. For the delay of the end-users, the aim is to reduce the delay time. And the delay contains three parts in this experiment. The first part is the transport delay between end-user and datacenter. The second part is the process time of each request. The third part is a random waiting time in queue inside the datacenter. From the three parts of delay, the definition of delay for end-users can get. Just as the method evaluating bandwidth utilization, it can achieve low delay aD_{min} at end-users in this Nash bargaining game. Given the above models, it can combine the aB_{max} and aD_{min} to get aJ_{max}. And a larger aJ_{max} means a better feasible request allocation solution. So, in PSO algorithm, this study uses joint optimization model aJ_{max} to find the global optimal solution. And comparing these algorithms in terms of bandwidth utilization and average request delay, and showing the effectiveness of our PSO algorithm.

B. RESULTS

In this section, this study does some experiments and uses figures to show experimental results.

Fig. 4 shows the changes of joint optimization value aJ_{max} in each iteration, and illustrates the speed of convergence in PSO algorithm is fast. The joint optimization value aJ_{max} gets convergence in nearly 10th iteration, and it gets the maximum

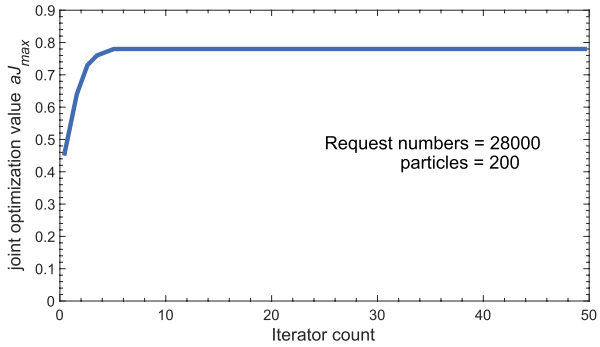

FIGURE 4. The changes of joint optimization value in each iteration.

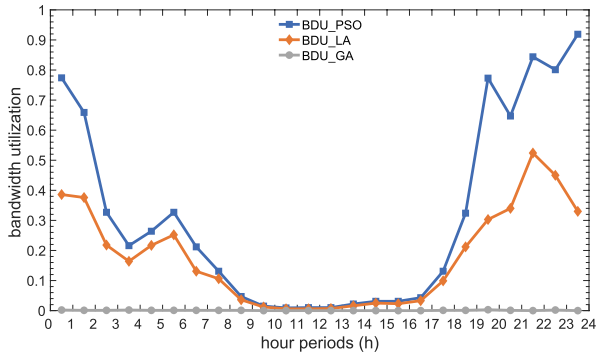

FIGURE 5. The value of bandwidth utilization in each hour.

value. And the characteristics of fast convergence can reduce the running time of the algorithm and improve the efficiency of the request allocation.

1) BANDWIDTH UTILIZATION

This section presents the results of the evaluation of the bandwidth utilisation of the algorithm. Fig. 5 and Table 2 show the value of bandwidth utilization in each hour. It indicated that GA performs worst in each hour and did not illustrate it for clarity of this figure. GA searches the minimum delay for each request. In this way, for most users, many requests would be allocated in the same datacenter and occupy all bandwidth of this datacenter while other datacenters cannot even receive requests. This is the reason for why the bandwidth utilization rate is too low. For LA, the situation that some datacenters are overloaded while others experience low bandwidth utilization is the same as GA, but LA performs better than GA due to different algorithm characteristics. PSO performs the best and maximum improvement is almost 0.8 and 0.4, compared with GA and LA. Besides, the overall trend is in line with the request distribution which is shown in Fig. 3.

2) AVERAGE REQUEST DELAY

The previous section shows the results of PSO compared to GA and LA, demonstrating the effectiveness of PSO in terms of bandwidth utilisation. Next the average request delay of the algorithm is evaluated. Fig. 7 shows the value of request average delay in each hour.

There are two reasons that GA performs best and has minimum average request delay. One is that GA searches the minimum request delay for each request. The other is that this

TABLE 2. The value of bandwidth utilization in each hour.

time period	PSO	LA	GA
0	0.774	0.386	0.002
1	0.659	0.376	0.001
2	0.327	0.218	0.001
3	0.216	0.164	0.002
4	0.264	0.217	0.001
5	0.327	0.252	0.001
6	0.212	0.131	0.001
7	0.131	0.106	0.001
8	0.047	0.036	0.001
9	0.015	0.012	0.000
10	0.009	0.006	0.000
11	0.010	0.006	0.000
12	0.010	0.007	0.000
13	0.022	0.016	0.000
14	0.031	0.025	0.000
15	0.031	0.023	0.000
16	0.043	0.033	0.000
17	0.131	0.099	0.001
18	0.324	0.212	0.001
19	0.773	0.303	0.003
20	0.647	0.340	0.001
21	0.844	0.524	0.000
22	0.801	0.450	0.002
23	0.919	0.330	0.000

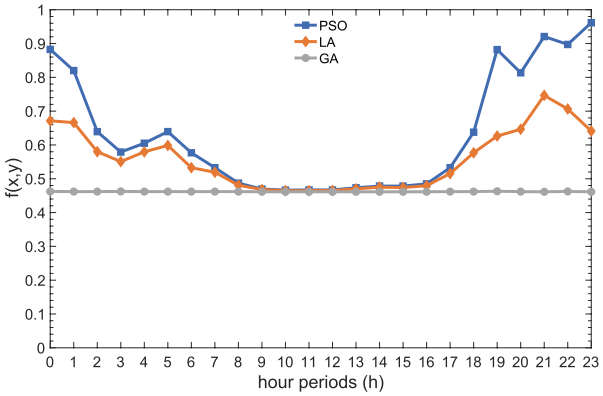

FIGURE 6. The evaluation result in each hour.

experiment only calculates the successfully handled requests. Although GA has minimum average request delay, it only handles partial requests. The difference between PSO and LA is not significant. And even in some hours, PSO performs better than LA.

3) OVERALL EVALUATION

Fig. 6 shows the results of the overall evaluation of PSO, LA and GA in terms of bandwidth utilization and request delay.

The evaluation results show that PSO can achieve better performance than LA and GA. Therefore, it can prove that PSO based on joint optimization model can improve the bandwidth utilization and keep the average request delay within acceptable limits or even better. These simulation experiments show that our proposed PSO can get better performance compared to the general algorithm.

V. RELATED WORK

Request allocation has received considerable attention from the research group in the past few years. There are two factors

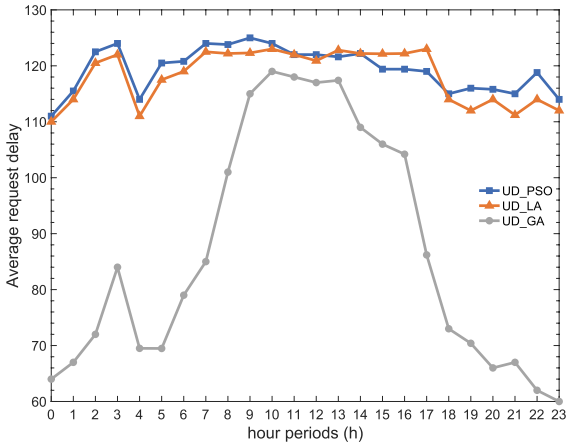

FIGURE 7. The value of average request delay in each hour.

needing to be concentrated on in this problem. On the one hand, request allocation aims to improve bandwidth utilization of datacenter providers. Different datacenters may reach their peak workload at different times. Some datacenters may reach peak workload while others have a low bandwidth utilization at a moment. On the other hand, large-scale applications provide service for many users so that it should reduce request delay. However, existing solutions only focus either on the benefit of end-users or datacenters.

For example, Najm *et al.* [21] propose a delay-aware resource allocation method that considers an adaptive delay warning threshold for various users. Xu and Li [22] adopt a general fairness criterion based on Nash bargaining solutions, and present a general optimization framework that models the realistic environment and practical constraints that a cloud faces. Wu *et al.* develop a prototype generic workflow system by leveraging existing technologies for a quick evaluation of scientific workflow optimization strategies. Zhang *et al.* [23] present a method to jointly optimize the cost and the performance of delivering traffic from an online service provider (OSP) network to its users. Their work mainly targets in the benefit of end-users.

Other solutions consider the benefit of datacenter providers, De *et al.* [24] employ a modified hierarchical clustering algorithm to categorize workloads according to their resource usage. The methodology presented provides insights to cloud service providers in optimizing resource allocation and improving profit. Laoutaris *et al.* [1] have designed, implemented, and validated NetStitcher, a system that employs a network of storage nodes to stitch together unutilized bandwidth, whenever and wherever it exists. Qureshi *et al.* [25] characterize the variation due to fluctuating electricity prices and argue that existing distributed systems should be able to exploit this variation for significant economic gains. Li *et al.* [26] present TrafficShaper, a new scheduler that shapes the inter-DC traffic to exploit the ''free'' time slots involved in the qth percentile charging model, so as to reduce or even minimize the transmission cost Mohammad Noormohammadpour *et al.* [2] presents QuickCast solution which can construct the multiple trees and determine the rate and schedule of flows on multiple forwarding trees in order to reduce the bandwidth needs and improve completion times of point-to-multipoint transfers. Yang *et al.* [27] propose a progressively-decrending algorithm (PDA) to schedule bulk transfers that minimizes bandwidth costs by finishing all transfers on time and with as little bandwidth as possible. Their work mainly focuses on the benefit of provider or datacenters.

There are also few solutions considering benefits of end-users and datacenter providers. Yue *et al.* [28] devise a two-phase optimization solution (TPOS), which contains a mapping phase to map service function chain requests (SFCRs) on servers and an adjustment phase to optimize the placement of virtual network functions (VNFs) and VNF requests. Li *et al.* [16] apply the software defined network (SDN) controller to enable the central control of the entire network, and propose a joint optimization model to consider high bandwidth utilization for provider and low delay for users.

This paper mainly focus on the joint optimization for bandwidth utilization for datacenter providers and request delay for end-users. This study uses meta-heuristics algorithms called particle swam optimization due to its smart iteration process and widely used in resource allocation problems to address the problem. In PSO, this study uses the same method as Kumar *et al.* [29] to make PSO suitable for discrete optimization. The algorithm proposed in this paper can achieve improved bandwidth utilization, and although it cannot achieve a significant reduction in the average request delay, it has been achieved to keep the average request delay within acceptable limits, or even better.

VI. CONCLUSION

Request allocation is one of the significant challenges in large-scale distributed service. The aim of this paper is that the low delay of end-users and high bandwidth utilization of datacenters. This study achieves this goal by leveraging the particle swam optimization technique and encounter three challenges. The first is how to make joint optimization for bandwidth utilization and request delay in our problem, the second is how to make particle swarm optimization suitable for discrete optimization problem and the third is how to determine the value of each parameter of particle swarm optimization when updating velocity of particle. To address the first challenge, this study first constructs a mathematical model to describe the problem where each feasible solution is represented by two matrixes. One represents bandwidth utilization of datacenters. The other represents delay of end-users. Then it combines the requirement of high bandwidth utilization and low request delay to quantitatively evaluate each solution. To address the second challenge, this study adjusts the coding scheme and redefines the operator in particle swarm optimization. In this way, it can make PSO suitable for discrete optimization problem. To address the third challenge, this study determines the value of each parameter when updating velocity of particle by using fitness function which

considers bandwidth utilization and request delay. It determines the value of fitness function using global best solution, local best solution and current solution. Finally, this study does some simulation experiments showing that our proposed particle swam optimization based request allocation algorithm can get better performance compared to other general algorithms.

REFERENCES

[1] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with netstitcher," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 74–85, Oct. 2011.

[2] M. Noormohammadpour, C. S. Raghavendra, S. Kandula, and S. Rao, "QuickCast: Fast and efficient inter-datacenter transfers using forwarding tree cohorts," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2018, pp. 225–233.

[3] R. Kiruthiga and D. Akila, "Heterogeneous fair resource allocation and scheduling for big data streams in cloud environments," in *Proc. 2nd Int. Conf. Comput., Autom. Knowl. Manage. (ICCAKM)*, Jan. 2021, pp. 128–132.

[4] M. Guo, L. Li, and Q. Guan, "Energy-efficient and delay-guaranteed workload allocation in iot-edge-cloud computing systems," *IEEE Access*, vol. 7, pp. 78685–78697, 2019.

[5] E. T. J. Kleinberg, *Algorithm Design*. New York, NY, USA: Person, 2005.

[6] M. E. Wolf and M. S. Lam, "A data locality optimizing algorithm," *ACM SIGPLAN Notices*, vol. 26, no. 6, pp. 30–44, Jun. 1991.

[7] M. Dorigo and T. Stätzle, *Ant Colony Optimization: Overview and Recent Advances*, vol. 272. Cham, Switzerland: Springer, 2019, pp. 311–351.

[8] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks: Theory and Applications*. Cham, Switzerland: Springer, 2019, pp. 43–55.

[9] B. Chopard and M. Tomassini, "Particle swarm optimization," in *An Introduction to Metaheuristics for Optimization*. Cham, Switzerland: Springer, 2018, pp. 97–102.

[10] J. Singh, K. Chatterjee, and C. B. Vishwakarma, "SISO method using modified pole clustering and simulated annealing algorithm," in *Advances in System Optimization and Control*, vol. 509, S. N. Singh, F. Wen, and M. Jain, Eds. Singapore: Springer, 2019, pp. 161–169.

[11] S. Wang, Y. Zhang, Z. Dong, S. Du, G. Ji, J. Yan, J. Yang, Q. Wang, C. Feng, and P. Phillips, "Feed-forward neural network optimized by hybridization of PSO and ABC for abnormal brain detection," *Int. J. Imag. Syst. Technol.*, vol. 25, no. 2, pp. 153–164, Jun. 2015.

[12] E. Feller, L. Rilling, and C. Morin, "Energy-aware ant colony based workload placement in clouds," in *Proc. IEEE/ACM 12th Int. Conf. Grid Comput.*, Sep. 2011, pp. 26–33.

[13] X. Xia, L. Gui, F. Yu, H. Wu, B. Wei, Y.-L. Zhang, and Z.-H. Zhan, "Triple archives particle swarm optimization," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4862–4875, Dec. 2020.

[14] X. Xia, L. Gui, G. He, B. Wei, Y. Zhang, F. Yu, H. Wu, and Z.-H. Zhan, "An expanded particle swarm optimization based on multi-exemplar and forgetting ability," *Inf. Sci.*, vol. 508, pp. 105–120, Jan. 2020.

[15] B. Wei, X. Xia, F. Yu, Y. Zhang, X. Xu, H. Wu, L. Gui, and G. He, "Multiple adaptive strategies based particle swarm optimization algorithm," *Swarm Evol. Comput.*, vol. 57, pp. 1–16, Jun. 2020.

[16] W. Li, H. Qi, K. Li, I. Stojmenovic, and J. Lan, "Joint optimization of bandwidth for provider and delay for user in software defined data centers," *IEEE Trans. Cloud Comput.*, vol. 5, no. 2, pp. 331–343, Apr. 2017.

[17] A. Muthoo, *Bargaining Theory With Application*. Cambridge, U.K.: Cambridge Univ. Press, 1999.

[18] *Amazon Web Services*. Accessed: 2015. [Online]. Available: http://aws.amazoncom

[19] S. S. Patnaik and A. K. Panda, "Particle swarm optimization and bacterial foraging optimization techniques for optimal current harmonic mitigation by employing active power filter," *Appl. Comput. Intell. Soft Comput.*, vol. 2012, Jan. 2012, Art. no. 897127.

[20] *Wikibench*. Accessed: 2009. [Online]. Available: http://www.wikibench.eu

[21] M. Najm, M. Patra, and V. Tamarapalli, "An adaptive and dynamic allocation of delay-sensitive vehicular services in federated cloud," in *Proc. Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2021, pp. 97–100.

[22] H. Xu and B. Li, "A general and practical datacenter selection framework for cloud services," in *Proc. IEEE 5th Int. Conf. Cloud Comput.*, Jun. 2012, pp. 9–16.

[23] Z. Zhang, M. Zhang, A. Greenberg, Y. C. Hu, R. Mahajan, and B. Christian, "Optimizing cost and performance in online service provider networks," in *Proc. Usenix Symp. Networked Syst. Des. Implement.*, San Jose, CA, USA, Apr. 2010, pp. 33–48.

[24] N. Dezhabad, S. Ganti, and G. Shoja, "Cloud workload characterization and profiling for resource allocation," in *Proc. IEEE 8th Int. Conf. Cloud Netw. (CloudNet)*, Nov. 2019, pp. 1–4.

[25] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for Internet-scale systems," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 123–134, Aug. 2009.

[26] W. Li, X. Zhou, K. Li, H. Qi, and D. Guo, "TrafficShaper: Shaping inter-datacenter traffic to reduce the transmission cost," *IEEE/ACM Trans. Netw.*, vol. 26, no. 3, pp. 1193–1206, Jun. 2018.

[27] Z. Yang, Y. Cui, X. Wang, Y. Liu, M. Li, S. Xiao, and C. Li, "Cost-efficient scheduling of bulk transfers in inter-datacenter WANs," *IEEE/ACM Trans. Netw.*, vol. 27, no. 5, pp. 1973–1986, Oct. 2019.

[28] Y. Yue, B. Cheng, X. Liu, M. Wang, B. Li, and J. Chen, "Resource optimization and delay guarantee virtual network function placement for mapping SFC requests in cloud networks," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 1508–1523, Jun. 2021.

[29] D. Kumar and Z. Raza, "A PSO based VM resource scheduling model for cloud computing," in *Proc. IEEE Int. Conf. Comput. Intell. Commun. Technol.*, Feb. 2015, pp. 213–219.

XU LIU was born in Tianjin, China, in 1985. He received the Ph.D. degree in computer science and technology from the Beijing University of Posts and Telecommunications (BUPT), China, in 2018.

From March 2011 to August 2020, he was a Lecturer at BUPT. Since September 2020, he has been an Engineer at the China Academy of Industrial Internet. He is the author of more than ten articles. His current research interests include mainly in the area of industrial Internet, big data, and cloud computing.

ZHENHAO LI was born in Sichuan, China, in 1993. He received the M.S. degree in computer science from the Beijing University of Posts and Telecommunications (BUPT), China, in 2019.

PENG XU was born in Inner Mongolia, China, in 1977. He received the Ph.D. degree in communication and information system from the Beijing University of Posts and Telecommunications (BUPT), China, in 2004.

From 2004 to 2008, he was a Lecturer with BUPT. Since 2008, he has been a Vice Professor with the State Key Laboratory of Networking and Switching Technology, BUPT. He is the author of two books, more than 40 articles, and more than 15 inventions. His current research interests include the area of big data, cloud computing, and the Internet of Things.

JIALING LI was born in Liaoning, China, in 1997. She is currently pursuing the M.S. degree in computer science from the Beijing University of Posts and Telecommunications (BUPT), China.