TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 09/25 pp625-631
Volume 13, Number 5, October 2008

Intelligent Scheduling of Public Traffic Vehicles Based on a Hybrid Genetic Algorithm*

ZHANG Feizhou (张飞舟)**, CAO Xuejun (曹学军), YANG Dongkai (杨东凯)†

School of Earth and Space Sciences, Peking University, Beijing 100871, China;
† School of Electronic Information Engineering, Beihang University, Beijing 100083, China

Abstract: A genetic algorithm (GA) and a hybrid genetic algorithm (HGA) were used for optimal scheduling of public vehicles based on their actual operational environments. The performance for three kinds of vehicular levels were compared using one-point and two-point crossover operations. The vehicle scheduling times are improved by the intelligent characteristics of the GA. The HGA, which integrates the genetic algorithm with a tabu search, further improves the convergence performance and the optimization by avoiding the premature convergence of the GA. The results show that intelligent scheduling of public vehicles based on the HGA overcomes the shortcomings of traditional scheduling methods. The vehicle operation management efficiency is improved by this essential technology for intelligent scheduling of public vehicles.

Key words: genetic algorithm (GA); hybrid genetic algorithm (HGA); intelligent transportation system (ITS); intelligent scheduling; public traffic

Introduction

Urban public transportation systems seek to meet the increasing demands of all kinds of passengers while earning profits with high quality of service (QoS) using a limited number of public vehicles. The contradiction between the supply and demand of public vehicles has existed for many years because the demands are increasing while the vehicle resources cannot be expanded infinitely. The scheduling of public vehicles seeks to manage vehicle resources effectively by reasonably allocating limited resources, with better results realized through an optimized schedule. Since an optimized scheduling scheme is very complex, the operating environment is often simplified in practical systems. The results then fail to meet the real time needs when the scheduling scheme is not fully optimized[1,2]. In some cases, such problems have been successfully addressed by the use of interactive decision support systems to generate "good quality" solutions[3]. Some local search techniques such as the multi-start descent methods using simulated annealing (SA), the tabu search (TS), and other neighborhood exploration methods have been successfully applied in some vehicle routing systems[4,5]. Despite their flexibility and relatively high efficiency, strategic support systems and local searches tend to require extensive manual intervention and/or reevaluation of candidate solutions. Although these disadvantages are sometimes overcome by efficient schemes that evaluate perturbations of existing solutions, this method is not valid in all cases. Vehicle transportation itself is a nonlinear programming (NP) problem that is difficult to schedule[6], since candidate solutions are costly to evaluate and the potential for user insight is limited.

Many researchers have introduced the genetic algorithm (GA) into schemes to manage the operation and

Received: 2006-10-08; revised: 2007-01-09
* Supported by the National High-Tech Research and Development (863) Program of China (No. 2004AA133020)
** To whom correspondence should be addressed.
E-mail: zhangfz@pku.edu.cn; Tel: 86-10-62753281

scheduling of public vehicles to obtain a reasonable and effective scheduling scheme even in actual complex environments[6-11]. However, different problems need different object functions and genetic operators. The GA has been used to obtain optimal solutions for demand responsive services by public taxis[6,7]. The GA has also been used for vehicle scheduling solutions when traffic is disrupted[8,9]. Benyahia and Potvin[10] used the fleet routing to carry passengers to different locations. Wang[11] elaborated on the architecture of information acquisition and infusion for dynamic routing, but did not analyze the routing algorithm. The GA has a strong search capability in large search space but its search capability decreases in a small space. It can quickly arrive at 90% of an optimized solution, but it takes a very long time to arrive at 100% of the optimizing solution[12].

Urban public vehicle scheduling in China generally operates on fixed routes, with the operating mode of each vehicle changeable to meet different demands during the day. Not every vehicle stops and services at all bus stops and not every vehicle needs to run the whole route. The scheduling task is to assign the operating mode and order of each vehicle to obtain the best QoS level. Currently, vehicle scheduling is still mostly performed by experts because the task is very complex and requires high-level cognitive abilities. There are no existing computer aided solutions to maximize the service level and the real time route conditions are not considered, i.e., existing schedule strategies are not automatic or dynamic. This paper takes Beijing as an example and analyzes the public vehicle scheduling problem using a GA and a hybrid GA (HGA).

1 Public Vehicle Scheduling

1.1 Objective function

Vehicle scheduling problems can be studied as sort order or resource distribution problems theoretically and as transportation task planning and management in practice. That is, the scheduling problem invokes the time distribution of shared resources. Usually, the scheduling method uses an object function for restricted conditions as an equation or inequality. The restriction condition generally involves resource restrictions in sort order restrictions. The objective function can be the time span, time delay, or resource utilization efficiency. For most urban public transportation systems, which are stochastic service systems, the vehicle scheduling can be solved as a sort order problem.

In Beijing, the public buses run mainly as a full-course bus, an intersection bus, or a main-stop express bus. Let M be the entire vehicle set and J the bus stop set, and each bus should arrive at its stops on time, especially the main stops. For example, if there are N vehicles serving one route, the bi-directional (from original stop to the final stop and vice versa) vehicle set is represented as $M=\left\{m \middle| m=1,2,...,2\times N\right\}$. There are P bus stops along the route, $J=\left\{j \middle| j=1,2,...,P\right\}$. If one vehicle runs the full course, it will stop at all P stops. If one vehicle runs in the intersection mode, it will only stop (i.e., service) at the first Q ($Q<P$) stops. The express bus will only stop at some important stops (define $O<P$ stops in advance and include the final stop P) along the whole route.

Public transportation services in Beijing always seem to have long intervals between buses providing passenger service. Therefore, the passengers must wait a long time for their incoming bus. Then sometimes two buses get very close to each other so that the later one does not pick up any passengers. In both scenarios, the QoS is not satisfactory and the vehicle resources are not utilized efficiently and effectively. To obtain a high QoS level and to efficiently utilize the resources of public vehicles, the operating mode and order for each vehicle must be arranged according to an optimization index. Also, the operation time for each vehicle should be as short as possible to enable the passengers to arrive at their destinations within a reasonable time. Statistically, the optimization index for the vehicle scheduling problem seeks to minimize the operating time according to the ratio of vehicles used during slack and rush times. Thus, the objective function can be defined as

$$\min T_0=\sum_{m\in M}\max T_i+\rho\sum_{j\in J}\max(0,\ t_{s,j}-t_{d,j}) \quad (1)$$

where T_0 is the operational vehicle turnover time from the first vehicle leaving to the N-th vehicle returning, T_i represents the execution ratio of the time for the m-th vehicle to complete the whole transportation task (stop and service O, P, or Q stops), $t_{s,j}$ and $t_{d,j}$ are the service times at stop j and the stated service time, and ρ is weighting coefficient which depends on the operating time. Expert experience suggests that the weighting coefficient follows a stochastic Gauss distribution during slack times and a fixed value during rush hours.

1.2 Genetic coding

For public vehicle scheduling in Beijing, the execution task sequence for the vehicle m can be expressed as

$$\mathrm{OP}(m)=\{O_{m,J(1)},\ O_{m,J(2)},\ \cdots,\ O_{m,J(X)}\} \quad (2)$$

where $O_{m,J(j)}(j=1,2,...,X)$ stands for vehicle m serving bus stop $J(j)$ and $J(X)$ stands for the number of bus stops where vehicle m serves. The basic conditions are: If X=P, then $J(j)$=j and the operation mode is the full course coded as 0; If X=Q, then $J(j)$=j and the operation mode is the express mode coded as 1; If X=O, then $J(j)\neq j(j\neq 1)$ and the operation mode is the intersection mode coded as 2.

With this problem description, the vehicle scheduling solution can be expressed by chromosomes,

$$\mathrm{SOP}=\{\mathrm{OP}(m)\,|\,m\in M\} \quad (3)$$

Each chromosome is called an individual in the GA.

1.3 Genetic operation

In Eq. (3), all $\mathrm{OP}(m)$ $(m=1,2,...,2\times N)$ are genes. The gene position in the Beijing public vehicle scheduling problem is also a sort-order problem since the gene itself will not be changed in the genetic operations. One-point crossover, two-point crossover, and interchange operators were used for the gene operations.

1.4 Fitness value calculation

The optimization index, T_0, of each chromosome can be calculated on the basis of the scheduling results with its sequential fitness value, f. However, Eq. (1) requires the minimum performance index, T_0, which should be the maximum fitness value in the GA. Therefore, the optimization index, T_0, of each chromosome is mapped into a fitness value using the transpositional function

$$f=cT_0+d \quad (4)$$

where c and d are constants, selected as described by Chen et al.[12] The selection of c and d should make the fitness value map into the region $[C,D]$ after the population scale initialization. In this way the difference between fitness value among populations will be preserved and the individual with the minimum fitness value has an opportunity to participate in the competition.

1.5 Scheduling algorithm procedure

The scheduling algorithm is described as follows.

Step 1 Stochastically produce an even number of solutions to act as the initial population, storing the optimal individual and its setting as the evolutional generation number n_e=0.

Step 2 If the evolutional generation number n_e is larger than a given value, or the performance index of the optimal individual reaches a given value, the evolution process ends. Otherwise, continue to Step 3.

Step 3 Select individuals from the population pool according to the selection probability which is directly proportional to their fitness value. The individuals are selected for crossovers and mutations in order to produce a new generation population. The crossover operation is performed for genes at the same position within two chromosomes. The mutation operation is also performed for genes within the chromosome. The feasibility of results is checked after the genetic operation with n_e= n_e+1.

Step 4 Calculate the fitness value as the evaluation standard for each individual.

Step 5 Replace the worst child individual with the best father individual based on the super excellence strategy.

Step 6 If the fitness value of the optimal individual in this generation is larger than that of the optimal individual in the corresponding father generation, set it as the optimal individual. Continue to Step 2.

The outbreeding strategy is applied in the evolution process to avoid inbreeding. The crossover will only be performed with two different individuals.

2 GA Applications

Heuristic rules are applied to the initial population to obtain solutions and improve the evolution efficiency. With regard to the vehicle scheduling problem, one-point and two-point crossover operations are used in the GA. The basic performance with these two crossover operations are analyzed based on three different problem scales with N=12, N=18, and N=21. Table 1 and Fig. 1 compare the results with these two genetic operations in terms of the computational time and the convergence speed.

Table 1 Computational times for the two crossover operations

Problem scale ($2\times N$)	Crossover type	Minimum value		Average value		Maximum value	
		Objective function (min)	Computing time (s)	Objective function (min)	Computing time (s)	Objective function (min)	Computing time (s)
24	One-point	287.36	0.73	287.36	0.73	287.36	0.73
	Two-point	287.36	0.94	287.36	0.94	287.36	0.94
36	One-point	367.62	0.80	375.42	0.82	383.22	0.84
	Two-point	393.07	1.13	412.50	1.20	421.93	1.25
42	One-point	428.23	0.89	434.94	0.91	451.65	0.93
	Two-point	468.23	1.47	418.70	1.49	495.27	1.59

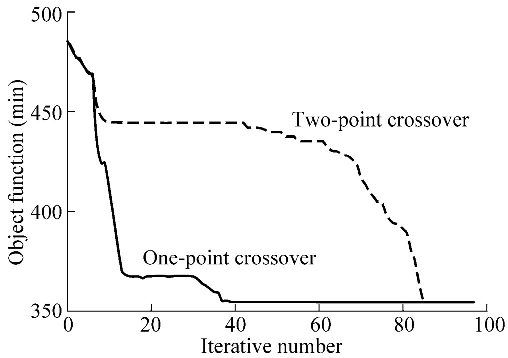

Fig. 1 Convergence speeds with *N*=18

As shown in Fig. 1 and Table 1, the algorithm needs about 40 generations and 0.82 s to reach the optimal solutions with the one-point crossover operation. The two-point crossover operation needs about 85 generations and 1.20 s to reach the optimal solution.

3 Hybrid Genetic Algorithm

3.1 Description

The GA has a strong search capability in the large search space but its search capability is reduced in a small space. Therefore, many methods have been introduced into the GA to improve the local search performance such as intercross restriction operators and combining the GA with traditional search algorithms such as SA and TS. The introduction of SA or TS into a GA to create an HGA not only enhances the local optimization of the GA, but also extends the local search areas of the SA and TS algorithms and strengthens the global optimization of the pure SA and TS algorithms[4,5,13,14].

There are three ways to integrate the GA with other optimization methods.

(1) Illuminating regulations The GA starts from an initial population string. Usually, the initial population strings are created randomly. However, the performance of the GA is influenced greatly by the primary solutions and the quality of solutions depends on the primary fitness function value. Therefore, initial population strings should be created by using illuminating regulations.

(2) Filtering procedure In the pure GA, the competition is carried out within each child generation with no competition between father and child generations. Therefore, the excellent genes in the father generation may be lost when transmitted to the child generation. The optimum group solutions are put directly into the next generation in some algorithms to preserve such genes. However, this can cause premature convergence. In addition, since the GA uses random crossover and mutation operators, the transmitted individuals are not all excellent ones. As a result, the initially excellent individuals can be damaged and the algorithm performance can be diminished. The preservation of excellent child generation individuals is guided by certain rules. Each child generation that cannot be preserved will be filtered. When the value of the fitness function is less than that of the father generation, the acceptance probability is taken as the survival probability of the weak child generation[10].

(3) Cultivation procedure The single optimization of certain individuals within the group to make it acquire more excellent characteristics after several generations of evolution is called cultivation. The "excellent generation" after the cultivation is returned to the original evolution procedure to continue the genetic evolution[14]. The TS, SA, or other methods can be used during the cultivating procedure.

The primary operations in the hybrid genetic algorithm are[15]:

$$\text{GA:}\ \ \text{POP}_k \xrightarrow{\text{Selection}} \text{Parents}_k \xrightarrow{\text{Crossover+mutation}} \text{POP}_{k+1};$$

$$\text{HGA:}\ \ \text{POP}_k \xrightarrow{\text{Selection}} \text{Parents}_k \xrightarrow{\text{Crossover+mutation}} \text{POP}_{\text{temp}} \xrightarrow{\text{Filter}} \text{POP}_{k+1},$$

$$\text{POP}_k \xrightarrow{\text{Selection}} \text{Parents}_k \xrightarrow{\text{Crossover+mutation}} \text{POP}_{\text{temp}} \xrightarrow{\text{Cultivation}} \text{POP}_{k+1}.$$

3.2 Tabu search algorithm

The tabu search is a meta-strategy iterative procedure for building extended neighborhoods with particular emphasis on avoiding being caught in a local optimum. TS was initially proposed by Glover[16]. The search space is explored by moving from a solution to its best neighbor to increase the likelihood of moving out of a local optima. Successive neighbors of a solution are generated and the corresponding objective function is evaluated since the results are sometimes worse than the previous solution. To avoid cycling, solutions that were recently examined are forbidden or declared tabu for a certain number of iterations. A move made in iteration X is called tabu until iteration $(X+\theta)$ where θ is the tabu duration randomly chosen on a pre-specified interval. Thus, the tabu list is an ordered queue containing forbidden moves. Whenever a move is made, it is inserted into the end of the tabu list and the first element from the list is removed. The best admissible move is then chosen as the highest evaluation move in the neighborhood of the current solution in terms of the objective function value and the tabu restrictions.

The key to the enlightening optimizing method is to decide the size of the tabu list. If the tabu list is too small, invalid iterations are not avoided, which will affect the optimizing function of the TS over the whole search area. If the tabu list is too large, the list will not only increase the computing complexity in time and space, but will also make it difficult for the TS to approach the vicinity of the optimum solution in the whole area.

3.3 GA-TS hybrid genetic algorithm

The TS algorithm solves hybrid optimization problems very efficiently because it repeatedly searches in the vicinity of current solutions and moves quickly in the optimization direction with high probability. However, the TS algorithm has adjustable parameters that have to be adjusted during the calculations. Since the parameter selection directly affects the final solutions the TS is not robust. The GA is used to adjust not a single solution but several limited parameters. Therefore, the combination of the GA and the TS gives a hybrid genetic algorithm with better performance than each one separately. When introduced into the GA, the TS is used to optimize all the individuals in the group in the "cultivating procedure". Then the mutation operation is used to acquire more optimum characteristics to reach the optimum result. A flow chart of the GA-TS HGA is shown in Fig. 2[17].

$$\text{POP}_k \xrightarrow{\text{Selection+Crossover}} \text{Parents}_k \xrightarrow{\text{TS-cultivation}} \text{POP}_{\text{temp}} \xrightarrow{\text{Mutation}} \text{POP}_{k+1}.$$

The GA-TS HGA has four basic steps.

(1) Initialization: several feasible solutions unrelated to each other are created randomly to get primary feasible solution.

(2) Genetic operations: selection and crossover are aimed at the group of feasible solutions to get a locally optimum solution.

(3) TS-searching: The locally optimum solutions in Step 2 are succeeded to get the optimum local solution.

(4) Mutation operation: The individuals from the optimum local solution in Step 3 are calculated to get new individuals.

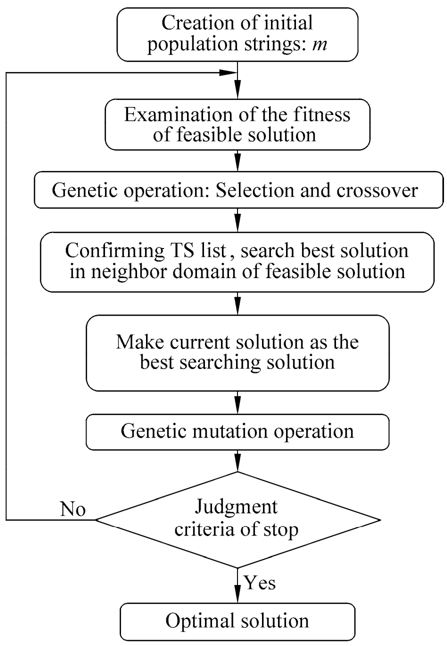

Fig. 2 Flow chart of GA-TS HGA

Steps 2, 3, and 4 are repeated until the algorithm stop criterion is met.

The efficiency of the GA-TS hybrid genetic algorithm for the optimal public vehicle scheduling problem was evaluated by comprising the GA-TS HGA and the pure GA separately to the optimal scheduling of 18 vehicles (N=18). The convergence speeds and optimizing qualities of the hybrid GA-TS genetic algorithm and the simple GA are shown in Fig. 3.

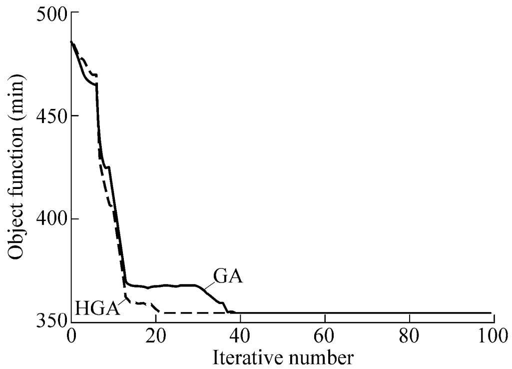

Fig. 3 GA-TS HGA and GA results

The results show that the GA-TS HGA possesses stronger convergence characteristics and moves quicker to a good search domain than pure GA. Thus, the number of generations is reduced and the premature convergence of the GA is avoided, which improves the scheduling performance.

4 Results and Analyses

The route 375 buses in Beijing urban public transportation system are taken as an example using the intelligent GA-TS HGA. Since the bus system is an open, dynamic system, many factors influence and restrict the operating speed of the buses. Statistical analyses of the urban traffic flow in Beijing show that the public vehicle traffic during slack times is basically steady but that the buses face traffic jams during rush hours. Different weights were used to model the values situations. The following simulations were for slack times with the weights ρ chosen as a stochastic Gauss distribution.

There are 18 vehicles on Route 375 in Beijing, i.e., N=18 with full-course buses (≈70%), intersection buses (≈15%), and express buses (15%). The line has 18 bus stops and the travel time from origin to destination is about 50 min. The vehicles are empty before starting or after working hours, so their operation is zero. Assuming that the vehicles use the same time between bus stops and stay at each stop for the same length of time, the operating parameters are then a scheduling interval of 8 min, the time at each stop of 0.25 min, and the run time between stops of 2.9 min.

To avoid propagation among close relatives during evolution, the intercross (or interchange) strategy was used before the intercross operation between two individuals in the father generation, and the individuals were comprised. If the two individuals were not the same, the intercross operation was permitted. If they were the same, then one of the two individuals was replaced randomly to create a new individual. For this scheduling problem, the HGA population scale was 60, the one-point crossover probability was $P_c=0.75$, and the mutation probability was $P_m=0.005$. The scheduling scheme for the 18 vehicles gave the result:

000010021001002002001000020000100120

The detailed information above shows that if the vehicular scheduling interval is 8 min, then 13 vehicles are enough for one turnover and the total time is about 366.3 min. If 18 vehicles are used, the scheduling interval will be 5.5 min. Thus, the optimal scheduling scheme obtained based on the GA-TS HGA can reasonably allocate public vehicle resources and improve transportation efficiency.

5 Conclusions

Intelligent scheduling methods for public traffic vehicles were developed using the GA and the HGA. The intelligence characteristics of GA improve the operating efficiency of the public vehicle system. The premature consequence of the GA was avoided by the introduction of the TS algorithm into the HGA to improve the convergence and optimization characteristics. The results show that the intelligent scheduling algorithm based on the combination of the intelligence technology and scheduling methods greatly improves the performance of the scheduling system in terms of shortening the turnover time and balancing the vehicle resources.

References

[1] Marian T B. Development of a transportation data processing system for metropolitan. In: Proceedings of the IEEE-IEE Vehicle Navigation and Information Systems Conference. Washington D.C., USA，1993: 186-190.

[2] Luc B, Stefan S. Genetic algorithms：Theory and application. *International Journal of System Science*, 1997, **38**(2): 13-23.

[3] Benyahia I, Potvin J Y. Decision support for vehicle scheduling using genetic programming. *IEEE Transactions on System，Man，and Cybernetics*, *Part A*: *Systems and Humans*, 1998, **28**(3): 306-314.

[4] Cai Yanguang, Qian Jixin, Sun Youxian. Simulated annealing for multiple demands vehicle routing problems. *Decisions and Decision System*, 1996, **l6**(4): 126-131. (in Chinese)

[5] Guly B, Demet O. A tabu search algorithm for vehicle routing problem. *Computer & Operation Research*, 1999, **26**(3): 255-270.

[6] Dessouky M, Adam S. Real-time scheduling rules for demand responsive transit systems. In: Proceedings of 1998 IEEE International Conference on Systems, Man, and Cybernetics. San Diego, Canada, 1998, **3**: 2956 -2961.

[7] Uchimura K C, Takahashi H, Saitoh T. Demand responsive services in hierarchical public transportation system. *IEEE Transactions on Vehicular Technology*, 2002, **51**(4): 760-766.

[8] Fayech B, Hammadi S, Maouche S. Partial network reconfiguration through a dynamic vehicle scheduling approach. In: Proceedings of 2002 IEEE International Conference on Systems, Man and Cybernetics. Boston, USA, 2002, **4**: 6-9.

[9] Fayech B, Hammadi S, Maouche S, Borne P. Urban bus traffic regulation by evolutionary algorithms. In: Proceeding of 2001 IEEE International Conference on Systems, Man, and Cybernetics. Tucson, AZ, USA, 2001, **2**: 1316-1322.

[10] Benyahia I, Potvin J. Generalization and refinement of route construction heuristics using genetic algorithms. In: Proceedings of 1995 IEEE International Conference on Evolutionary Computation. Perth, WA, Australia, 1995, **1**: 39-43.

[11] Wang Y N, Thompson R G, Bishop I. A GIS based information integration framework for dynamic vehicle routing and scheduling. In: Proceedings of 1999 IEEE International Vehicle Electronics Conference. Changchun, China, 1999, **1**: 474-479.

[12] Chen Guoliang, Wang Xifa, Zhuang Zhenquan, et al. Genetic Algorithm and Its Application. Beijing: People Post Press, 1996. (in Chinese)

[13] Jeong I K, Lee J J. Adaptive simulated annealing genetic algorithm for control applications. *International Journal of Systems Science*, 1996, **27**(2): 241-253.

[14] Glover F, Kelly J P. Genetic algorithms and tabu search: Hybrids for optimization. *Computers & Operations Research*, 1994, **22**(1): 111-134.

[15] Mantawy A H, Youssef A M, Selim S Z. Integrating genetic algorithms, tabu search, and simulated annealing for the unit commitment problem. *IEEE Transactions on Power Systems*, 1999, **14**(3): 829-836.

[16] Glover F. Heuristic for integer programming using surrogate constraints. *Decision Science*, 1977, (8): 156-166.

[17] Weng Miaofeng. Job-shop scheduling with a combination of evolutionary and heuristic methods. *Information and Control*, 1999, **28**(2): 81-85. (in Chinese)