Instantaneous overload optimization of scheduling algorithm for real-time systems of Linux operating system

Fuhuai Ding*, **Jinyi Chang**

Changshu Institute of Technology, Changshu 215500, Jiangsu, China

Received 1 June 2014, www.cmnt.lv

Abstract

Aiming at the real-time scheduling algorithm in practical application of Linux operating system and some problems such as instantaneous overload, this article proposed a based on local pheromone updating path optimization ant colony algorithm optimization and excitement factor of Linux operating system real-time scheduling model, first of all we use local pheromone update strategy, we introduce the ant- cycle model, and improve the convergence performance of standard ant colony algorithm, then we use the concept of excitement in artificial fish algorithm, the standard ant colony algorithm optimization process is divided into two stages of optimization, in order to improve the searching capability of the ant colony algorithm, finally improve the algorithm of pheromone persistence parameters, in order to improve the original routing capabilities of the algorithm. The simulation experiments show that the ant colony algorithm proposed in this article which is based on local pheromone updating optimization and excitement factor in comparison with the standard ant colony algorithm, has better convergence performance and convergence speed, and can better solve the Linux operating system real-time scheduling of instantaneous overload problem.

Keywords: Linux operating system, real-time scheduling, improved ant colony algorithm, local pheromone updating, excitement factor

1 Introduction

Real-time operating system is widely applied in various fields of the current, more and more cause the attention of people. Linux with its open source code, the fine kernel, efficiency and robustness, and effective network communication support ability, easy to implement security policy, rich open the advantages of the application software, has become one of the major operating systems of contemporary influential, also make the Linux get rapid development in the field of real time [1]. However, the existing Linux is a general-purpose operating system, although it USES a lot of technology to speed up the system running and reaction, it is not essentially a real-time operating system, did not meet the requirements of real-time operating system has the characteristics of fast speed and predictability, cannot be directly applied in the real-time environment [2]. To make Linux more suitable for real-time applications, it requires the adoption of a certain technical method, in view of the Linux real-time defect for real-time transformation to make it meet the requirements of real-time operating system [3]. Therefore, the research of Linux real life technology to promote the development of real-time field and industrial field has important significance.

Linux's rising make Linux scheduling algorithm has become the focus of attention. At present there are based on the study of scheduling algorithm, periodic tasks, aperiodic tasks, static, dynamic, and so on. the most influential one is proposed by C. L. Liu and J. W. Layland based on fixed priority flat rate algorithm and the earliest time priority algorithm based on dynamic priority, marking real-time scheduling algorithm sets a new milestone, many other algorithms are based on the two algorithms are improved, such as minimum margin LSF priority algorithm, the supreme value HVF priority algorithm, priority table design PTD and highest value density priority algorithm HDF, etc. [4]. Deng Z proposed based on CUS and TBS reserved bandwidth algorithm on the basis of study the CUS and TBS, then on this basis, build a two-layer scheduling framework, the framework can effectively solve the problem of real-time periodic task which is accidental, coexist [5]. Alex F analysed the GEDF scheduling algorithm and the advantages of the algorithm in dealing with soft real-time tasks and put forward on multiprocessor scheduling stochastic framework of soft real-time tasks [6]. S. Banruah studied RM algorithm symmetric multiprocessor platform for the processing of periodic tasks, and give a simple and effective test method of the periodic task using the RM scheduling algorithm on the symmetric multiprocessor scheduling [7]. Theodore Ebkaer mainly analysed the EDF scheduling algorithm and its implementation on multiprocessor platform, and the algorithm was to schedule ability analysis, put forward relevant improvement ideas based on results of the analysis of these aspects [8]. Marko Bertogna proposes a hybrid finite capturing real-time scheduling algorithm, the algorithm running is low-overhead, reservation system can run full pre-emption algorithm [9]. Li Ju put forward a kind of embedded MPSoC system task scheduling management methods, shows the new situation for multiple processor system chip such as mobile communication, network

* ***Corresponding author's*** e-mail: dingfuhuai@126.com

security, and multimedia applications in the field of embedded application [10]. Xu Jia proposed MTCO scheduling algorithm based on an improved minimum credible price of grid task, this algorithm not only considers the task scheduling time budget and time cost, and considered the status of the real-time service grid resources, MTCO algorithm can effectively under the same deadline and budget to ensure completion of task scheduling [11].

This paper proposed Linux operating system real-time scheduling algorithm based on improved ant colony algorithm, aiming at the real-time scheduling algorithm in practical application process also some problems such as instantaneous overload, we optimized it using local pheromone update strategy and path optimization strategy based on excitement factor and improved the pheromone persistence parameters.

2 The ant colony algorithm defect analysis based on the TSP problem

Before using ant colony algorithm Linux momentary overload operating system optimization, this paper analyses the performance of TSP problem based on ant colony algorithm first. TSP problem is finding a through every city once and returned to the starting point of the minimum cost of a closed loop for a given *n* cities set. The objective function TSP problem is:

$$\min D = \sum_{t=1}^{n-1} d(i,i+1) + d(n,1) . \tag{1}$$

Among them, *n* means *n* cities ready to Traversal, $d(i,i+1)$ said the distance between city i and $i+1$, $d(n,1)$ says from the first n cities back to the starting point to the distance.

Set $b_i(t)$ the number of ants which is located in the section *i* point the number of ants at time *t*, then $m = \sum_{i=1}^{n} b_i(t)$; $\tau_{ij}(t)$ is the pheromone concentration of the path at *t* time, $\Gamma = \{\tau_{ij}(t) \mid v_i, v_j \in V\}$ is the collection of residual pheromone concentration of set V nodes in two connected side e_{ij} at *t* time. In every path pheromone concentration is equal to the initial time, and set $\tau_{ij}(0) = const$, based on ant colony algorithm, the optimal route choice is by looking for nodes in the directed graph $G = (V, E, \Gamma)$ to obtain the minimum cost value.

Ants $k(k = 1,2,...,m)$ which is in the process of movement, determines the transfer direction according to various path pheromone concentrations. Here we record ant *k* current node with table $tabu_k(k = 1,2,...,m)$, the path set makes dynamic adjustment with evolution of $tabu_k$. In the process of path selection, ants calculate the state transition probability of the path according to various path pheromone concentration and heuristic information. Set $p_{ij}^k(t)$ means the state transition probability of Ant *k* nodes by *i* to *j* at *t* time.

$$p_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ik}(t)]^\beta}{\sum\limits_{s \in allowed_k} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta} \\ 0, \qquad else \end{cases} . \tag{2}$$

In the Equation (2): $allowed_k = \{V - tabu_k\}$ means the next available node Ant *k* may choose; *α* is the heuristic factor for information, says the relative importance of the trajectory, reflects accumulated pheromone and the role in the process of Ant's movement. The greater its value, then the ant more tend to choose the path of the other ants have been, the more collaborative between ants; *β* is the expect heuristic factor. According to the relative importance of visibility, reflects inspire pheromone's important degree in ant path choosing, the greater its value, is more close to the state transition probability greed rules; $\eta_{ij}(t)$ is the heuristic function, and its expression is as follows:

$$\eta_{ij}(t) = \frac{1}{d_{ij}} . \tag{3}$$

In the Equation (3): d_{ij} - the path $<i, j>$ distance between two adjacent nodes *i* and *j*. For Ant *k*, smaller d_{ij}, then $\eta_{ij}(t)$ is bigger, so as $p_{ij}^k(t)$. Obviously, the heuristic function said expectations of the ants moving from node *i* to node *j*.

In order to avoid excessive residual pheromone caused by residual information submerged inspired, after an ant walk the one step or complete after traversal of all *n* nodes, update the remaining information. $\tau_{ij}(t)$ means the pheromone concentration of path $<i, j>$ at *t* time, then the pheromone concentration at *t*+1 time is

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) , \tag{4}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^k(t) . \tag{5}$$

In the Equation (4): ρ - the pheromone volatilization coefficient, $\rho \in [0,1)$, then $1-\rho$ express residual pheromone concentration factor, means the relative important degree said residual pheromone. $\Delta\tau_{ij}(t)$ means the pheromone concentration increment of the path $<i, j>$ between time *t* and *t*+1, at the initial time $\Delta\tau_{ij}(0) = 0$, $\Delta\tau_{ij}^k(t)$ means the increase of pheromone concentration on path $<i, j>$ of the Ant k between time *t* and *t*+1.

From the process, which the ant colony algorithm solves TSP problem, you can see that ant colony algorithm has the following deficiencies:

Ant colony create the first path based on data of distance between cities, which result in that the left pheromone by ant colony does not have to show the direction of the optimal path. If the amount of information on a spread out in all directions, then it will mislead ant colony in choosing path. The random strategy which Ant colony algorithm use can lead to slow evolution and convergence speed can't be optimized. Leading to the smaller probability that the chosen route will be selected, thus appear pause phenomenon.

Existence of evaporation coefficient *ρ* will cause the pheromone amount of those unsearched path reduced to 0 those unsearched path, which results in the decrease of the algorithm's global search ability. If *ρ* is too big, will cause the searched path is repeatedly larger choosing possibility, affect the algorithm's global search ability, thus trapped in local optimal solution; Reduce *ρ*, while it is possible to improve the algorithm's global search ability, but it makes the convergence rate of the algorithm reduced.

3 Based on improved ant colony algorithm of real-time scheduling instantaneous overload optimization

The priority of the ant colony algorithm is only determined by its cycles, which makes some tasks while real-time demand is higher, but because of its cycles determines that it does not have a high priority, so we need to change some task priority scheduling system meet the needs of the urgent tasks. The dynamic changing tasks cycle is a good way to solve this problem.

3.1 LOCAL PHEROMONE UPDATING OPTIMIZATION

For ant colony algorithm in the Linux operating system, real-time scheduling time overload optimization has slow convergence speed and prone to stagnation phenomenon of the defect, this paper first uses the local pheromone update strategy for the optimization.

Set Ant *k* in the task *i*, in accordance with the pseudo-random proportion rule we select task *j* as its next scheduled tasks. The rules are given by the following Equation:

$$j = \begin{cases} \arg\max\{\tau_{il}[\eta_{il}]^{\beta}\}, q \le q_o \\ J, else \end{cases} \tag{6}$$

In Equation (6) *q* is evenly distributed in the interval [0, 1] on a random variable, $q_o (0 \le q_o \le 1)$ is a parameter, *J* is probability distribution of a random variable given by Equation (2). This selection strategy increased the diversity of ant search path can effectively avoid premature algorithm trapped in local optimum, and into stagnation.

We added the local pheromone update strategy in the algorithm, using the ant-cycle model, we combine the global pheromone updating and local pheromone updates. After the first Ant *k* walking one side, namely the press type local pheromone update:

$$\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} + \rho\Delta\tau_{ij} , \tag{7}$$

which

$$\Delta\tau_{ij} = \frac{Q_1}{L_z} \tag{8}$$

In Equation (8) Q_1 press calculation:

$$Q_1 = s \cdot e^{1-s} \cdot Q, \forall i, j = 1, 2, ..., n , \tag{9}$$

n is the size of the Linux system task, *s* means the number of elements in $tabu_k$, L_z is the first Ant *k* the current through the length of the path.

In order to expand the searching space of solution, avoid algorithm falls into local optimum. When all the ants finished a cruise, we randomly choose one ant's parade path length L_k as standard, other ants travel path length $L_i (i \ne k)$ compare to this path, update the pheromone concentration of path L_k and shorter path's pheromone concentration. Which makes the algorithm have the ability to expand the search space of solution and makes the randomness of the path, which is more likely to jump out of local optimal solutions and search the optimal solution. Specific means is:

1) When the random choice of path L_k is not the optimal path, which found from the ants cruise, in this paper, we use the following Equation to do global pheromone update:

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}, \rho \in (0,1) , \tag{10}$$

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} , \tag{11}$$

which $\Delta\tau_{ij}^{k}$ are defined as follows:

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q_2}{L_k - nBestTourLen + 1}, j \in allowed_k \\ 0, \qquad else \end{cases} . \tag{12}$$

2) When L_k was just randomly chosen path in this tour, when we find the best solution ,this paper choose Equations (10), (11) to update global pheromone in order to avoid falling , $\Delta\tau_{ij}^{k}$ of these are defined as follows:

$$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q_2}{nBestTourLen}, j \in allowed_k \\ 0, \qquad else \end{cases} , \tag{13}$$

Q_2 is a constant and $Q_2 \in (0,1)$, L_k is the path length of Ant *k* in the circulation, *nBestTourLen* is the optimal solution in this cycle. This Equation can make the increase of the near optimal solution path's pheromone become more obvious, at the same time to avoid the algorithm faster fall into the local optimum, which is conducive to accelerate the search speed and improve the ability of searching the optimal solution.

3.2 BASED ON THE EXCITEMENT FACTOR OF THE PATH OPTIMIZATION

Improved ant colony algorithm based on excitement factor can be divided into two stages, the first stage of an ant routing strategy is doing path optimization according to the excitement function, determine the scope of the optimal solution. The second phase, we speed up the convergence process, based on basic ant colony algorithm, until we find the global optimal solution.

1) The first phase of path optimization.

This article was inspired by artificial fish algorithm food concentration Y , put forward the excitement factor of ants, used to represent an ant with the similar degree of optimal solution, the greater the excitement of ants, spell out the path is more close to the optimal solution, conversely, the farther from the optimal solution. Excitement function can be calculated by the Equation (14):

$$D_i = \gamma \frac{\sum_{k \in path} \tau_k(t)}{\sum_{k \in path} d_k}, \tag{14}$$

which, $path$ for collection of the path, $\tau_k(t)$ for information on the path *k*, d_k is the length of the path *k*, γ for regulating factor ($0 < \gamma < 2$), it showed the importance of pheromone when calculating the excitement.

Then, according to the excitement has been state transferred. Set the current ants in the current city's excitement factor X_i , explore its horizon's largest excitement of ants $X_{\max}$, if $X_i < X_{\max}$, and with the transfer direction of the $X_{\max}$ have the same number *n* of ants partners meet $n/N < \delta$, indicates that the $X_{\max}$ direction is close to the optimal solution, and it's not too crowded around, then transfer to the $X_{\max}$ direction. If $X_{\max} < X_i$ or $n/N > \delta$, perform the following actions: Ants X_i in the visible range within the scope random select a ant with excitement X_j Ant, if $X_j > X_i$, then in the transfer of X_j direction, On the other hand, within the visible range again choose a random ant, judge whether it meet the conditions. After the test of T , if it still does not meet the conditions, then move randomly.

Through ants' transfer path finding strategy, this paper gives early detailed search process based on excitement factor process which is as follows:

a) M ants excited degree is the initial value is 0, in the before m times cycle of each ant climbing strategy is adopted to improve the random routing, and USES the model updating pheromone ant weeks, as shown in Equation (15).

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k}, j \in allowed_k \\ 0, \ else \end{cases}. \tag{15}$$

b) Search process begins in round $m+1$, every ant was calculated the excitement factor right after it completed the cycle, this round of search still adopt the strategy of random.

c) In the hereafter search process, the road rules adopt the strategy of excitement. After this stage complete n round of search, we determine the optimal solution in minimum range.

2) The second phase of path optimization.

At the same time initialize pheromone, and uses the basic ant colony algorithm to search. In order to speed up the process of convergence, information stimulating factor can be set larger value, make ants accumulated sports information in the process plays a great role in the ant movement, strengthen the collaboration between the ants. Reduce the expected heuristic factor at the same time, the state transition probability from greedy rules.

3.3 THE PHEROMONE PERSISTENCE PARAMETER OPTIMIZATION

Choice of pheromones persistence parameter ρ determines the importance of the prior knowledge of the ants in routing: when ρ is small, after the pheromone updated, unselected/less likely chosen path's pheromone concentration will be decreased rapidly. In the next round of routing, the subsequent ants tend to choose the walked path, lead to a drop in the ant search scope, premature algorithm trapped in local optimal solution, but the algorithm convergence speed is faster. Conversely, when ρ is larger, unselected or less chosen path's pheromone concentration reduced slower, subsequent ants may choose those path which do not taken or rarely walked, the ant's search scope will expand. At this point, the algorithm is not easy to fall into local optimal solution, but due to the increase of the paths optional number, convergence time/number of iterations will increase.

At the beginning of the improved ant colony algorithm, because we don't know the task scheduling information of Linux operating system, we need consider selecting the optimal path from alternative paths. ρ should be set a larger value, in order to expand the range of ants routing. And seek different paths which reach the target domain waiting for scheduling task node. After the algorithm undergoes certain number of iterations, path pheromone's difference is bigger, we can accelerate the algorithm convergence speed, ρ should be set to a smaller value. In this article, we set ρ the current number of iterations n_{it} index function, the expression of ρ was shown in Equation (10). In this way, value of ρ is bigger at beginning of the improved ant colony algorithm, the ant search scope is larger, with the increase of iterations n_{it} , ρ value decreases, and improves the convergence speed of the algorithm.

$$\rho=\rho_0\cdot e^{-n_{it}/N_{it}\cdot 10}(0\le n_{it}\le N_{it})\,. \qquad (16)$$

The $0<\rho_0<1$, N_{it} is the total number of iterations of the algorithm. ρ_0 for the initial value of ρ, take a larger value.

4 Algorithm performance simulations

In order to verify the effectiveness of the improved algorithm proposed in this paper, simulation experiments is run. First we choose TSPLIB problem Oliver30 to improve ant colony algorithm simulation. Experimental parameters Settings: number of ants $k=30$, $\alpha=1$, $\beta=5$, $\rho=0.3$, $Q=10$, $\tau_{ij}(0)=10$. The maximum number of iterations is 50; we ues the improved ant colony algorithm proposed in this paper to solve Oliver30, the results are shown in Table 1, the evolution process is shown in Figure 1.

TABLE 1 Simulation results

Experiments No.	Iterations	Target value	Run time
1	17	423.74	27
2	21	423.74	25
3	33	423.74	24
4	13	423.74	28
5	21	423.74	27
6	27	423.74	26
7	12	423.74	27
8	13	423.74	25
9	33	423.74	24
10	23	423.74	25

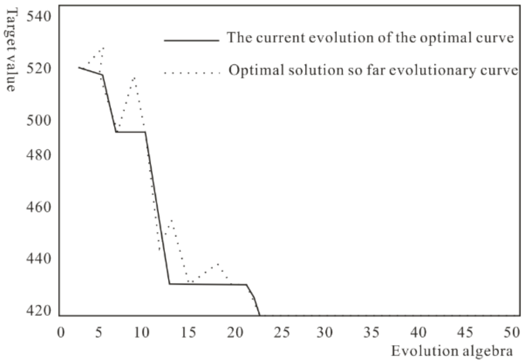

FIGURE 1 Evolutionary state

The experimental results of table 1 shows that the proposed ant colony algorithm improved in the 10 times experiments are able to search the optimal solution, the success rate was 100%, the visible algorithm's performance is very good, and average calculated time is 25.8 seconds, computational efficiency is very high. From Figure 1, it is not hard to see the convergence of this algorithm is also very good, when it evolved to 23 generation the optimal solution has been converged.

Then, with the improved ant colony algorithm to optimize Linux operating system real-time scheduling optimization of instantaneous, the results are as follows:

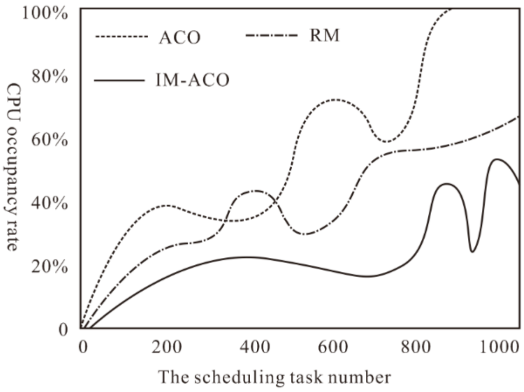

FIGURE 2 The simulation results of real-time scheduling of Linux operating system

From the simulation results showed that the proposed ant colony algorithm based on local pheromone updating optimization and excitement factor path optimization can well solve the problem of the Linux operating system instantaneous overload, improve the performance of real-time scheduling.

5 Summaries

In the method of improving Linux real-time performance, the main methods are refined clock granularity, enhance the Linux kernel pre-emption, improve the Linux verify sort of scheduler scheduling strategies, including process scheduling algorithm is one of the important factors affecting system real-time performance. Linux operating system real-time scheduling algorithm is proposed in this paper based on improved ant colony algorithm, simulation results show that this algorithm is successful in solving the instantaneous overload Linux operating system, and good performance.

Acknowledgments

This work was supported by Education department of Jiangsu province (No. 12KJB520001).

References

[1] Li Yi 2014 A real-time scheduling approach based on priority table *Computer Application Research* **31**(5) 1410-3

[2] Hu Y 2013 Hybrid real-time tasks scheduling with virtual characteristic parameters and its application *Application Research of Computers* **30**(4) 1004-6

[3] Feng H 2013 Channel-based real-time scheduling algorithm for WiMAX system *Application Research of Computers* **30**(1) 224-6

[4] Jiang X 2013 EPA across a network segment delay adaptive real-time scheduling algorithm research *Industrial Automation* (10) 35-9

[5] Huang S 2013 Multi-core Platform, a new real-time scheduling model *Journal of Huazhong university of science and technology: natural science edition* (3) 44-7

[6] Nie P 2012 A lightweight multicore real-time scheduling algorithm *Journal of Lanzhou University (Natural Science)* **48**(5) 118-23

[7] Zhao Q 2012 Load balancing and admission control algorithms in the real-time virtualization environment *Computer Engineering and Design* **33**(10) 3862-9

[8] Zhang Y 2012 An Effective Algorithm of Mixed-Criticality Driven Real-Time Scheduling *Journal of Northwestern Polytechnical University* **30**(4) 617-21

[9] Zhang D-s 2012 The optimal energy saving cost sensitive multi-processor real-time scheduling algorithm *Journal of computers* **35**(6) 1297-312

[10] Wu X 2012 Energy-Aware Scheduling of Hard Real-Time Tasks in VFD-Based Multi-Core Systems *Journal of Computer Research and Development* **49**(5) 1018-27

[11] Bai Lu 2012 Analysis on schedulability of fixed-priority multiprocessor scheduling *Journal of Computer Applications* **32**(3) 603-5

Authors

Fuhuai Ding, 10.1988, China.

Current position, grades: postgraduate student of Soochow University, China.
University studies: Bachelor's degree in computer science and technology from Applied Technology College of Soochow University, China in 2011.
Scientific interest: embedded operating system and the technology of the internet of things.

Jinyi Chang, 02.1955, China

Current position, grades: A professor of Changshu Institute of Technology.
Scientific interest: Spatial decision support system, Software engineering services and Database application technology.