NEW GLOBAL OPTIMIZATION ALGORITHM (SIA) FOR IN-CORE FUEL MANAGEMENT

Zhihong Liu Gong Shi
Department of Engineering Physics,
Tsinghua university,
Beijing 100084 P. R. China
lzh@tsinghua.org.cn ; shig@mail.tsinghua.edu.cn

Yongming Hu
Institute of Nuclear Energy Technology,
Tsinghua university
Beijing 100084 P. R. China
yongming@dns.inet.tsinghua.edu.cn

ABSTRACT

In the field of global optimization, there are many algorithms such as Simulated Annealing Algorithm (SA), Genetic Algorithm (GA), Artificial Neural Network (ANN), and so on. They are all based on the imitation of nature. This paper presents a new global optimization algorithm--statistic inductive algorithm (SIA), which is based on probability theory. The calculated results on some standard testing functions and a 'travelling salesman problem' (TSP) show that SIA is of more effective quality on global optimization than SA and GA. In addition to these results, according to this fundamental optimization idea, this paper analyzed why SA and GA can have effect on global search, and what inherent defects SA and GA have. Reloading optimization code used SIA has been completed. Calculations were performed on various reactor cores. Compared with SA, GA, and Ant System, SIA need less computation time, while the result is much better than them. Conclusion is drawn that SIA is more suitable for in-core fuel management.

1. INTRODUCTION

To solve a global optimization problem, probabilistic algorithms are often used, such as Simulated Annealing Algorithm (SA), Genetic Algorithm (GA), Artificial Neural Network (ANN), etc[1- 6]. However, the strict mathematical principle of SA and GA has not been described clearly in previous references. It is only explained with the simulation of natural phenomena. This report exposes the mathematical hints of SA

and GA based on a more fundamental optimization idea and draws a conclusion that they all have inherent defects in finding a global optimization solution. According to this optimization idea, a new global optimization algorithm -- statistic inductive algorithm (SIA) is presented. The calculated results on some well-known benchmark testing functions and on a difficult NP problem -- 'travelling salesman problem' (TSP), show that SIA is of more effective quality in solving global optimization problems than classical SA and GA methods.

Reloading optimization code used SIA has been completed. Calculations were performed on the core of *Dayabay Nuclear Plant* and two nuclear plants in France. Compared with SA, GA, and Ant System, SIA need less computation time, while the result is much more better than them. It can be found that SIA is more suitable for in-core fuel management.

2. BASIC PHILOSOPHY OF SIA METHOD

For a given optimization problem, the goal is to have the greatest probability of obtaining a solution with high quality in a specific calculating time. If we take an analysis in theory, we can draw a conclusion that this probability is associated with two aspects; one is the searching process of our algorithm, the other is the given probability conditions of the optimization problem itself. If we don't know any probability conditions of the problem itself, then none of the algorithms can solve the problem efficiently, including the SA, GA, ANN and all the other algorithms. They will all have the equal efficiency as a blind search.

Therefore, when we prepare to solve a real optimization problem, we must have insight about the problem itself. In other words, the problem must provide some preconditions on probability. So when we are at a specific stage of optimization process, the information which can lead us searching forward can be divided into two parts.

I. The function values of the points (solutions) which have been calculated previously.

II. The properties of the function itself which are known or assumed before solving. (e.g. continuity, relativity, and so on)

It should be pointed out that the information **II** is very important to our optimization, as it constructs the relationship between different points, tells us how to extract useful information from the information **I** and how to utilize this information **I** most efficiently. If the information **II** is lost, then the information **I** will be of no use to the following searches, and the problem can not be solved efficiently, whether any of the

algorithms are used.

In fact, when the algorithms above (SA, GA, and so on) are used on a real optimization problem, information is always obtained from **I** and **Π** (though sometimes we used information **Π** unconsciously). However, each of these algorithms only partly use the information **I** and **Π**, so each has its inherent limitation. Take SA and GA as examples.

The idea of **SA** comes from the annealing process of crystal, but according to the theory above, the mathematical principle of SA can be explained as follows.

For SA, the information **Π** utilized is that under the given neighborhood rule, the function is approximately smooth in global and local meaning (see fig 4.1). Its process of obtaining the global optimum can be explained as follows:

Stage 1: On the first stage, the annealing temperature is high, according to the Metropolis accepting rule, the searching process is not sensitive to the small variance of function value. It only be affected by the largely varying trend of function value, so the whole function acts as the broken line 'a' to the search, see fig 1. After a large number of stochastic moving (under the accepting rule), it will go to a point not far from the peak of broken point 'a'. It is a good point in the global view.

Stage 2: Reduce the annealing temperature and start from the end point of the last stage. Because the annealing temperature is reduced, the search process will now be more sensitive to the variance of function value, so the function acts as the broken line 'b' to the search in this stage and the end point of this stage will be near the peak of broken point 'b'.

Stage 3: Reduce the annealing temperature, repeat stage 2, until it goes to a local optimum.

the first stage the second stage

fig 1 searching process of SA

From the search process mentioned above, we can see that if SA is to be efficient, the function property must approximate the information **Π**. In other words, on each annealing stage, the outline of the function (i.e. the broken lines in fig 1) must encompass only one dominant peak. However, to many combined optimization

problems, giving a neighborhood rule to make the function have such a property is difficult, so the effect of SA will not be satisfactory.

The idea of **GA** is from the evolution of species, but according to the theory above, the mathematical truth of GA can be explained as follows.

For GA, the information **Π** that has been used is that the parts of the variable code approximately independently affect the function value.

Assume there are two individuals in the last generation, which will be in mating. Their codes are:

$$\times\times\times \quad \boldsymbol{a} \quad \boldsymbol{b} \quad \times\times\times \qquad \text{and} \qquad \times\times\times \quad \boldsymbol{c} \quad \boldsymbol{d} \quad \times\times\times$$

After mating, create the new individuals:

$$\times\times\times \quad \boldsymbol{a} \quad \boldsymbol{d} \quad \times\times\times \qquad \text{and} \qquad \times\times\times \quad \boldsymbol{c} \quad \boldsymbol{b} \quad \times\times\times$$

Because code ***a, c*** and ***b, d*** affect the function value independently, the two new individuals should have approximately equal average function value as the last two individuals. Then Remove the bad individuals in the process of 'duplicating ', the average quality of the next generation will be improved. However, in this improving process, there are no operations related to 'local' or 'neighborhood', and the generation quality is improved by incoming new individuals, so it should not converge to a local optimum too quickly.

To many optimization problems, the primary code of variables haven't the property given by the information **Π** above, so the code form of variables must be changed, with the purpose of satisfying the information **Π**. However, not all the problems can find a code form like this easily (e.g. the loading pattern optimization problem in reactor fuel management), so the performance of GA will be not so good.

In view of solving real optimization problems, the properties of real problems (i.e. information **Π**) differ greatly, while SA or GA always use one fixed property as information **Π**, so both have their own inherent limitations. In order to search forward most efficiently in an optimization process, we must utilize the information **I** and **Π** completely and suitably. SIA (statistic inductive algorithm) realizes this goal by fully taking into account the information **I** in various ways with the change of information **Π** for a pre-defined optimization problem. For each character of a problem specified as the information **Π**, SIA can use it efficiently in searching the optimized solution, so its efficiency of searching the global optimization solution is much higher than SA and GA.

3. SCHEME OF SIA METHOD

SIA analyzes the information **Π** of a real problem, then decides how to extract useful information from the information **I**. According to these useful information, gives the searching direction of the next stage. The basic step of SIA is:

Step 1. Give *N0* feasible points (solutions) randomly, and calculate their function values.

Step 2. Sort these *N0* points by their function values and take out *N1* better points as the statistic object of the next step.

Step 3. According to the information **Π** already known or assumed, identify the items which need statistic result, then analyze these items of the *N1* better points, and obtain the statistic distribution of each item.

Step 4. According to these statistic results, construct *N1* new points whose statistic distribution on these items are the same as that of the last *N1* better points.

Step 5. Put these *N1* new points into the calculated points, sort again, and take out *N1* better points. Then return to step 3.

If no important statistic item has been lost in the step 3, then the *N1* new points constructed in step 4 should have equal average function value as that of the last *N1* better points, and the searching scope will converge in a speed of about $1 / 2^n$. Here 'n' is the number of the iteration above.

When we implement SIA on a real problem, the process of 'obtaining statistic distribution' and 'constructing new points' in step 3 and 4 need a little skill, but the calculated result will not be impacted, regardless of how it is realized. The implementing methods of SIA on some real optimization problems will be introduced in the next section.

4. VALIDITY TESTS OF SIA METHOD

In order to test the performance of SIA, it was implemented on both continuous optimization problems and a combined optimization problem. All the optimization problems are taken from the references [7] and [8], so that it can be compared with SA and GA.

4. 1 CONTINUOUS OPTIMIZATION PROBLEM

Table 1 shows the test result of SIA, SA, and GA on some standard test functions. The test functions are taken from the reference [8]. These four functions all have many local optimums. The functions can be expressed as $F = F(x_1, x_2, \ldots, x_i, \ldots, x_n)$.

These four problems hadn't given any function property before being solved, so a function property as the information **Π** must be assumed. The information **Π** used here for SIA assumes that the function value is not acutely changed by the varying of each variable. In other words, if the function value is good when the variable x_i is equal to λ, then the function value will also be good when the variable x_i is changed to $\lambda + \varepsilon$. Here the range of ε is related to the converging degree of this statistic item(i.e. the converging degree of the item 'x_i'). This is to say, the statistic items $\{G_i\}$ selected for the problem are

$$G_1 = x_1, \quad G_2 = x_2, \quad G_3 = x_3, \quad \ldots \quad, G_n = x_n \qquad (1)$$

According to the information **Π** given above, we can realize step 3 in the follows. To each variable x_i, it has *N1* various values in the *N1* old better solutions (statistic object). *N1* Gauss distributions are now given, whose centers are these *N1* values respectively. These *N1* Gauss distributions are then added and a 1-D distribution curve is obtained. In step 4, this curve is regarded as the probability distribution curve of x_i, and give *N1* new values of x_i. The half width б of the Gauss distributions above is related to the converging degree(here is asymmetry degree of points distribution) of item x_i, given by calculation.

Table 1 shows that SIA can obtain the global optimum in smaller number of function evaluations, which shows that SIA is more efficient on global optimization than SA and GA.

Tab 1 The number of function evaluations of various algorithms

Function	Dimension	SA	GA	SIA
F6	20	>9900	9900	6600
F7	10	>8699	8699	2500
F8	10	—	59520	9000
F12	5	9018	—	2400

If a real continuous optimization problem were to give some additional function properties, then statistic items should be added in step 3. The converging speed of SIA would then be quicker, but the implementation of SIA would be more troublesome.

4. 2 COMBINED OPTIMIZATION PROBLEM

An NP-hard problem --'travelling salesman problem' (TSP) was selected as the testing problem, as follows. *N* cities are given, and the distances between each two cities are known. An optimal path which passes through each city and has the shortest total distance is to be obtained.

For TSP problem, the information **II** we have known before is: the function value is only be affected by which two cities are linked in the solution. So in step 3, the statistic items $\{G_i\}$ selected is:

$$G_i = \text{the number of city linked with city } i \qquad (2)$$

N statistic items are selected in all. And in step 4, we should give *N1* new solutions whose statistic distribution of each item G_i is similar as that of the old *N1* better solutions.

This algorithm has been used on a TSP problem with 144 cities, and compared with other algorithms. This TSP problem is taken from reference [7] and uses the 144 big cities in China. Table 2 shows the calculated results of SA and other algorithms modified from SA. These results are also given by reference [7]. Each algorithm performs 20 calculations with different initial solutions. Here the 'N_c' in tab 2 is the number of function evaluations.

Tab 2 Calculated results on TSP144

Algorithm	N_c	The best solution (L^+)	The worst solution (L^-)	Average solution ($\bar{L}$)
SA	600,000	30566	32652	31143
Modified 1	>600,000	30708	31397	31010
Modified 2	>600,000	30573	30947	30831
Modified 3	>600,000	30421	31014	30660
SIA	100,000	30355	30740	30545
SIA	300,000	30349	30487	30375

'The best path known until now' given by reference[7] is 30380. SIA soon obtained a better path with a length of 30355. This proved that SIA is more skillful on global search than SA, GA, and all the other algorithms which have previously been used.

10 calculations have been performed with SIA, with the number of function evaluations controlled to be 100,000 and 300,000 respectively. The result is shown in table 2 and clearly shows that SIA needs a smaller number of function evaluations and obtains a much better solution than other algorithms.

From the result on TSP144, three aspects of SIA can be compared with the other

algorithms:

(i). Efficiency: If all the algorithms attain the same solution quality (e.g. 30545), the number of function evaluations using the other algorithms are about 60 times greater than the number of SIA.

(ii). Globality: The good solutions which can be easily obtained by SIA (such as 30355, 30349), can not be obtained by any of the other algorithms (SA, GA etc).

(iii). Stability: The distance between 'the best solution' and 'the worst solution' of SIA is much smaller than the other algorithms.

SIA can also be easily used to solve other NP problems, such as 'zero-one knapsack problem' (ZKP), 'max cut problem' (MCP), etc. In fact, SIA is suitable for almost all the global optimization problems which need to obtain more global optimum with a lesser number of function evaluations paid out.

5. APPLICATION OF SIA TO RELOADING OPTIMIZATION

Mono-cycle optimization for reactor fuel management optimizes the fuel loading pattern and the burnable poison arrangement for a specified fuel cycle subject to the safety and engineering limits, so that the depletion process of the fission material in reactor core is controlled and the required objectives are fulfilled. The objectives to be optimized can be maximization of cycle length, discharged burn-up, etc., and minimization of the power peaking factor, as well as other objectives. The optimization problem can be expressed as below:

$$\text{Maximize} \qquad g\,(\mathrm{LP}, \mathrm{BP_k}) \tag{3}$$

$$\text{Subject to} \qquad F_i < F_{lim}, \quad i = 1, 2, \ldots, N_F$$

where g is the objective function which could be any objectives mentioned above. F_i is the relative power of the assembly i. F_{lim} is the limit of the pre-specified relative assembly power peaking factor required by thermal-hydraulic criteria. N_F is the total number of fuel assemblies in reactor core. LP stands for loading pattern in reactor core. $\mathrm{BP_k}$ is the loaded quantity of burnable poison in k-th fuel assembly where the burnable poison can be arranged.

Some other necessary limits in engineering must also be fulfilled, such as the requirement of negative moderator temperature reactivity, maximum allowable burn-up and the tolerable burnable poison quantity, etc.

It has been described before that each type of global optimization method must utilize the inherent characteristics of optimization problems for possibly achieving the optimization effect. SA and GA method always use a type of fixed characteristics while ignoring whether or not this type of characteristics exists in the calculated problem. The characteristics used in SA algorithm suppose that the change of objective function is not drastic, and GA assumes that the interactive influences of various codes of variables on objective function values are relatively independent. However, SIA algorithm can flexibly select the items to be statistically calculated, based on the characteristics of the optimization problem itself, and can combine the statistical characteristics together in a statistical way, otherwise their relations could hardly be discovered. Thus SIA method has a stronger capability for solving optimization problems and has a wider range of suitability than SA and GA algorithms.

In a mono-cycle optimization problem for fuel assembly arrangement, the limitation of power peaking factor results in many unacceptable points. The objective function changes quite drastically, so that it is not compatible to the assumed characteristics in simulated annealing algorithm. Moreover, the relative power at a location in reactor core is influenced not only by the status of the assembly there, but also strongly by the status of the neighboring fuel assemblies. This means that the coupling of variables among various sections is quite strong, so its conformance to the characteristics assumed to exist in GA method is not satisfactory. However, SIA method has displayed its advantages in dealing with the optimization problem for reactor fuel management.

For in-core fuel management optimization problem, the goal is to obtain a solution whose objective is large enough and constraints can be satisfied. We know the objective and constraints are all associated with assembly relative power F_i. While the F_i is mainly associated with the k_∞ of assembly on position i and the assemblies near position i. This is just the information $\mathbf{\Pi}$ of this problem. According to this information $\mathbf{\Pi}$, statistic items $\{ G_i \}$ of SIA can be selected as

$$G_i = G_{k,j} = k_{\infty,k,j} + w_1 . k_{\infty,k-1,j} + w_2 . k_{\infty,k+1,j} + w_3 . k_{\infty,k,j-1} + w_4 . k_{\infty,k,j+1}, \qquad (4)$$

$$i = 1, 2 , \ldots , N_F$$

w_1, w_2, w_3, w_4 are weight of the 4 assemblies near assembly i. There values are given by calculation, whose rule is to make the converging degree of statistic item G_i is largest.

After statistic items are ensured, in step 4, we should construct *N1* new LPs, whose statistic distribution of each item G_i is similar as that of the old *N1* better solutions.

A flow chart of reloading optimization with SIA is shown as Fig 2. This code has been completed, and some results are as below.

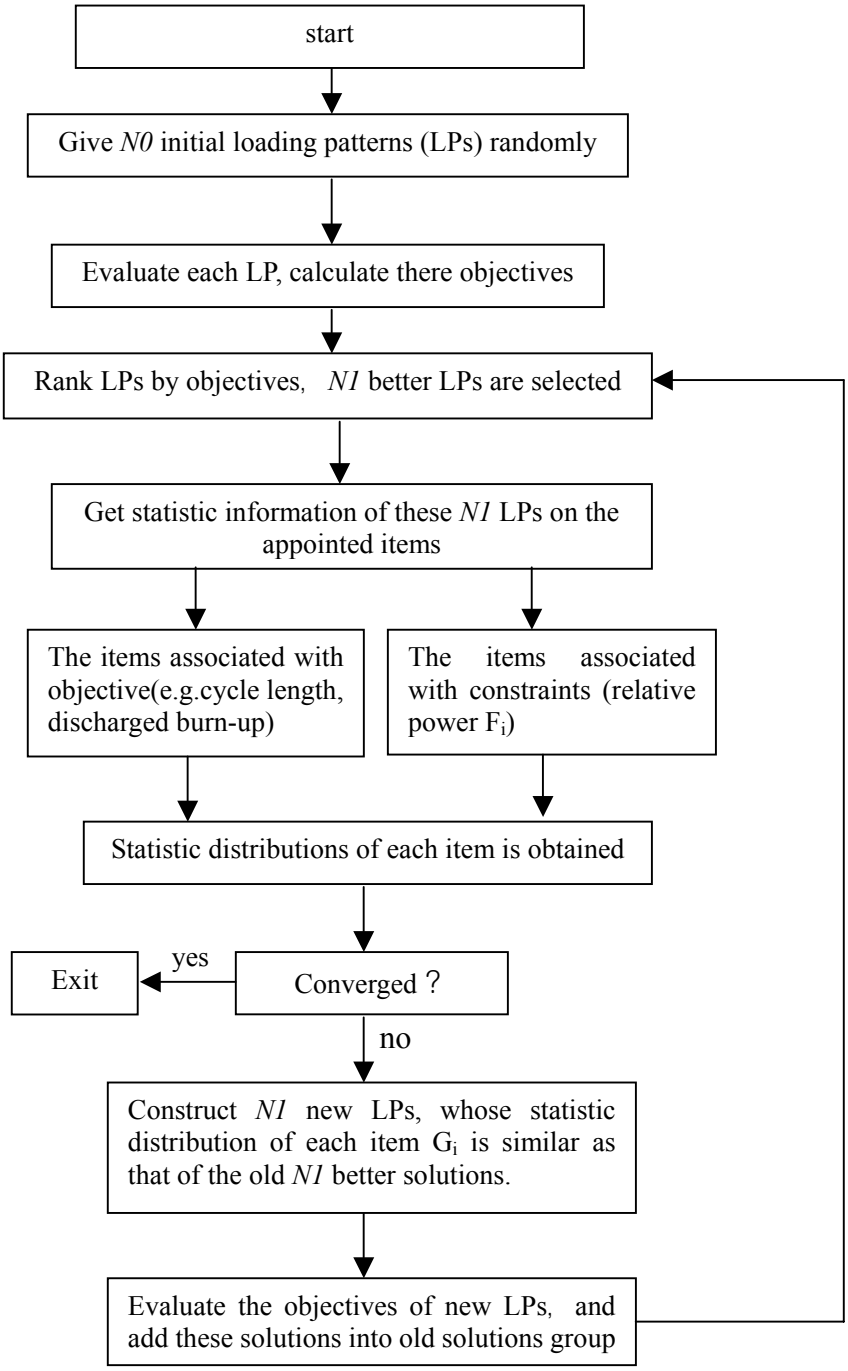

Fig 2 Flow chart of reloading optimization with SIA algorithm

The first cycle of *Dayabay Nuclear Plant* is taken as an instance. This core have burnable poisons. Tab 3 shows the results of reference LP and optimized LPs for varied objectives. The reference LP is the real loading pattern of *Dayabay Nuclear Plant*. The relative power limitation is set as 1.25. All the optimized solutions satisfy

this limitation, and are one-eighth symmetry cores. BOC boron concentrations are all under 1200ppm. Positions of control rods have been considered. The number of function evaluations is about 2000 for each objective, and 1 to 2 hours is needed on a computer (PIII 667).

The results of SIA have been compared with that of SA and GA. The advantages of code with GA are 2.7%, 3.3% and 3.2%, respectively. The advantages of code with SA are 3.0%, 3.5% and 6.4%, respectively. The number of function evaluations of these codes are about 12000. It can be found that the computing time of SIA is only 1/6 of SA or GA, while the solutions are much better than them.

Tab 3 optimization results for the first cycle of *Dayabay Nuclear Plant*

Objective	Cycle length (d)	F_{xy}	Discharged Burn-up(GWd.t $^{-1}$)	Advantage
Reference LP	335	1.250	13163	---
Cycle length	355	1.248	12138	6.0 %
Discharged Burn-up (cycle length fixed)	335	1.248	13729	4.3 %
F_{xy} (cycle length fixed)	335	1.137	13203	9.0 %

The second instance is the reloading pattern optimization of *CRUSAZ nuclear plant* in France. This core have no burnable poisons. Tab 4 shows the results of reference core and optimized cores. The reference loading pattern is obtained by experiences and simple optimization. The objective is cycle length. Varied F_{xy} limitations are set and varied optimums are obtained. 10000 of function evaluations are needed for each computation. About 2 hours are consumed. It should be pointed out that, this code utilizes no experience, while the new assembly positions of its solutions are all same as the positions given by experiences.

Tab 4 optimization results for *CRUSAZ Nuclear Plant*

Core parameters / Loading patterns	Cycle length (d)	F_{xy}	Discharged Burn-up(GWd.t $^{-1}$)
Reference LP	270	1.257	44523
Optimized LP 1	275	1.247	44197
Optimized LP 2	274	1.242	44408
Optimized LP 3	271	1.218	44555
Optimized LP 4	270	1.201	44542

This code is also compared with a code used Ant Colony Optimization (ACO)[5]. For

a very sample problem(use simplified assemblies), SIA and ACO have same results, but the computation time of SIA is more than ACO; For a more complex problem (use real assemblies of *CRUSAZ* above, but the positions of fresh assemblies and some oldest assemblies are fixed), SIA obtains the optimum more quickly than ACO; For a real complex problem (no assembly position is fixed), SIA gets the results as Tab 4, while ACO can not obtain an acceptable solution in 20000 of function evaluations. It can be found that SIA is more suitable for complex problems.

SUMMARY

Because SIA has overcome the inherent defects which SA and GA have according to a deeper insight of optimization, it has a much better performance than SA and GA in finding globally optimized solutions for engineering problems. The main characters by which SIA method is different from other global optimization algorithms in include:

- SIA can select varied statistic items flexibly for varied real problems, which make the information Π of problem can be used completely and suitably.

- One-dimensioned probability distributions are used to describe the known information, which is more suitable for complex conditions of a specified problem.

- SIA is derived from more precise probability theory, which make it more efficient than other algorithms.

SIA can be easily modified to be executed in parallel form. This will save a lot of computation time. So SIA is very suitable for solving the complex optimization problems in many engineering fields, such as fuel loading pattern optimization in the field of reactor fuel management. Besides global optimization, the basic idea of this algorithm can also be used on common numeral calculations.

REFERENCES

1. G.T.Parks, “Multiobjective pressurized water reactor reload core design by non-dominated genetic algorithm search,” *Nuclear Science and Engineering*, **124**, pp. 178-187 (1996)
2. Chapot et al, “A New Approach to Use of Genetic Algorithms to the Pressurized Water Reactor’s Management Optimization Problem,” *Annals of Nuclear Energy*, **26**, pp. 641-655 (1999)
3. Jung Hoon Lee, “Incorporation of neural networks into simulated annealing algorithm for fuel assembly loading pattern optimization in a PWR,” *Transactions of the American Nuclear Society*, **80**, pp. 228-300 (1999)
4. Akio YAMMOTO, “A Quantitative Comparison of Loading Pattern Optimization Methods

for In-Core Fuel Management of PWR," *Journal of Nuclear Science and Technology*, **34**, pp. 339-347 (1997)

5. T. Stutzle, H. H. Hoos, "Max-Min Ant System," *Future Generation Computer System*, **16**, pp. 889-914 (2000)
6. XING Wenxun , XIE Jinxing, *Modern optimization algorithm*, Tsinghua university press, Beijing, China (1999)
7. KANG Lishan , XIE Yun , YOU Shiyong, *Non-numerical parallel algorithm -- simulated annealing algorithm*, Scientific Press , Beijing, China (1998)
8. LIU Yong , KANG Lishan , CHEN Yuping, *Non-numerical parallel algorithm -- genetic algorithm*, Scientific Press , Beijing, China (1998)