Design of College Discrete Mathematics Based on Particle Swarm Optimization

Jiedong Chen[1, a]

[1]Automotive Department, Guangdong Industry Technical College, China

[a]176783107@qq.com

Keywords: Discrete mathematics, Particle swarm optimization, Exchange of information, Mutation operator

Abstract: Aiming at exploration and development capacity, it is usually hard to achieve effective utilization and balance by using just one algorithm, which will influence the solving accuracy and efficiency of the algorithm accordingly. This thesis applies particle swarm optimization (PSO) to discrete multi-objective mathematical optimization, and proposes a discrete multi-objective optimization algorithm based on PSO. This algorithm adopts binary mechanism to achieve the position vector of the particles. Meanwhile, it builds a non-dominated solution set to storage the researched non-dominated solution, so as to increase the diversity of non-dominated solutions. The introductions of genetic algorithm crossover, mutation operator replacing particle velocity and location updating have reduced the computational complexity. In the thesis, it sets up mathematical models for these two problems respectively, and the experimental results show that the algorithm is effective.

Introduction

Discrete mathematics is defined to be the core backbone curriculum of computer science by IEEE in 1977. In 2004, among the five professional training plans related to computer with tutorial certification, discrete mathematics is the significant core curriculum of three majors, which are Computer Engineering (CE), Computer Science (CS) and System Engineering (SE) [1,2,3]. As the preceding curriculum of specialized courses, such as data structure, fundamentals of compiling, database principle, operating system and artificial intelligence, discrete mathematics not only needs to provide necessary basic knowledge, but develop the students' abstract thinking ability and logical thinking ability, so as to further strengthen their program design ability. The constrained optimization problem of unlinear discrete variable is the most common problem in engineering practice. Due to its non-convexity, there exists a great deal of difference between global optimal solution and locally optional solution. In order to solve global optimization problem, people try to find a stochastic optimization algorithm which has a lower requirement or even no requirement to objective function and constraint function. The evolutionary algorithms [4,5]based on biological evolutionism and hereditism and the swarm intelligence algorithm[6]based on biotic community behavior are proved to be effective.

Based on particle swarm optimization, the thesis introduces some experimental designs appropriately in teaching process of discrete mathematics, which is not only a good verification of basic theory of discrete mathematics, but builds up the students' operation ability of computer language and foreshadows the learning of other curriculum.

Design model and discrete variable of discrete mathematics

Design model of discrete mathematics

The relation in the discrete mathematics can be represented by matrix method. Matrix can be stored by a two dimensional array in a computer. The establishment and storage of array are introduced in computer language; therefore, this part will not be restated in this thesis, but focuses on the discussion of the implementation of the algorithm. In this thesis, we assume Relation1 (R1) and

Relation2 (R2) are both binary relations in the set X, where there are n elements in X. And we assume the relationship matrix of R1 and R2 are M1 and M2. They can not be directly obtained from the matrix diagram, therefore, we can realize the judgment of transitivity through the transitive closure of relations, and the realization of transitive closure needs compound operation of the relations. Thus, the algorithm designs of relationship complex operation and closure operation should be given firstly.

Set R1 and R2, and calculate relationship matrix M of composition relation R of R1 and R2:

(1) Suppose i=1, j=1；

(2) Calculate $\mathrm{M(i,j)} = \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{M1(i,k)} \times M2(k,j)$ through logic multiplication and logic plus;

(3) j=j+1, if j≤n, transfer to (2); if not, transfer to (4)；

(4) i=i+1, if i≤n, transfer to (2); if not, stop calculation.

If $\mathrm{R} = \bigcup_{i}^{n} R_i$, R has the transmission, where R^{i} represents the i times compound operation of R. Therefore, we can calculate it through calling the compound operation of relations.

Dealing with discrete variable through mapping function

In this thesis, the three particle swarm optimization algorithms (PSO, PSOPC and HPSO) all use mapping function method to deal with the discrete variables. After sorting with a certain rule, the discrete variable set $\mathrm{S_d}$, which has P discrete variable plate thickness, can be show as follows:

$$\mathrm{S}_d = \{X_1, X_2, \cdots, \ \mathrm{X}_j, \cdots, \ \mathrm{X_p}\}, 1 \le j \le \mathrm{p} \quad (1)$$

Suppose a mapping function and use the serial number to replace the discrete value in Sd, that is, h (j) = $\mathrm{X_j}$. Select its serial number to replace the specific value of discrete variables to begin to search, and try to make its searching value as continuous as possible, so as to improve the efficiency of the search. For example, n particles are in D-dimensional searching space, the position of the i particle can be represented as vector $\mathrm{X_i}$, that is：

$$\mathrm{X}_i = \{X_i^1, X_i^2, \cdots, X_i^d, \cdots, X_i^D\}, 1 \le d \le D, i = 1, \cdots, n \quad (2)$$

Namely, $\mathrm{X_i} \in \{1,2, \cdots,\mathrm{p}\}$, through mapping function h (j), the corresponding set of discrete variable is $\{\mathrm{X_1,X_2,X_3,\ldots,\ Xp}\}$.

Through mapping, particle can only search in integer space of the whole searching space; that is to say, all the components of vector $\mathrm{X_i}$ are all integers.

Improved PSO based on discrete mathematics model

(1)Optimized mathematical model of discrete mathematics

Mathematical model of discrete multi-objective optimization is showed in formula (3).

$$\min f(x) = [f_1(x), f_2(x), f_3(x), \cdots, f_m(x),] \quad (3)$$

On the basis of dispersion multi-objective optimization of particle swarm optimization (DMOPSO) multi-objective particle swarm optimization algorithm, we adopt the mechanism of binary particle swarm optimization algorithm to calculate discrete variables. In this way, particle search space correspondingly converses to N-dimensional binary space, marked as $\mathrm{B^n}$. In order to apply particle swarm optimization algorithm discrete variables optimization, we make the position vector of each particle in that algorithm belong to binary space (binary strings composed of 0 and 1). However, its velocity vector still belongs to real number space $\mathrm{R^n}$， that is to say, $\mathrm{v} \in [\mathrm{-v_{max}, v_{max}}]$. In this way, the algorithm can preserve the speed formula in the basic particle swarm algorithm. While only modifying its position update formula, the speed and position updating formula of each particle are as follows:

$$v_{ij}(t+1) = w \times v_{ij}(t) + cl \times r1 \times (pbest_i - x_{ij}(t)) + c1 \times r1 \times (pbest_i - x_{ij}(t)) \quad (4)$$

$$x_{ij}(t+1) = \begin{cases} 0 & else \\ 1 & rand < S[v_{ij}(t+1)] \end{cases} \quad (5)$$

Where, rand is a random number between [0, 1], and the other parameters in the algorithm are the same as the parameters in the basic particle swarm optimization algorithm. The probability that a particle will be taken as 1 or 0 is determined by the current speed value of the algorithm is:

$$P[x_{ij}(t+1)=1] = S(v_{ij}(t+1)) \quad (6)$$

That is, the probability selection parameter S is determined by the particle speed within the range of [0，1]. If S is close to 1, the particle may be selected as 1; if S is close to 0, the particle is more likely to be selected as 0. Kennedy and other people has used Sigmoid function to obtain the parameters. Sigmoid function is a common non-linear action function used in neural network, it expression formula is :

$$\text{s} = S(v(it)) = \frac{1}{1+e^{-v_{ij}(t)}} \quad (7)$$

(2)Minimum position value rule

Update the position of each element in the arrangement of with a continuous PSO:

$$x_{id}^{(t+1)} = x_{id}^{(t)} + w(x_{id}^{(t)} - x_{id}^{(t-1)})) + c_1 r_1 (p_{id}^{(t)} - x_{id}^{(t)}) + c_2 r_3 (p_{id}^{(t)} - x_{id}^{(t)}) \quad (8)$$

The new particle position of real-value can be obtained by the iteration of that formula (intermediate result). Then, with the principle that elements of the minimum value possess the earliest scheduling, conduct an arrangement through transforming the intermediate result to its corresponding solution. The specific steps are as follows:

Assume the updated position of the particle to be real vector $\mathrm{X}_i = \{x_{i1}, x_{i2}, \cdots, x_{in}\}$, and its corresponding solution is arranged as an integer vector $\prod_i = \{\pi_{i1}, \pi_{i2}, \cdots, \pi_{in}\}$. With the principle that elements of the minimum value possess the earliest scheduling, confirm $\prod_i$ through X_i . Assume intermediate result $\mathrm{X}_i = (1.45，3.54，2.67，2.29，\text{-}4.02)$ .Its minimum value is -4.02, which is corresponding to the fifth element of X_i. Therefore, the fifth value is the earliest one to be dispatched, which means π_{i1}=5. The second smallest value is -3.54, it corresponds the second element of π_{i1}. Thus, the second value is the second one to be dispatched, which means π_{i2}=2. Calculating in this way, we obtain the solution arrangement $\prod_i(5,2,4,1,3)$. At last, conduct inverse operation to $\prod_i(5,2,4,1,3)$, and we can obtain the integer vector $\mathrm{X}_i(4,2,5,3,1)$ of the corresponding new particle position.

(3)Initialized particle swarm

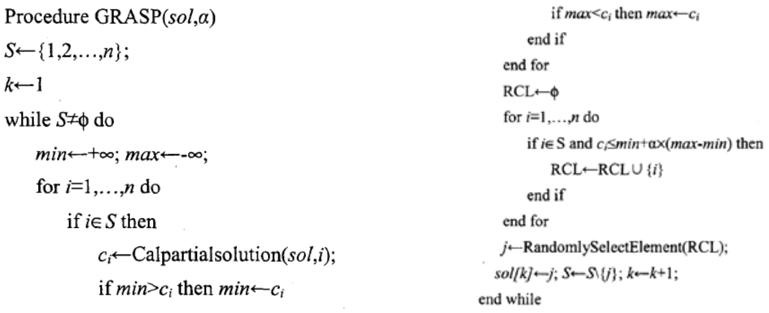
Various solutions producing method involves two stages: one is to use Greedy Randomized Adaptive Search Procedure (GRAsp) to produce initial solution; the other is to use local search algorithm to explore the field of its initial solution and find the most advantages local point. Repeat these two stages and finally produce P size different solution arrangement, which makes up solution set P。 P size should be larger than the size of swarm b of particle swarm. Next, select b_l optimal solution from this P size different solution to put into the initial swarm of particle swarm, and then delete them from solution set P. In the end, repeat the following steps for b: times (b_2=b-b_l):

A. Newly calculate the minimum value of the distance between each of the surplus solution of P and all solutions in the initial swarm of current particle swarm.

B. Select the solution with the minimum distance from the solution of P, and put it into the initial particle swarm.

C. Delete the selected solution from P.

Chart 1 is the pseudocode of GRAsp, whose process of obtaining the solution is as follows: Assume the solution is composed of many solution elements (for example, the solution of LOP problem is the integer between 1-n). According to a sort of heuristic criteria, calcuate an evaluation value for each solution elements out of the partial solutions. It represents the good degree of adding this solution element to partial solution in the current situation, and the partial solution elements with high evaluation value constitutes a restricted candidate list. And then, randomly select a solution element from RCL to partial solution. Keep on repeating this process, until the structure of the solution is finished. And the partial solution is the null solution at the beginning.



```
Procedure GRASP($sol,\alpha$)
$S \leftarrow \{1,2,\ldots,n\}$;
$k \leftarrow 1$
while $S \neq \phi$ do
    $min \leftarrow +\infty$; $max \leftarrow -\infty$;
    for $i=1,\ldots,n$ do
        if $i \in S$ then
            $c_i \leftarrow$ Calpartialsolution($sol,i$);
            if $min > c_i$ then $min \leftarrow c_i$
            if $max < c_i$ then $max \leftarrow c_i$
        end if
    end for
    RCL $\leftarrow \phi$
    for $i=1,\ldots,n$ do
        if $i \in$ S and $c_i \leq min + \alpha \times (max - min)$ then
            RCL $\leftarrow$ RCL $\cup \{i\}$
        end if
    end for
    $j \leftarrow$ RandomlySelectElement(RCL);
    $sol[k] \leftarrow j$; $S \leftarrow S \setminus \{j\}$; $k \leftarrow k+1$;
end while
```

Chart 1. GRASP Pseudocode

(4) Local search algorithm

Adopt the local search algorithm, which is on the basis of hisert mover. The defination plays the role in hisert mover operation INSERT-OVE(π_{j}, i) of solution arrangement $\prod = \{\pi_1,\pi_2,\cdots,\pi_n\}$, for example: firstly, it deletes the j element π_{j} from solution arrangement n, the j'+1~n element of solution arrangement $\prod$ moves forward for one step. Then, insert this element into the i position of solution arrangement, the original i~n-1 element of solution arrangement $\prod$ moves backward for one step, thereby, it produces a new solution arrangement $\prod{}^1$

$$\prod{}^1 = \begin{cases} (\pi_1,\cdots,\ \ \pi_{i-1},\pi_j,\pi_i,\cdots,\pi_{j-1},\pi_{j+1},\cdots,\pi_n) & \text{if i < j} \\ (\pi_1,\cdots,\ \ \pi_{j-1},\pi_j,\pi_i,\cdots,\pi_i,\pi_{i+1},\cdots,\pi_n) & \text{if i > j} \end{cases} \qquad (9)$$

The field of inquiry of the hisert mover operation SERT-MOVE (π_{j}, i) is:

$$\mathrm{N}_2 = \{\prod{}^1 : INSERT\_MOVE(\pi_j, i),\ forj = 1,\cdots,n\ and\ i = 1,2,\cdots,j+1,\cdots,n\} \qquad (10)$$

N$_2$ can be divided into n sub-division (sub-field). All the sub-fields are related to the solution element π_{j}. The definition of N_2^J is:

$$\mathrm{N^J}_2 = \{\prod{}^1 : INSERT\_MOVE(\pi_j, i),\ forj = 1,\cdots,j+1,\cdots,n\} \qquad (11)$$

“The sum of all element values in corresponding line of the solution element on E” is regarded to be the attractive value of sub-field. Rank these attractive value, so, when making local search, firstly explore the sub-field with the biggest attractive value.

Process of discrete multi-objective optimization algorithm based on PSO

The key problem of PSO is the selection of P best and G best. Among the problem of single object optimization, we can make the choices through objective functional value. However, it is hard to make sure which particle is better through objective functional value in multi-Objective Optimization Problem, thus, no way can be found to select the individual optimal value and local optimal value. Maintaining non-dominated solution set is to provide diversified P best for the iterative search of particle swarm. At the same time, maintain and further strenghten the diversity of the soluton in non-dominated solutions through selecting an appropriate P best for each particle. The thesis uses non-dominated solution set to store the non-dominated solutions produced by iterative search. And conduct updating maintenance for non-dominated solution set timely. During the process of optimization routine, the P best of the particle conducts iteration update by dominating set. G best is randomly selected from non-dominated solution set, so as to ensure that non-dominated solution set is gradually close to the optimal solution of Pareto.

The process of discrete multi-object algorithm based on particle swarm optimization is:

(1) Initialize particle swarm P, and set the max iterations M, particle swarm scale N, particle dimensionality j of DMOPSO algorithm. Randomly produce the position xij and speed vij of each particle. The initial value of iterations counter is set to be 0.

(2) Set iterations counter to be 1, and calculate each particle's fitness value $f_k(x_{ij})$ of multiple objective function $f_k(x)$

$$fitness(k,i) = f_k(x_{ij}) \quad (12)$$

Calculate the fitness value of the particle in the light of formula (13). On the basis of domination conception, rank the fitness value from top to bottom. Keep the first N_1 particle of the particle swarm in dominant sub-set NP, and the rest N_2 particle in non-dominant sub-set P. The positions of these particle is Non-dominated solution.

$$fitness(i) = \frac{1}{Z}\sum_{k=1}^{Z} fitness(k,i) \quad (13)$$

(3) According to formula (7), update the speed and position of N_1 particle in dominated sub-set NP . Determine the best initial position P best[k,i] of each particle as their own initial position, and the global optimum G best is randomly selected from non-dominated set N_2. In this way, it can increase the diversity of solutions and avoid producing the situation of local optimum.

(4) Dominate thought in terms of the fitness value, and update the P best[k,i] of the particles in non-dominating subgroup P.

$$p\mathrm{best}(i) = \frac{1}{Z}\sum_{k=1}^{Z} pbest(k,i) \quad (14)$$

(5) Maintain the non-dominating subgroup P. For the newly produced non-dominating solution $x_{ij}(t+1)$, we will adopt the following strategies:

When the newly produced solution is dominated by the members of non-dominating subgroup P, prevent the new solution from joining P; when the newly produced solution dominates part members of P, remove the members which is under the domination in P, and add the newly produced solution into this non-dominating subgroup P; When the newly produced solution and the members of non-dominating subgroup P are not dominated by each other, add the newly produced solution into P directly. When the size of non-dominating subgroup P reaches or excees the maximum size specified in advance, sort by the fitness value, and delete the non dominated solutions which is dominated by all the other members. The fitness value is calculated by (13).

(6) Check if it reaches the maximum number of iterations, if so, stop searching; if not, continue to search by transfering to (2).

Test based on discrete multi-objective optimization algorithm of PSO

In order to verify the performance of the algorithm in this thesis, we adopt the TSP problem provided by TSPLIB which is International general as the control standard of known optimal solution for test data. The experimental environment is Celeron 1.7G CPU, 256M RAM, Win2000 OS and VC6.0. The experimental contrast is divided into two parts. The first part is to compare with similar particle swarm algorithm; and the second part is to compare with other evolutionary algorithms. Through constructing the data, it shows some experimental results of the algorithm on solving the large-sized TSP problem provided by TSPLIB in this thesis.

Through choosing to keep the distance of the nearest city every time with greedy thought, it will form better sub fragment, so as to increase the local search speed rapidly in the initial period of search. However, a few good continuous sub segments take advantage in the group soon, which leads to the focus of particles. The search behavior at this time is not conducive to the global detection of particle swarm optimization on optimal solution. However, in this thesis, it keeps the better sub fragment while using the failure mechanism of poor fragment, that is to say, continuously destroys the sub fragments which are easy to produce inferior solutions and produces new sub fragment, which makes the particles show a trend of diversification. Therefore, it can make the algorithm come to the optimal solution with a few number of iterations, meanwhile, provide a very good solution to the defect that the greedy algorithm is usually easy to be attracted by the local extreme value and hard to escape.

Table 1. Contrast experiment for solving TSP problem in PSO algorithm

Solution space size	（14-1）！ /2=3,113,510,400		
Particle number	100		
Average evolution algebra	20,000	1000	20
Average search space size	20,000*100=2,000,000	1,000*100 =100,000	20*100=2,000
Ratio of Search space and solution space	2,000,000/3,113,510,400=0.064%	100,000/3,113,510,400=3.211E-5	2,000/3,113,510,400=6.42E-7
Best solution	3->1->0->9->8->10->7->12->6->11->5->4->3->2		
Value of best solution	30.8785 (Equal to the best known result in the world)		

Table 2. shows the contrast results of the algorithm in this thesis, PSO (Standard PSO algorithm) provided by literature [49], RPAC (random perturbation ant colony and SCPSO (Particle swarm optimization based on subgroup hybridization mechanism). The experiment data adopts the manage meng of three classic TSP problems provided by TSPLIB, which are gr24,ayes29 and gr48.The group size choose to use 100 (500 in other algorithm groups), average value is obtained by 200 iteration per time and keeps 30 times repeating totally.In the experiment process of this thesis, both gr24 and bayes29 problem obtain the known optimal solution during the 30 times repeating; moreover, gr48 reaches up to 93.3% for the times of obtaining optimum solution during the results of 30 times repeating, but no optimum solution is obtained for other comparison algorithms. From Table 2., we can see that the convergence rate of the algorithm in this thesis and the quality of the solution are obviously better than the other several contrasted algorithms, even with some smaller group sizes.

Table 2. Contrast experiment 2 for solving TSP problem in different PSO algorithms

TSP Problem	TSP Optimum Solution	PSO Average Solution	RPAC Optimization Solution	SCPSO Average Solution	Average Solution of Algorithm in this paper	Best Solution of Algorithm in this paper
gr24	1272	1305	1278	1274	1272	1272
Bayes29	2020	2102	2042	2028	2020	2020
gr48	5046	5528	5265	5199	5048	5046

Take gr24 problem as an example, Chart 2 shows the convergence curve which compares the difference between adopting genetic algorithm to crossover operator and multi group hybrid in original PSO algorithm. Compared with genetic algorithm to crossover operator of POS algorithm and original PSO algorithm, the quality and convergence speed of adopting multi group hybrid of PSO algorithm are both better than the first two algorithms, through using some better solutions fragments from other subgroups. However, in multi subgroup hybrid PSO algorithm the subgroups probably conduct their own evolution independently at the very beginning. So, only when the fitness change of all the subgroups stop, ionic exchange among the subgroups begin to conduct, which doesn't make the convergence speed of the algorithm effectively improved. At the same time, for the ionic exchange among the subgroups begin to conduct only when the fitness change of all the subgroups tends to stop, the diversity of each subgroup it is not ideal overall, although it surely has some improvement. The algorithm in this thesis adopts information exchange strategy and dynamic division search strategy; therefore, it turns out to be better than the other three comparison algorithms no matter in the convergence rate of the algorithm or in the guarantee of the particle diversity.

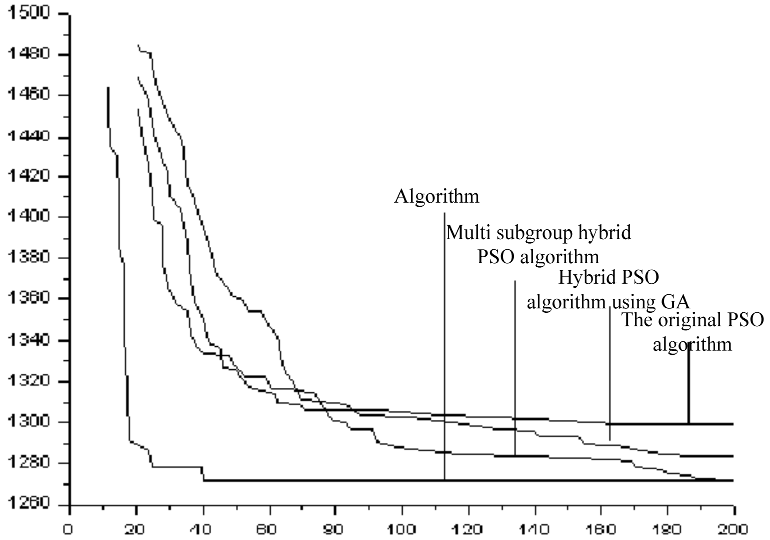


Chart 2. Convergence curve chart of comparison between the algorithm in this thesis and other PSO algorithms

Summary

In recent years, PSO has acquired good results on continuous optimization problem, however, it is still a brand new attempt on solving discrete optimization problem. In this chapter, the improved particle swarm optimization proposed in the thesis is successfully applied to TSP, which is a classic discrete optimization problem. According to PSO, through designing an information exchange strategy among particles with greedy thoughts, the new algorithm makes the individual particle acquire more useful information of other particles, so as to present different motion behaviors during the process of evolutionary. Improving PSO through combining with the dynamic division search strategy will effectively balance the search efficiency and accuracy. Through a large number of experiments, it shows that the improved PSO in this thesis can be effectively applied to TSP problem. The solving accuracy and efficiency are still relatively satisfactory.

References

[1] He Yijun. Swarm Intelligent Optimization and its Applied Research in Chemistry and Chemical Engineering. Hangzhou: Zhejiang University, 2008.

[2] Cui Guangzhao, Li Xiaoguang, etc.. Optimization of DNA Coding Sequence Based on Modified Particle Swarm Genetic Algorithm. Journal of Computer Science. Vol , 2, 2010, p311-316.

[3] Yu Xuejing, Ma Xiaofei, Xia Bin. Dynamic Particle Swarm Optimization. Computer Engineering. Vol 4, 2010, p 193-194.

[4] Pan Feng, Tu Xuyan, Chen Jie, etc.. Hybird Particle Swarm Optimization-HPSO. Computer Engineering. Vol , 31(1), 2005, p 169-171.

[5] Zeng Jianchao, Jie Jing, Cui Zhihua. Particle Swarm Optimization. Beijing: Science Press, 2004.

[6] Yang Hongru, Gao Hongyuan, Pang Zhengwei, etc.. Multi-user Detector Based on Discrete Particle Swarm Optimization. Harbin Institute of Technology Press. Vol 37(9), 2005, p 1303-1306.

[7] Yang Wei, Li Qiqiang. Review of Particle Swarm Optimization. Chinese Engineering Science, Vol 6(5), 2004, p 87-94.