IF : 3.62 | IC Value 80.26

Volume-5, Issue-12, December - 2016 • ISSN No 2277 - 8160

Original Research Paper

Engineering

Performance Optimization of Distributed Face Recognition Based on Genetic Algorithm

Jie Shi

School of Computer Science and Software Engineering, Tian Jin Polytechnic University, Tianjin 300387, China

ABSTRACT

Due to the tasks partition into the agents unevenly will cause the time of processing videos is too long and the CPU utilization of the agent which processes the most tasks will be explosion. In order to solve the above problems, an improved genetic algorithm was proposed to balance the tasks of agents. Firstly, each agent counted the number of the videos and the number of pedestrians in each video. Secondly, the agents send the statistical data to server by SOAP. Finally, server reallocated the videos to agents by using improved genetic algorithm. Experimental results show that the performance of distributed face recognition model has been effectively improved by using improved genetic algorithm to realize load balance.

KEYWORDS : distributed face recognition, multi-agents, load balance, genetic algorithm

1 INTRODUCTION

Multi-agents based face recognition system is a new system, which contains one server and multiple agents. The agent can process multiple videos at the same time. However, if the tasks allocate to the agents unevenly, it will lead to the time of processing videos is too long, and the CPU utilization of the agent which processes the most tasks will be explosion. In order to solve this problem, an improved genetic algorithm (IMGA) was proposed to balance the tasks of agents.

Genetic algorithm (GA) is a commonly algorithm for load balance, Ted Scully et al. [1] and Ashish Jain et al. [2] has used GA to solve load balance problem in wireless local area network (LAN). Kousik Dasgupta et al. [3] applied GA in cloud computing load balance problem. M. Effatparvar et al. [4] solved the problem of multiprocessor load balance by using GA. The above GAs totally are traditional GA, which is able to get the optimal solution through random search, but the traditional GA is easy to premature convergence, and it's performance is not stability. In addition, Hui Cheng et al. [5] and M. Kaliappan et al. [6] has used dynamic GAs to solve the problem of load balanced clustering problem in mobile ad hoc network, dynamic GAs include GA with migration scheme and GA with memory scheme, the GA with migration scheme migrate maintains the diversity level of the population through replacing some individuals of the current population with random individual, and GA with memory scheme saves the useful individuals of current environment and applies them in the new environment. These two kinds of GA have the step of deleted genes node. However, the videos and the pedestrians in the videos can't be deleted. Therefore, an improved genetic algorithm (IMGA) is proposed to balance the tasks of agents. According to the demand of the agents load balance problem, this algorithm improves the crossover operation and the mutation operation. In addition, a stopping criterion formula is proposed in this paper to reduce the number of iterations. The performance of multi-agents based distributed face recognition system has been effectively improved by using the improved GA to realize load balance.

2 MULTI-AGENTS BASED DISTRIBUTED FACE RECOGNITION SYSTEM

Traditional video based face recognition systems are mostly centralized systems, which processes face detection, face tracking, feature extraction, face recognition and matching operation in the same server. With the continuous development of science and technology, the videos which needed to be processed are also increased. In order to improve the efficiency and scalability of the face recognition system, the multi-agents based distributed face recognition system has been proposed. As shown in figure 1, this model has one server and multiple agents, each computer has one agent, and every agent can process one or more pedestrians in multiple videos at the same time.

The agent in the multi-agents based distributed face recognition system is able to perform face detection, face tracking and feature extraction. And the server only performs the face recognition.

Agent and server send information to each other through the SOAP (Simple Object Access Protocol).

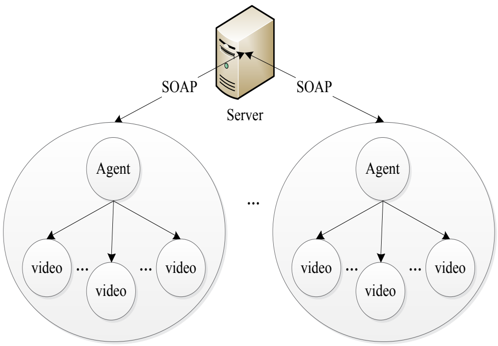

Figure1: multi-agents based distributed face recognition system

3 IMGA APPLIED TO LOAD BALANCE

Genetic algorithm is a meta-heuristic intelligent optimization algorithm, which can solve NP complex problems. In this paper, according to traditional GA, we improve the mutation operator to make it fit for the problem of domain partition and avoid premature convergence, and a stopping criterion formula is proposed in this paper to reduce the number of iterations.

3.1 PROBLEM DESCRIPTION

Assume that we partition m videos to K agents. The problem of domain partition can be described as follows.

(a) Let $A=\{A_1, A_2, ..., A_K\}$ be the set of K agents, A_i represents the i[th] agent.

(b) Let $AIP=\{IP_1, IP_2, ..., IP_K\}$ be the set of the IP address of agents.

(c) Let $V_i=\{V_{i1}, V_{i2}, ..., V_{in_i}\}$ be the set of n_i videos of i[th] agent, V_{ij} represents the j[th] video of i[th] agent $\left(0<j\le n_i\right)$.

(d) Because of the number of pedestrians in each video is not necessarily equal, the set of $P_i=\{P_{i1}, P_{i2}, ..., P_{in_i}\}$ represents the pedestrians of n_i videos processed in i[th] agent.

(e) The set of processing time of K agent is represented by $T=\{T_1, T_2, ..., T_K\}$, computation formula of T_i is as follows.

$$T_i=\sum_{j=1}^{n_i} t_{ij} \tag{1}$$

Where t_{ij} represents the average processing time that i[th] agent processes per frame of j[th] video. Computation formula of t_{ij} is as follows.

$$t_{ik}=t_{de}+t_{track}+t_{fe}+P_{ik}*t_{recog} \tag{2}$$

Where t_{de} represents face detection time, t_{track} represents face tracking time, t_{fe} represents feature extraction time, the above processing time are obtained by agents. And t_{recog} represents face recognition time, which obtained by server.

3.2 GENETIC OPERATION

Initialization: in order to distinguish the number of pedestrians received by the server is sent by which video processed by which agent. When the server has received the IP, V_{ij} and number of pedestrians of m videos, it will save the above information for one gene node, and the gene nodes which have the same IP nodes stored in the same chromosome, K chromosomes after initialization are shown in figure 2.

$$\begin{aligned}
&chrom_1:(IP_1,V_{11},P_{11}),(IP_1,V_{12},P_{12}),\ldots,(IP_1,V_{1n_1},P_{1n_1})\\
&chrom_2:(IP_2,V_{21},P_{21}),(IP_2,V_{22},P_{22}),\ldots,(IP_2,V_{2n_2},P_{2n_2})\\
&\vdots\\
&chrom_K:(IP_K,V_{K1},P_{K1}),(IP_K,V_{K2},P_{K2}),\ldots,(IP_K,V_{Kn_K},P_{Kn_K})
\end{aligned}$$

Figure 2: K chromosomes

The fitness function is used to measure the fitness value of the individual in IMGA. Because our purpose is made the management time minimum, the fitness function is as follows.

$$F_i=\frac{1}{c}\left(2\overline{T}-T_i\right)^{1/2} \tag{3}$$

Where c is a random positive integer that greater than 1, $\overline{T}$ represents the average processing time of K agents. Computation formula of $\overline{T}$ is as follows.

$$\overline{T} = \frac{1}{K}\sum_{i=1}^{K} T_i \quad (4)$$

Selection method: in this paper, we use the roulette wheel selection method to select chromosomes. Selection function is as follows.

$$P_{chrom}(i) = F_i / \sum_{i=1}^{K} F_i \quad (5)$$

Crossover operator: In order to generate better sub-chromosomes for the next generation, we select two best chromosomes for crossover operation. In this paper, we use partially matched crossover. The two crossing points are randomly selected. If the length of shortest chromosome is $L_{\min}$, the scope of first intersection select from $[1, L_{\min}/2)$, the scope of the second intersection select from $[L_{\min}/2, L_{\min}]$.

Mutation operator: In the traditional genetic algorithm, the mutation operation is performed on the two best chromosomes. In this paper, we perform mutation operation on the best chromosome and the worst chromosome to maintain the diversity of individuals in the population. We exchange the maximum gene in the worst chromosome and the minimum gene in the best chromosome.

Stopping criterion: In order not to miss the optimal solution, we set a stopping criterion formula, formula is expressed as.

$$\overline{F} / MaxF > \xi \quad (6)$$

$\overline{F}$ is the average fitness value, $MaxF$ is the maximum fitness value. Parameter ξ is the threshold value, which is a given number less than 1, but close to 1.

3.3 THE BASIC STEPS OF IMGA

Step 1: Initial chromosomes according to information received by server.

Step 2: Calculate the fitness value F_i of each agent according to the Eq. (3).

Step 3: Based on formula (6), do the following Steps from Step 4 to Step 6 if the threshold function is not satisfied.

Step 4: Perform crossover operation with crossover probability *Pc*, and re-calculate the fitness values for each chromosome according to Eq. (3).

Step 5: Perform mutation operation with mutation probability *Pm*, and re-calculate the fitness values for each chromosome according to Eq. (3).

Step 6: Go to step 3 to continue.

4 EXPERIMENTAL RESULTS AND ANALYSIS

In order to verify the proposed optimization method can improve the efficiency of the multi-agents distributed face recognition system, we did simulation experiment in a variety of conditions. All videos are acquired by a resolution of 704 x 516 Axis 215 PTZ network camera, frame rate of videos is 5 fps. The processors of server and agents is Intel(R) Core(TM) i7-3770M CPU@3.40GHz, the operating system is Windows7. The threshold of stop criterion of IMGA is 0.87.

The experimental simulation under the condition of the two agents, we simulated 4, 6, 8, 10, 12, 14 videos randomly assigned into 2, 3 agents respectively. The processing time and the maximum CPU utilization among the agents were used in this experiment to evaluate the performance of the IMGA algorithm and the random method. The results are showed in figure 3 and figure 4, the processing time include face detection time, face tracking time, feature extraction time and recognition time. In figure 3, it shows that the processing time decrease 34.71%

by using IMGA when the number of agent is 2, and the processing time decrease 33.83% when the number of agent is 3.

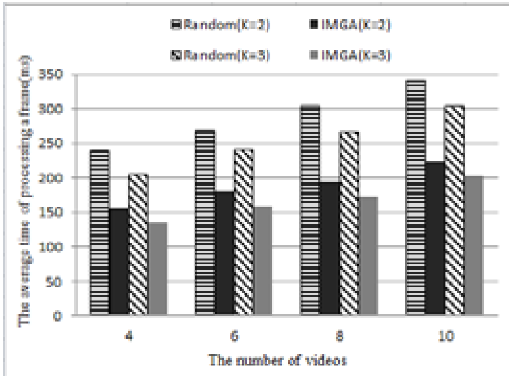

Figure 3: The average time of processing time

It is shown in figure 4 that the maximum CPU utilization of 2 agents and 3 agents respectively decrease by 21.34% and 25.65% through comparing IMGA and random method.

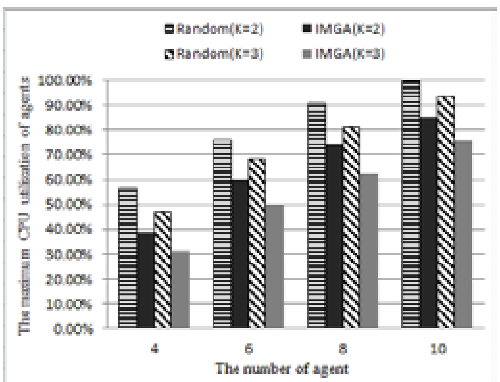

Figure 4: The maximum CPU utilization of agents

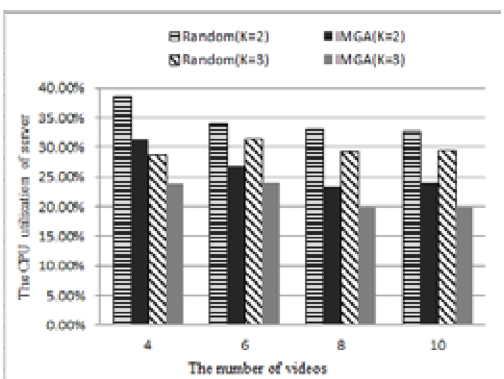

Figure 5: The CPU utilization of server

The result in figure 5 shows that the CPU utilization of server of 2 agents and 3 agents respectively decrease by 23.75% and 25.96% through comparing IMGA and random method.

5 CONCLUSIONS

An improved genetic algorithm is proposed in this paper to improve the performance of multi-agents based face recognition system. In order to make the genetic algorithm to adapt to the problem of load balance, and have a better performance, two improvements are proposed based on the traditional GA. We improve the mutation operator to avoid premature convergence and maintain the diversity of the population. In addition, a stopping criterion formula is proposed to reduce the number of iterations. Experiments show that the IMGA proposed in this paper effectively balance the load of agents and improved the performance of the multi-agents based distributed face recognition system.

REFERENCES:

[1] S. TED, N. B. KENNETH. (2009), "Wireless LAN load-balancing with genetic algorithms." Knowledge-Based Systems, 22, 529-534.

[2] A. JIAN, N. S. CHAUHARI. (2012), "Genetic algorithm based concept design to optimize network load balance." Ictact Journal on Soft Computing, 2(4), 357-360.

[3] K. DASGUPTA, B. MANDAL, P. DUTTA. (2013), "A Genetic algorithm (GA) based load balancing strategy for cloud computing." Procedia Technology, 10(2), 340-347.

[4] M. EFFATPARVAR, M. S. GARSHASBI. (2014), "A genetic algorithm for static load balancing in parallel heterogeneous systems." Procedia - Social and Behavioral Sciences, 129, 358-364.

[5] H. CHENG, S. X. YANG, J. N. CAO. (2013), "Dynamic genetic algorithms for the dynamic load balanced clustering problem in mobile ad hoc networks." Expert Systems with Applications, 40(4), 1381-1392.

[6] M.KALIAPPAN, S.AUGUSTINE, B. PATAMASIVAN. (2016), "Enhancing energy efficiency and load balancing in mobile Ad hoc network using dynamic genetic algorithms." Journal of Network & Computer Applications, 73, 35-43.