Практическая значимость. Проанализированы различные потребности пользователей, эксперименты дали хороший результат. В сравнении с другими методами, предложенный метод поиска изображений по представляющей интерес области является более быстрым и эффективным.

Ключевые слова: *NSCT-контурная трансформация, семантическая брешь, релевантная обратная связь, поиск изображений, мера расстояния Канберра*

Рекомендовано до публікації докт. техн. наук М.О. Алексєєвим. Дата надходження рукопису 16.11.14.

Yihua Lan[1],
Yanwei Tian[2],
Yan Tian[1],
Jinjiang Liu[1],
Xiao Jia[1,3]

1 – Nanyang Normal University, Nanyang, China
2 – Anyang Normal University, Anyang 455002, China
3 – Southern Medical University, Guangzhou 510515, China

TSP PROBLEM SOLVING METHOD BASED ON BIG-SMALL ANT COLONY ALGORITHM

Іхуа Лань[1],
Яньвей Тянь[2],
Янь Тянь[1],
Цзіньцзян Лю[1],
Сяо Цзя[1,3]

1 – Наньянський педагогічний університет, м. Наньян, провінція Хенань, КНР
2 – Аньянський педагогічний університет, м. Аньян, провінція Хенань, КНР
3 – Південний медичний університет, м. Гуанчжоу, провінція Гуандун, КНР

МЕТОД РІШЕННЯ «ЗАДАЧІ КОМІВОЯЖЕРА» НА ОСНОВІ АЛГОРИТМУ «ВЕЛИКИХ І МАЛИХ МУРАШОК»

Purpose. The traditional ant colony optimization algorithms have been used to solve the NP-hard problem, Traveling Salesman Problem (TSP), which is based on the rule that ants tend to choose high pheromone concentrated path. The max-min ant system (MMAS) most commonly achieves the nearest neighbouring city and always formulates the local optimal solution.

Methodology. The "big-small ant colony" algorithm with a kind of "jump pit strategies" has been formulated. Where, the big ants can carry much more pheromones and are prone to making mistakes.

Findings. First, the "big-small ant colony" algorithm was employed to accelerate the convergence speed. Then, by using a kind of jump pit concepts, a wider range path searching was provided, where the "small jumping strategy" allowed more than one ant to go along a different path, and the "big jumping strategy" put a barrier on the pheromone convergence path forcing the ants to choose other different paths. The experimental results showed that the modified algorithm always converges to the optimal results unlike the MMAS.

Originality. The modified ant colony optimization algorithm was studied and the effectiveness of the idea, which was put forward, was discussed.

Practical value. The proposed algorithm may be employed to solve other problems, especially together with some deterministic algorithm to realize quick global optimization.

Keywords: *ant colony optimization, traveling salesman problem, max-min ant system, NP-hard problem, global search, pheromones, "jump pit strategy"*

Introduction. In nature, ants crawl out from the cave to find a food source, upon finding it they come back to the colony. They leave a path marked by pheromones substances. In an ant colony, the algorithm of information interaction is based on the pheromones mainly. The ants are able to perceive the existence of this kind of material and its strength in the absence of visual signs. At the initial stage, in the environment there are no pheromone paths, and the ants search for things in a random way. Then the process of the food source search is affected by the previous residues of the ants' pheromone. Ants tend to choose the path with the high concentration of the pheromone. At the same time, the pheromone is a kind of volatile chemicals, which evaporates slowly. The longer is a path, the more time an ant spends on travelling down the path and back, the more time the pheromones have to evaporate. The pheromone residual is relatively higher on a shorter path. Subsequently the probability that other ants will choose the shorter path is large. This leads to the more and more ants walk along the short path. Consequently, the pheromone concentration that remains on the path will also increase. Therefore, the ants' collective behaviour constitutes the pheromone positive feedback process, which allows finding the shortest path. The positive feedback mechanism strengthens the performance of better solutions, but it may cause the ant colony algorithm easily fall in premature phenomenon and stagnant phenomenon when solving problems. In real life, many ants cannot complete complex tasks, but can find the current optimal solution path to adapt to the changes in the environment. Ant Colony Optimization (ACO) algorithm is a method, which is used to find an optimal path through graphs. Marco Dorigo put it forward in

1992, in his PhD thesis. The ACO was inspired by the behaviour of ants searching for food. The ACO algorithm is a kind of simulated evolutionary algorithm. Previous studies showed that the algorithm has many good properties.

The ACO algorithm together with the current popular genetic algorithm, particle swarm optimization algorithm and artificial neural network, artificial immune algorithm and artificial fish algorithm belongs to the group of bionic optimization algorithms. They all belong to a class of mimic natural biological systems; they depend entirely on the biological instincts, through unconscious behaviour to optimize the condition for optimum need intelligent optimization algorithm to adapt to the environment [1–3]. Recently, many ACO applications have been found. Camp et al. use the ACO to develop for discrete optimization of space trusses [4]. Shuang et al. use parallel computation mechanism for the ACO [5]. Mavrovouniotis et al. solve the dynamic TSP by ACO [6]. The ACO algorithm is successful to solve the traveling salesman problem (TSP), which is to find the shortest possible route that visits each city exactly once and returns to the origin city [7]. If assuming each city as a node of the graph, then the traveling salesman problem is to find a cost minimum circuit on the complete graph of N nodes.

Traditional ant colony algorithms. Ant colony algorithm has the following advantages: the positive feedback, strong robustness, distributed computing, etc. At the same time, the ant colony algorithm has its own deficiencies, such as computing time is long, prone to stagnation phenomenon, etc. Starting from the original ant system, scholars have carried out many researches on its improvement, mainly including the systems such as basic graph ant system (BGAS), ant system with elitist strategy (ESAS), rank-based version of ant system (RBVAS), ant colony system (ACS), and max-min ant system (MMAS).

Basic graph ant system (BGAS) algorithm process can be expressed as follows [8]:

For (the number of iterations arriving or iterative condition satisfaction)

For (all ants foraging)

Calculate the ant routing path, recording the path for the best path;

Update pheromone (including volatile and the best path);

End

End

Ant colony system (ACS) algorithm process can be expressed as follows [9].

In the algorithm initial moment, every path has the equal amount of pheromone, assume $\tau_{ij}(0) = C$ (constant number), Put m ant in n city based on some rules. At t time, the probability of ant k located in the city i choosing moving to the city j can be expressed as follows.

$$p_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum\limits_u \tau_{iu}^{\alpha}(t)\eta_{iu}^{\beta}(t)}, j \in allowed \\ 0 \qquad , otherwise \end{cases}. \quad (1)$$

Where, τ is the amount of pheromone between the cities, η is the heuristic information between the cities, α and β reflect the ants accumulated in the process of sports information and inspiration in the ant selected the relative importance path respectively.

When all the ants completing a cycle, every path pheromone amount is adjusted as follows.

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^k(t, t+1). \quad (2)$$

The first part of the original formula is the pheromone retaining part, ρ is the retention coefficient, while the latter part $\Delta\tau$ represents the amount of the pheromone added for each ant walking along the path.

Max-min ant system (MMAS) algorithm process can be expressed as follows:

MMAS and ACS mainly have three aspects of difference. After each cycle, only an ant pheromone update. The possible ant is the one finding the optimal solution in the current loop of ants (local optimal solution), may also the ant finding optimal solutions since the experiment starting (the global optimal solution). To avoid the stagnation of the search, the elements of each solution pheromone track are limited to a maximum and minimum range. At last, the pheromone track initialized to the maximum updates as follows:

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}(best). \quad (3)$$

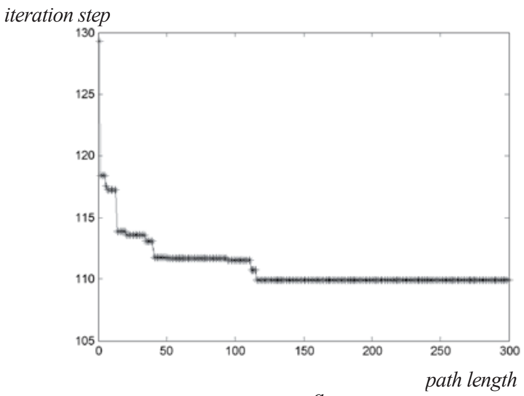
At the same time, the pheromone smoothing mechanism has also been applied to the MMAS. The MMAS is recognized to be one of the best ant colony algorithms.

The "big-small ant colony" algorithm. In view of the MMAS pheromone update methods, we proposed a "big-small ant colony" algorithm. Small ants are the ordinary ants, while the big ants are the ants that can carry a much more pheromone, and the big ants are prone to make mistakes. The big ants 0% say that, there are no big ants, only small ants; while big ants 100% say that, there are no small ants, only big ants. The article presents the experiment results of the optimal path length for 30 cities and the convergence speed with different proportion of the ants. Experimental parameters setting: volatilization coefficient is 0.5, the pheromone is 1, heuristic factor is 5, a small ant ring constant is 1, a big ant ring constants is 5, each generation contains a total of 30 ants, the number of iterations is 300. The algorithm is based on the MMAS.

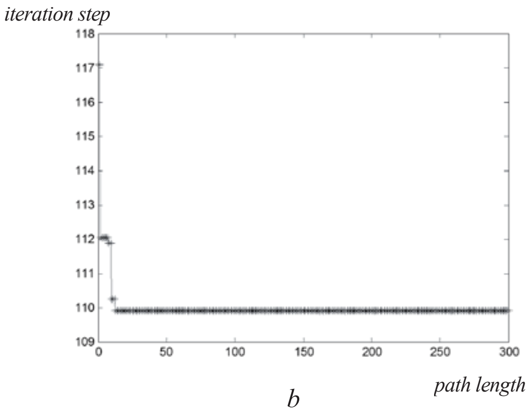
Experimental conclusion: although the big ant introduction to find better TSP path is not working, it accelerate the convergence speed a lot (Fig. 1).

Ant colony algorithm based on "jumping". In order to get the optimal solution, we proposed a novel ant colony algorithm based on "jumping". By using the original MMAS ant colony algorithm, we often can converge to the following result for the 30 city problem. The result of the path length is 135.9017, which is not the optimal result, but it most commonly occurs in the process of converging results (Fig. 2).

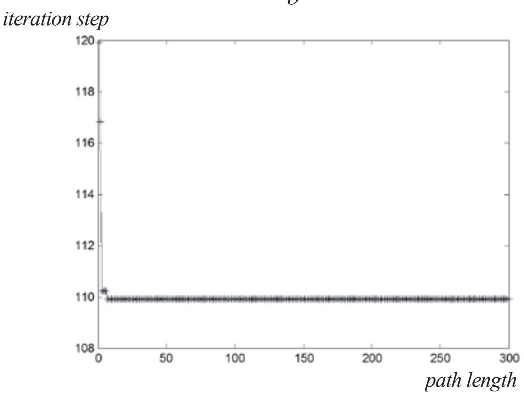
The rectangular areas surrounding small parts of the routing show the difference between the results. The rest of the routing is similar. This mutation is called a "small jump pit". But it does not result in the optimal path. In the process of debugging of MMAS algorithm, it has been found a shorter path with the length of 135.1807 (Fig. 3).



a



b



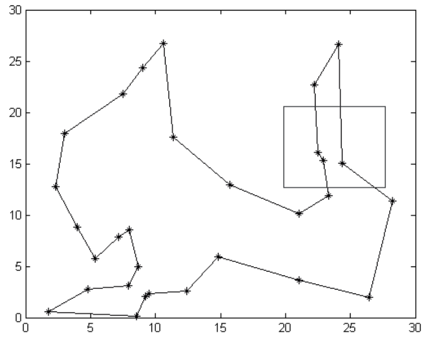
c

Fig. 1. Convergence speed curve of the "big-small ant colony" algorithm: x axe is the iteration step, y axe is the path length; a – 0 % big ants; b – 25 % big ants; c – 50 % big ants

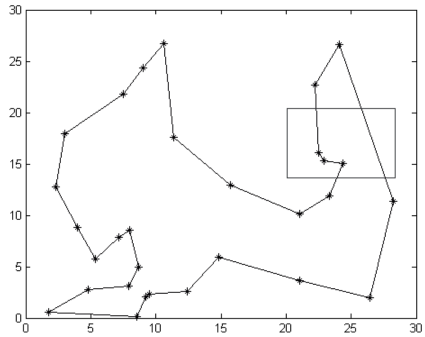
Unfortunately, the results appear only once. If we compare the result shown in Fig. 3 with the suboptimal results, they differ more than in one place. Actually, the routing by more than 50% is different; in addition, such a strong change of routing that appeared in the upper left corner of the path is almost impossible when using the MMAS algorithm.

The root cause consists in the heuristic information, which told all ants to walk as near as possible, and try not to walk to the city that is far from the current one, which can be understood as the myopic behaviour of ants, and this is inevitable in MMAS.

So we think the subprime to optimal process should be a "big jump pit", namely to change most of the routing, keep only a small number of routing. Therefore, we must add the "jump pit" in the MMAS algorithm. There are two basic classes of "jump pits": large and small, as shown in the Fig. 4.

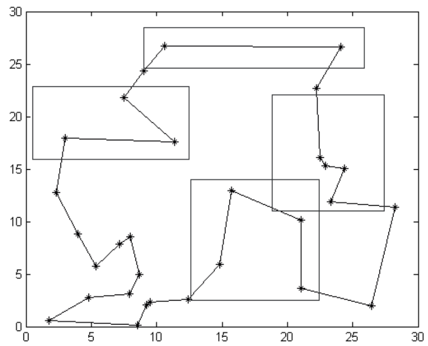


a

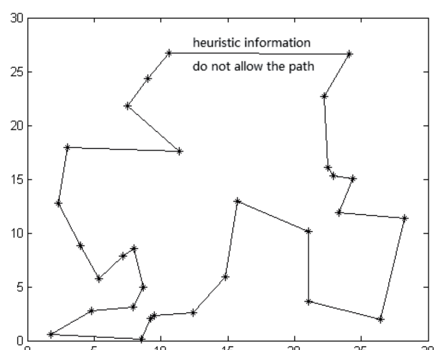


b

Fig. 2. 30 city problem suboptimal results based on MMAS: a – suboptimal result is 135.9017; b – suboptimal result is 135.6100



a



b

Fig. 3. 30 city problem optimal result: a – the jumping path; b – the heuristic information based on jumping

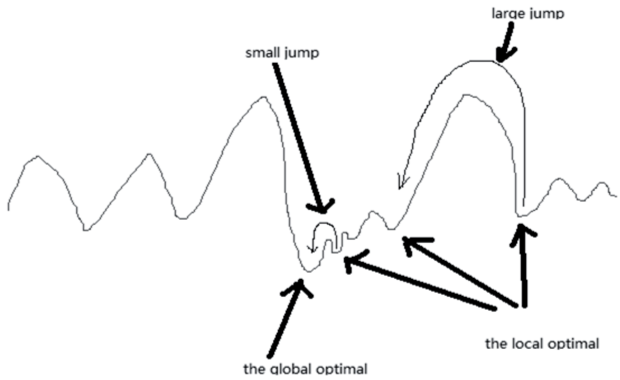


Fig. 4. The concept of a "jump pit"

In the TSP problem, we define the following concepts: referring to the "big jump pit", there is at least more than 10% of the routing path changing process, while referring to a "small jump pit", there is less than 10% of the routing path changing process. In other words, the "big jump pit" causes a big change, the change was almost beyond recognition, the "small jump pit" causes small change. The MMAS algorithm of small and medium size barely able to jump pit (based on the randomness of probability). The ACO algorithm, as a fast convergence algorithm, can be improved by the "jumping strategy". The key study problem is to design appropriate strategies of "big and small jump pit", because for many NP–hard algorithms, it is very important to design a good "jumping strategy".

Based on the analysis above, the strategies of "big and small jump pit" have been developed.

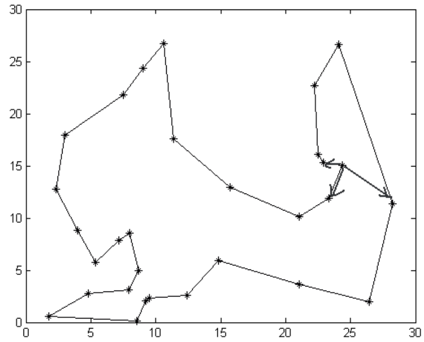
"Small jumping strategy". In each route let more than one ant starting along a different path at the same time, as shown in the Fig. 5; by this way, we can let the ants go not only to the nearest city, but also far away from the city, so as to realize "small jumping pit".

"Big jumping strategy". When the pheromone convergence, the ant must walk the red path in the perspective of probability; therefore, a barrier is designed in the middle of the red path, which is forcing the ants to search a different path.

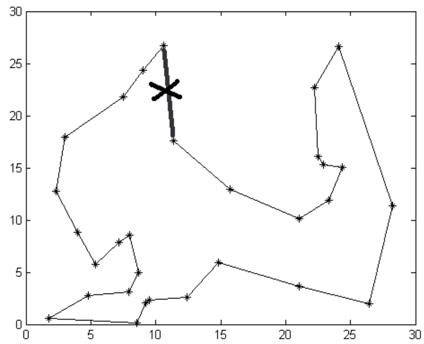
Experiments and results. The design of the experiment process: introducing 50% big ants, calculating 500 steps based on the MMAS algorithm, and then for the result of the convergence path, in turn, put obstacles in the path of each knots, at the same time solving the TSP problem with 3 ants climbed out based on the MMAS algorithm, and climbed 300 generations.

The experimental results show that the previous 500 step MMAS solving process has converged to suboptimal results, then, when the "big jump pit" was applied, it has converged to the optimal results. By placing obstacles of the "big jump pit" and forcing the ants to go along a new road, we have proved the effectiveness of the "jumping strategy". The effectiveness of the idea has been proved by the results of many standard database TSP problem solving (Fig. 6–7).

Discussion and conclusion. The "jumping pit strategy" needs to be improved, primarily the "big jumping strategy". The main problem that must be solved is the reduction of the computation time. For the city problems concerning more than 100 cities, the time spent on calculation often takes a few hours, which is unacceptable.

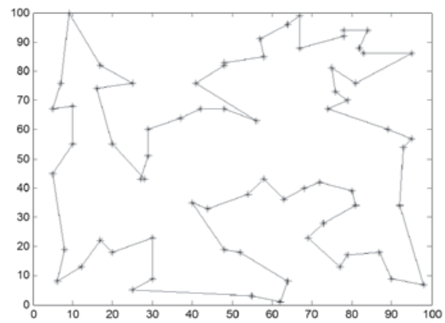


a

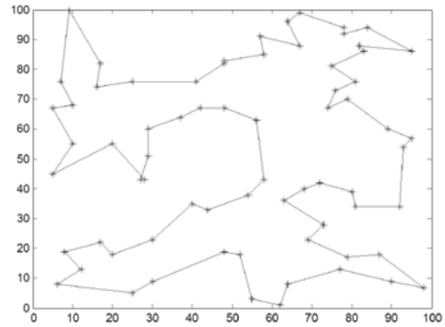


b

Fig. 5. Jumping strategies: a – "small jumping strategy"; b – "big jumping strategy"

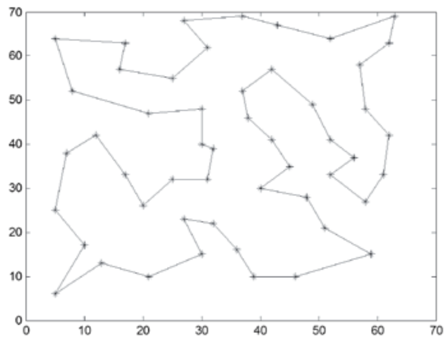


a

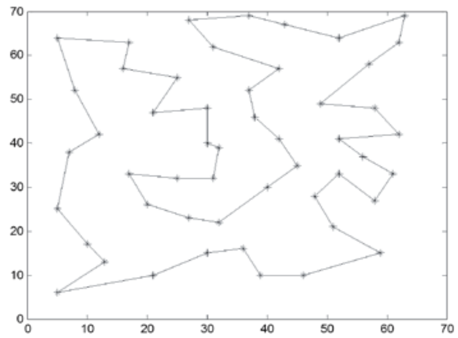


b

Fig. 6. St70 results: a – MMAS result is 696; b – result obtained by our method is 675 (optimal)



a



b

Fig. 7. Eil51 results: a − MMAS result is 428; b − result obtained by our method is 426 (optimal)

Acknowledgements. This work was financially supported in part by the National Natural Science Foundation of China (Grant No. 61401242), China Postdoctoral Science Foundation (Grant No. 2014M552045), the key scientific and technological project of Henan (152102210335), the Research Foundation for Advanced Talents of Nanyang Normal University (ZX2014058).

References / Список літератури

1. Pang, S., Ma, T., and Liu, T. (2015), "An improved ant colony optimization with optimal search library for solving the traveling salesman problem". *Journal of Computational & Theoretical Nanoscience*, vol.12, no.5, pp. 440−1444.

2. Bu, Y., Li, T. Q., and Zhang, Q. (2015), "A weighted max-min ant colony algorithm for tsp instances". *Ieice Trans Fundamentals*, vol.98, no.3, pp. 894−897.

3. Basheerjasser, M., Sarmini, M., and Yaseen, R. (2014), "Ant colony optimization (ACO) and a variation of bee colony optimization (BCO) in solving tsp problem, a comparative study". *International Journal of Computer Applications*, vol.96, no.1, pp. 1−8.

4. Camp, C. V., and Bichon, B. J. (2014), "Design of space trusses using ant colony optimization". *Journal of Structural Engineering*, vol.130, no.5, pp. 741−751.

5. Shuang, B., Chen, J., and Li, Z. (2011), "Study on hybrid ps-aco algorithm". *Applied Intelligence*, vol.34, no.1, pp. 64−73.

6. Mavrovouniotis, M., and Yang, S. (2011), "A memetic ant colony optimization algorithm for the dynamic travelling salesman problem". *Soft Computing*, vol.15, no.7, pp. 1405−1425.

7. Dorigo, M., and Blum, C. (2005), "Ant colony optimization theory: a survey". *Theoretical Computer Science*. Vol.344, no.1, pp. 243−278

8. Dorigo, M. (1997), "Ant colonies for the traveling salesman problem". *Biosystems*, vol.12, no.2, pp. 667−670.

9. Dorigo, M., and Stützle, T. (1999), "The ant colony optimization meta-heuristic". *New Ideas in Optimization*, vol.28, no.3, pp. 11−32.

Мета. Класичний „мурашиний" алгоритм оптимізації використовувався для вирішення завдання недетермінованої поліноміальної складності – „завдання комівояжера", грунтуючись на тому, що мурашки, як правило, обирають дорогу, марковану великою кількістю феромону. Алгоритм Max-min „мурашиної системи" (MMAS), переважно, приводить до найближчого сусіднього міста та завжди формулює локально оптимальне рішення.

Методика. Сформульовано алгоритм „колонії великих і малих мурашок" зі, свого роду, стратегією „стрибок через яму". При цьому великі мурашки можуть переносити більшу кількість феромону та більш схильні до помилок.

Результати. Уперше використаний алгоритм „колонії великих і малих мурашок" для збільшення швидкості сходження. Потім, за використання концепції „стрибок через яму", був реалізований пошук шляхів у більш широкому діапазоні, де стратегія „малого стрибка" дозволяє більш ніж одній мурашці вибрати альтернативний шлях, а стратегія „великого стрибка" дозволяє ставити перешкоди на дорозі, відміченій феромоном, та примушувати мурашок обирати іншу дорогу. Результати експериментів показали, що модифікований алгоритм завжди досягає оптимального результату на відміну від MMAS-алгоритму.

Наукова новизна. Вивчено новий „мурашиний алгоритм" і розглянута ефективність ідеї, що раніше не обговорювалася.

Практична значимість. Запропонований алгоритм може використовуватися для вирішення багатьох завдань, особливо у поєднанні з яким-небудь детермінованим алгоритмом для реалізації швидкої загальної оптимізації.

Ключові слова: *мурашиний алгоритм, „завдання комівояжера", Max-min мурашина система, завдання недетермінованої поліноміальної складності, глобальний пошук, феромони, стратегія „стрибок через яму"*

Цель. Классический „муравьиный" алгоритм оптимизации использовался для решения задачи недетерминированной полиномиальной сложности – „задачи коммивояжера", основываясь на том, что муравьи, как правило, выбирают путь, маркированный большим количеством феромона. Алгоритм Max-Min „муравьиной системы" (MMAS), преимущественно, приводит к ближайшему соседнему городу и всегда формулирует локально оптимальное решение.

Методика. Сформулирован алгоритм „колонии больших и малых муравьев“ со, своего рода, стратегией „прыжок через яму“. При этом большие муравьи могут переносить большее количество феромона и более склонны к ошибкам.

Результаты. Впервые использован алгоритм „колонии больших и малых муравьев“ для увеличения скорости схождения. Затем, при использовании концепции „прыжок через яму“, был реализован поиск путей в более широком диапазоне, где стратегия „малого прыжка“ позволяет более чем одному муравью выбрать альтернативный путь, а стратегия „большого прыжка“ позволяет ставить преграды на пути, отмеченном феромоном, и вынуждать муравьев выбирать другой путь. Результаты экспериментов показали, что модифицированный алгоритм всегда достигает оптимального результата в отличие от MMAS-алгоритма.

Научная новизна. Изучен новый „муравьиный алгоритм“ и рассмотрена эффективность идеи, ранее не обсуждавшейся.

Практическая значимость. Предложенный алгоритм может использоваться для решения многих задач, особенно в сочетании с каким-либо детерминированным алгоритмом для реализации быстрой общей оптимизации.

Ключевые слова: *муравьиный алгоритм, „задача коммивояжера“, Max-Min муравьиная система, задача недетерминированной полиномиальной сложности, глобальный поиск, феромоны, стратегия „прыжок через яму“*

Рекомендовано до публікації докт. техн. наук В.І. Корнієнком. Дата надходження рукопису 21.11.14.

Heng Yang

Nanyang Normal University, Nanyang, China, e-mail: yheng_sir@sina.cn

A REPLACING STRATEGY BASED LEAST RECENTLY USED ALGORITHM IN STORAGE SYSTEM

Хен Ян

Наньянский педагогічний університет, м. Наньян, провінція Хенань, КНР, e-mail: yheng_sir@sina.cn

СТРАТЕГІЯ КЕШУВАННЯ, ЗАСНОВАНА НА АЛГОРИТМІ ВИТІСНЕННЯ ДАВНО НЕВИКОРИСТАНИХ ЕЛЕМЕНТІВ У ПАМ'ЯТІ СИСТЕМИ

Purpose. Big data is a very large vocabulary we have heard recently. No matter for business or personal users, there is always a lot of important data to store. When we use the storage system, we often want the system to respond quickly enough to reach a state of no delay. This is a big challenge to the storage system. Scientists have already done many researches on this topic and they found that the use of cache in the storage system can improve the performance of storage system greatly.

Methodology. Cache algorithm is a hot research field in the current storage area. Least Recently Used algorithm (LRU) is a commonly used cache replacement algorithm.

Findings. Since the new hash value that appears atop of the stack needs to adjust the stack even if the visited page is already in memory, this takes much time. We need a better cache replacement algorithm to improve the performance.

Originality. A new replacing strategy based on the LRU algorithm has been developed; it is called Improved LRU algorithm (ILRU). It can increase the hit rate when more users suddenly have a higher access to the unfamiliar page. We determine whether the hash value is added to a page or not through searching the access hash value in the LRU queue.

Practical value. The test results show that the design of ILRU algorithm can improve the performance comparing to traditional LRU algorithm. At the same time, ILRU algorithm has higher hit rate than FIFO algorithm.

Keywords: *replacing strategy, least recently used algorithm, ILRU, hit rate, performance*

Introduction. The development of internet is very rapid. According to the U.S. Department of Commerce survey, the data on internet traffic is going to be double on average every one hundred days. The fast development brings not only the tremendous business opportunities but also creates a big problem to storage system. People will have high requirements to the storage system performance.

In recent years, people have used the cache to increase the performance of the storage system. The main function of the storage system is as a cache between the high speed equipment and the low speed device. A good cache replacement algorithm may be designed to improve the hit rate of the content. When the content of the request is hit, it can be obtained directly from the cache, which can reduce the response time of the request [1].

In order to improve the hit rate of the cache system, we should design different cache algorithms to manage the data and make out the elimination decision when the buffer space is full. The cache algorithm generally determines which data needs to be cached and the data blocks are selected when the space is full.