

An Improved Nested Partitions Algorithm Based on Simulated Annealing in Complex Decision Problem Optimization

CHANG-RUI YU¹, YAN LUO²

¹School of Information Management and Engineering
Shanghai University of Finance and Economics
No.777, Guoding Rd. Shanghai.
P.R.China

yucr@sjtu.edu.cn

²School of Management
Shanghai Jiao Tong University
No.535, Fahuazhen Rd. Shanghai.
P.R.China
yanluo@sjtu.edu.cn

Abstract: - This paper introduces the main ideas of the nested partitions (NP) method, analyses its efficiency theoretically and proposes the way to improve the optimization efficiency of the algorithm. Then the paper introduces the simulated annealing (SA) algorithm and incorporates the ideas of SA into two of the arithmetic operators of NP algorithm to form the combined NP/SA algorithm. Moreover, the paper presents the explicit optimization procedure of the combined algorithm NP/SA and explains the feasibility and superiority of it. The NP/SA algorithm adopts the global optimization ability of NP algorithm and the local search ability of SA algorithm so that it improves the optimization efficiency and the convergence rate. This paper also illustrates the NP/SA algorithm through an optimization example.

Key-Words: - Nested partitions algorithm, Simulated annealing, Complex decision problem.

1 Introduction

The solution of many complex decision problems involves combinatorial optimization, i.e., obtaining the optimal solution among a finite set of alternatives [1] [2]. Such optimization problems are notoriously difficult to solve. One of the primary reasons is that in most applications the number of alternatives is extremely large and only a fraction of them can be considered within a reasonable amount of time. As a result, heuristic algorithms, such as evolutionary algorithms, tabu search, and neural networks, are often applied in combinatorial optimization. All of these algorithms are sequential in the sense that they move iteratively between single solutions or sets of solutions. However, in some applications to the complex decision it may be desirable to maintain a more global perspective, that is, to consider the entire solution space in each iteration. In this paper we propose a new optimization algorithm to address this difficult class of problems. The new method combines the nested partitions (NP) method and the simulated annealing (SA) method. It converges to a global optimum for combinatorial optimization problems in finite time, and effectively reduces the number of times

backtracking occurs in the nested partitioning. Numerical results demonstrate the effectiveness of our proposed method.

The remainder of the paper is organized as follows. In Section 2 we review the general procedure of the NP method and analyse its optimization efficiency in detail. In Section 3 we present a combined NP/SA algorithm, i.e. an improved NP algorithm enhanced with simulated annealing. In Section 4 we give a numerical example to illustrate the hybrid method, and Section 5 contains some concluding remarks and future research directions.

2 The Nested Partitions Method

The NP method, an optimization algorithm proposed by L. Shi and S. Ólafsson [1], may be described as an adaptive sampling method that uses partitioning to concentrate the sampling effort in those subsets of the feasible region that are considered the most promising. It combines global search through global sampling of the feasible region, and local search that is used to guide where the search should be concentrated. This method has

been found to be promising for difficult combinatorial optimization problems such as: the traveling salesman problem [4], buffer allocation problem [5], product design problem [6] [7], and production scheduling problems [8].

Suppose the finite feasible region of a complex decision problem is Θ . Our objective is to optimize the objective performance function $f: \Theta \rightarrow \mathbf{R}$, that is, to solve:

$$\max_{\theta \in \Theta} f(\theta)$$

where $|\Theta| < \infty$. Also, to simplify the analysis, we assume that there exists a unique solution $\theta_{opt} \in \Theta$ to the above problem, which satisfies $f(\theta_{opt}) > f(\theta)$ for all $\theta \in \Theta \setminus \{\theta_{opt}\}$.

Definition 1. A region partitioned using a fixed scheme is called a valid region. In a discrete system a partitioned region with a singleton is called a singleton region. The collection of all valid regions is denoted by Σ . Singleton regions are of special interest in the process of optimization, and $\Sigma_0 \subset \Sigma$ denotes the collection of all such valid regions.

The optimization process of the NP method is a sequence of set partitions using a fixed partitioning scheme, with each partition nested within the last. The partitioning is continued until eventually all the points in the feasible region correspond to a singleton region.

Definition 2. The singleton regions in Σ_0 are called regions of maximum depth. More generally, we define the depth, $dep: \Sigma \rightarrow N_0$, of any valid region iteratively with Θ having depth zero, subregions of Θ having depth one, and so forth. Since they cannot be partitioned further, we call the singleton regions in Σ_0 regions of maximum depth.

Definition 3. If a valid region $\sigma \in \Sigma$ is formed by partitioning a valid region $\eta \in \Sigma$, then σ is called a subregion of region η , and region η is called a superregion of region σ . We define the superregion function $s: \Sigma \rightarrow \Sigma$ as follows. Let $\sigma \in \Sigma \setminus \Theta$. Define $s(\sigma) = \eta \in \Sigma$, if and only if $\sigma \subset \eta$ and if $\sigma \subseteq \xi \subseteq \eta$ then $\xi = \eta$ or $\xi = \sigma$. For completeness we define $s(\Theta) = \Theta$.

A set performance function $I: \Sigma \rightarrow \mathbf{R}$ is defined and used to select the most promising region and is therefore called the promising index of the region.

In the k -th iteration of the NP method there is always a region $\sigma(k) \subseteq \Theta$ that is considered the most promising, and as nothing is assumed to be known about location of good solutions before the search is started, $\sigma(0) = \Theta$. The most promising

region is then partitioned into $M_{\sigma(k)}$ subregions, and what remains of the feasible region $\sigma(k)$ is aggregated into one region called the surrounding region. Therefore, in the k -th iteration $M_{\sigma(k)} + 1$ disjoint subsets that cover the feasible region are considered. Each of these regions is sampled using some random sampling scheme, and the samples used to estimate the promising index for each region. This index is a set performance function that determines which region becomes the most promising region in the next iteration. If one of the subregions is found to be best, this region becomes the most promising region. If the surrounding region is found to be best, the method backtracks to a larger region. The new most promising region is partitioned and sampled in a similar fashion.

2.1 The NP algorithm

The NP method comprises four basic arithmetic operators during the four steps respectively: partitioning the solution space, obtaining the sampling points, selecting a promising index function, and backtracking.

Step 1. Partitioning. After the k -th iteration ($k > 0$), the most promising region $\sigma(k)$ is further partitioned into $M_{\sigma(k)}$ subregions $\sigma_1(k), \dots, \sigma_{M_{\sigma(k)}}(k)$. What remains of the feasible region $\sigma(k)$, i.e., $\Theta \setminus \sigma(k)$, is aggregated into the surrounding region $\sigma_{M_{\sigma(k)}+1}(k)$. Then, $M_{\sigma(k)} + 1$ partitioned regions are obtained. When the first partition starts, the whole feasible region Θ is considered the most promising region, i.e., $\sigma(0) = \Theta$.

Since the feasible region Θ is finite, the partitioned regions we obtain will eventually be singleton regions, i.e., $M_{\sigma(k)} = 1$. Then two regions are obtained: $\sigma(k)$ and $\Theta \setminus \sigma(k)$.

Step 2. Random sampling. The next step of the algorithm is to randomly select N_j samples $\theta_1^{(j)}, \theta_2^{(j)}, \dots, \theta_{N_j}^{(j)}$, $j = 1, 2, \dots, M_{\sigma(k)} + 1$, from each of the subregions $\sigma_j(k)$ obtained by the partitioning operator. Because of the openness of the NP method, various random sampling methods can be adopted with a requirement that the possibility of each point in each region being selected is more than zero [9].

Step 3. Calculation of promising index. Given a promising index function $I: \Sigma \rightarrow \mathbf{R}$, sample each region $\sigma_j(k)$, where $j = 1, 2, \dots, M_{\sigma(k)} + 1$, according to the fixed sampling strategy and estimate the

promising index value of each region. For example, assume that the promising index value is the maximal objective function value of each region,

$$I(\sigma_j(k)) = \max_{\theta \in \sigma_j(k)} f(\theta), \quad j = 1, 2, \dots, M_{\sigma(k)} + 1.$$

Estimate the promising index value of each region $\sigma_j(k)$,

$$\hat{I}(\sigma_j(k)) = \max_{i=1,2,\dots,N_j} f(\theta_i^{(j)}), \quad j = 1, 2, \dots, M_{\sigma(k)} + 1.$$

Notice that $\hat{I}(\sigma_j(k))$ is a random variable.

As long as the promising index corresponds to the performance function in singleton region, it can adopt any form. That is to say, when $\sigma_j(k)$ is the region of maximum depth, i.e., $\sigma_j(k) = \{\theta\}$, $I(\sigma_j(k))$ must equal to $f(\theta)$, i.e., $I(\sigma_j(k)) = f(\theta)$. Except for this restriction the NP method does not have restrictions on the selection of promising index function, which indicates the openness of the NP methods. Then, the promising index values of the $M_{\sigma(k)} + 1$ regions are compared, and the most promising region is determined:

$$\hat{j}_k = \arg \max \hat{I}(\sigma_j(k)), \quad j = 1, 2, \dots, M_{\sigma(k)} + 1.$$

If $\hat{j}_k \leq M_{\sigma(k)}$, i.e., one of the subregions of the current most promising region is found to have the maximum promising index, then this subregion is the most promising region in the next iteration. If $\hat{j}_k = M_{\sigma(k)} + 1$, then the most promising region in the next iteration is determined by the backtracking operator.

Step 4. Backtracking. If the entire region except $\sigma(k)$ is found to be the most promising region, the algorithm backtracks to a larger region that contains the current most promising region $\sigma(k)$. The backtracking rules can be determined by the requirements. An obvious backtracking method is to make the superregion of the current most promising region the backtracking objective. The selection of the present most promising region is denoted as

$$\sigma(k+1) = \begin{cases} \sigma_{\hat{j}_k}(k) & \text{if } \hat{j}_k \leq M_{\sigma(k)} \\ s(\sigma(k)) & \text{otherwise} \end{cases}$$

Certainly, the entire finite feasible region Θ can be considered the backtracking objective, i.e., $\sigma(k+1) = \Theta$. Starting from the new most promising region $\sigma(k+1)$, the algorithm continues with the above-mentioned steps of partitioning, sampling, promising indices, and backtracking. Then, a sequence of partitioned regions is obtained. Finally, the algorithm comes to an end when the points in all feasible regions correspond to the singleton regions. The point in the singleton that has been considered

the most promising regions for the most times can be considered the global optimal solution.

2.2 The Analysis on Optimization Efficiency of the NP method

2.2.1 The significance of the number of times backtracking is implemented to the optimization efficiency of the NP method

During the optimization process using the NP method, if the current most feasible region is proved to be unsatisfactory by sampling and calculation of promising index, backtracking is then necessary. This implicates that the last time partitioning, sampling, and promising indices are invalid. The algorithm should backtrack to the last iteration and continue with sampling and promising indices. Therefore, backtracking implies the decrease of calculation efficiency.

In the k -th iteration of the NP method if the surrounding regions of $\sigma(k)$ is considered the most promising, it then backtracks to the superregion $s(\sigma(k))$ of the current most promising region and makes $s(\sigma(k))$ the most promising region for the next partitioning. In the condition that the partitioning and sampling schemes are fixed, each backtracking results in two more times of partitioning and $2N(M_{\sigma(k)} + 1)$ more points in the feasible regions are sampled, where $M_{\sigma(k)}$ is the number of feasible regions for partitioning with a fixed partitioning scheme, and N is the number of sampled points in each feasible region. Calculating the promising index at these points requires $2N(M_{\sigma(k)} + 1)$ performance functions of the promising index. The backtracking rate of the NP method is tightly related to the optimization efficiency indexes such as the convergence rate. If the backtracking is reduced once, $2N(M_{\sigma(k)} + 1)$ performance functions of the promising index are reduced, which consequently shortens the optimization route, reduces optimization time, and speeds up the convergence. Thus, the number of times backtracking occurs is an important criterion for measuring the efficiency of this simulated optimization method.

2.1.2 The analysis on optimization probability of the NP method

L. Shi and S. Ólafsson improved that the NP method converges to a global optimal solution with probability one [3]. Let $\eta_l \in \Sigma$ be a feasible region obtained by nested partitions, θ_l^* be the optimum we get after introducing some other local optimization algorithms (such as SA, tabu search, etc.) into sampling of the NP method, and θ_l' be the optimum we get using the other simple random sampling methods. Although we cannot assure that θ_l^* is the global optimum of the feasible region, the probability of θ_l^* being the global optimum is greater than the probability of θ_l' being the global optimum in that these local optimization algorithms are capable of avoiding getting trapped in the local optima, i.e., $P\{\theta_l^* \text{ is the global optimum of } \eta_l\} > P\{\theta_l' \text{ is the global optimum of } \eta_l\}$.

Suppose the global optimal solution to the original problem $\theta^* \in \eta_l \in \Sigma$, i.e., η_l is the feasible region that contains the global optimal solution. Then, in the process of nested partitioning, η_l is unavoidable in the way to the global optimal solution. Compare the promising index of η_l is compared with those of the other regions $\eta_i (i = 1, \dots, M_{\sigma(k)} + 1, i \neq l)$. If η_l is selected to be the most promising region, the backtracking is reduced for at least once. Therefore, we can infer that, if the probability of η_l being selected to be the most promising region is increased, the efficiency of the algorithm will be improved. The probability of η_l being selected to be the most promising region is:

$$P\{f(\theta_l^*) > f(\theta_1^*), \dots, f(\theta_l^*) > f(\theta_{M_{\sigma(k)}+1}^*)\} \\ = \prod_{\substack{i=1 \\ i \neq l}}^{M_{\sigma(k)}+1} P\{f(\theta_l^*) > f(\theta_i^*)\}.$$

where

$P\{f(\theta_l^*) > f(\theta_i^*)\} = \omega\rho + \psi(1 - \rho) = \rho + \psi(1 - \rho)$, ω is the probability of $f(\theta_l^*) > f(\theta_i^*)$ under the condition that θ_l^* is the global optimal solution, ρ is the probability of θ_l^* being the global optimal solution, and ψ is θ_l^* is the probability of $f(\theta_l^*) > f(\theta_i^*)$ under the condition that θ_l^* is the local optimal solution. As the l -th feasible region contains the global optimum, $\omega = 1$. The above probability function is shown as Fig. 1.

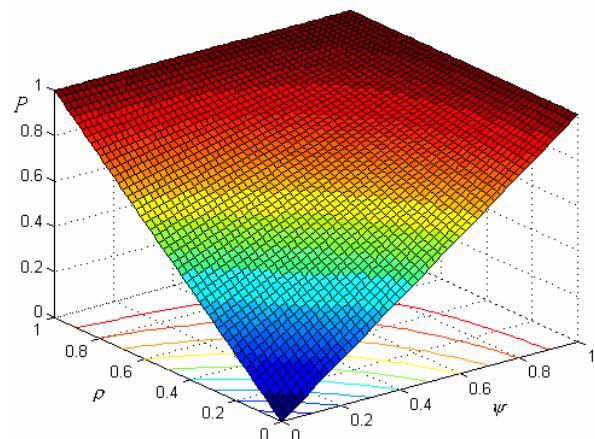


Fig. 1. The figure of the probability function

Therefore, the above probability equals the weighted average of 1 and ψ . And because $\psi \in (0,1)$, we have

$$\frac{\partial P}{\partial \rho} = 1 - \psi > 0.$$

If the probability ρ of θ_l^* being the global optimal solution is increased greatly, the above probability will correspondingly be increased. If the random sampling operator of the NP algorithm is changed and the probability of obtaining the global optimal solution in each region is increased, the convergence will be sped up and the efficiency of the algorithm will be improved greatly. The probability that the point we obtain using the local search of the SA method is the global optimal solution is much greater than the probability that the points we get using other simple randomized sampling methods are the global optima. Hence, the ideas of SA can be introduced into the NP method in order to increase the probability that η_l is selected properly, decrease the number of times that backtracking in the NP method is implemented, speed up the convergence, and eventually improve the optimization efficiency. In the next section we present a new algorithm combining NP and SA.

3 The Combined NP/SA Algorithm

3.1 The simulated annealing method

The simulated annealing algorithm (SA) is essentially a heuristic algorithm. The technique has been widely applied to a variety of problems including many complex decision problems. The term simulated annealing derives from the roughly analogous physical process of heating and then slowly cooling a substance to obtain a strong crystalline structure [10].

Often the solution space of a complex decision problem has many local minima. A simple local search algorithm proceeds by choosing random initial solution and generating a neighbor from that solution. The neighboring solution is accepted if it is a cost decreasing transition. Such a simple algorithm has the drawback of often converging to a local minimum. The SA method, though by itself it is a local search algorithm, avoids getting trapped in a local minimum by accepting cost increasing neighbors with some probability.

To solve the objective function $Z: \max_{s \in \Theta} f(s)$, over a feasible region Θ , SA is implemented in the following steps. Firstly, at temperature T , starting from an initial point $X^{(0)}$, randomly sample the feasible region. If $f(X^{(k)}) \geq f(X^{(0)})$, where $f(X^{(k)})$ is the function value of the sampled point $X^{(k)}$, $X^{(k)}$ is accepted and taken as the initial point $X^{(0)}$ to continue the optimization; otherwise, if $f(X^{(k)}) < f(X^{(0)})$, $X^{(k)}$ is accepted with a probability of $\exp((f(X^{(k)}) - f(X^{(0)}))/T)$. Then, beginning from the initial annealing temperature T_0 , the annealing temperature is lowered at a fixed temperature interval of ΔT . At each annealing temperature N points are randomly sampled. The above process is implemented repeatedly until the temperature reaches the final annealing one T_f [11] [12] and the algorithm converges to the global optimum.

3.2 The combined NP/SA algorithm

For a given feasible region the SA method focuses on searching for feasible points. It is capable of obtaining the global optima with a great probability and has a very strong local search ability. Applying the ideas of SA to the random sampling of the NP algorithm will greatly improve the ability of global optimization of the NP algorithm and the ability of local optimization of the SA method; hence the efficiency of the NP algorithm is improved greatly. Merging the SA method into the NP algorithm, we get the combined NP/SA algorithm.

Note that NP/SA is not simply merging the whole SA into the random sampling of the NP algorithm, but combining the basic optimization idea of SA with the complete optimization process of the NP algorithm properly in order to improve the optimization efficiency of the NP algorithm.

3.2.1 The implementation procedure of NP/SA

Similar to the preparatory work of SA implementation, firstly we need to set the initial annealing temperature T_0 , the final annealing temperature T_f , and the number N of random samples at each annealing temperature. NP/SA is an improvement of the NP algorithm. It has the same operations in partitioning, calculation of promising indices and backtracking. The random sampling of NP/SA is improved. Actually, NP/SA does not implement a complete annealing process in every sampled region to obtain an optimal solution over the region. Instead, NP/SA carry out the optimization according to the same annealing temperature over the feasible regions at the same depth. According to the maximum depth $dep(\sigma)$ ($\sigma \in \Sigma_0$) of singleton region in the feasible region, the annealing speed $\Delta T = (T_0 - T_f)/dep(\sigma)$ is set.

Respectively optimize the uncrossed $M_{\sigma(k)} + 1$ feasible regions obtained through the k -th partitioning at the annealing temperature $T_k = T_0 - dep(\sigma(k)) \cdot \Delta T$ according to the SA method. That is to say, starting from a certain initial point $X^{(0)}$, randomly sample the feasible regions. If $f(X^{(k)}) \geq f(X^{(0)})$, where $f(X^{(k)})$ is the function value of the sampled point $X^{(k)}$, $X^{(k)}$ is accepted and taken as the initial point $X^{(0)}$ to continue the optimization; otherwise, if $f(X^{(k)}) < f(X^{(0)})$, $X^{(k)}$ is accepted with a probability of $\exp((f(X^{(k)}) - f(X^{(0)}))/T)$ and taken as the initial point $X^{(0)}$ to continue the optimization. When N points are sampled, the function value $f(X^{(0)})$ at the optimal point is used as the promising index function of each feasible region to fix the next most feasible region. The pseudo-code of the optimization process is following.

```

σ(k) = Θ;
d(σ(k)) = 0;
Repeat
Partition the current promising
region σ(k) into Mσ(k) subregions.
T(k) = T(0) - dep(σ(k)) * ΔT
For i=1 to Mσ(k) + 1 do
For j=1 to N do
Generate_state_x(j);
δ = f(x(j)) - f(x(k));
if δ > 0 then k=j
else if random(0,1) < exp(-δ/T(k))
then
k=j;
Promising(i) = f(x(k));
End

```

```

if promising(i) > promising(m) then
m=i;
if m <= Mσ(k) then σ(k+1) = subregion(m);
    dep(σ(k)) = dep(σ(k)) + 1;
else backtrack(σ(k-1));
    dep(σ(k)) = dep(σ(k)) - 1;
until it reaches the maximum depth
and stabilizes.

```

We may notice that the same annealing temperature is applied to the sampling operation of $M_{\sigma(k)} + 1$ feasible regions at the same depth. When the depth of the feasible region is low, the annealing temperature is high and the probability of the worse solutions being accepted in sampling is also high. As the partitioning is moved on and the depth of the feasible region is increased, the annealing temperature used is comparatively low. At this temperature the probability of the worse solutions being accepted in sampling is hence low. NP/SA does not implement the complete annealing process of SA over every feasible region to be sampled.

3.2.2 Feasibility analysis on NP/SA

The openness of the NP algorithm allows for the introduction of other algorithm and thoughts. The NP algorithm implicitly contains a requirement: the modifications to the operators of the NP algorithm are allowed so long as two conditions are satisfied. They are: (a) the probability of each point in the feasible region being sampled is larger than 0, and (b) the promising index corresponds with the performance function of the singleton region.

Although NP/SA is different from the pure NP algorithm in fixing the optima in the partitioned regions, its essential sampling method is still random sampling. This ensures that the probability of each point in the feasible region being sampled is larger than 0. Therefore, NP/SA completely satisfies condition (a) of the NP algorithm. When the partitioning process of the NP/SA algorithm moves on to singleton, there is only one feasible point in the feasible region and only one point is obtained through sampling. The promising index at this point is the function value of this point; hence it corresponds with the performance function over the singleton. Thus, NP/SA satisfies the condition (b) of the NP algorithm.

In all, the introduction of SA into the NP algorithm satisfies the openness of the latter one, which ensures that NP/SA converges to the global optimal solution with a probability of 1.

3.2.3 Superiority analysis on NP/SA

As the NP algorithm evolves, the sequence of most promising regions $\{\sigma(k)\}_{k=1}^{\infty}$ forms a Markov chain with state space Σ . The singleton regions with the global optima are denoted as the absorbing states. In literature [3] and [4], L. Shi and S. Ólafsson proved that, the expected number of nested partitioning when the NP algorithm converges to the optimal solution is given by the following equation:

$$E[Y] = 1 + \sum_{\eta \in \Sigma_1} \frac{1}{P_{\eta}[T_{\sigma_{opt}} < T_{\eta}]} + \sum_{\eta \in \Sigma_2} \frac{P_{\Theta}[T_{\eta} < \min\{T_{\Theta}, T_{\sigma_{opt}}\}]}{P_{\eta}[T_{\Theta} < T_{\eta}] \cdot P_{\Theta}[T_{\sigma_{opt}} < \min\{T_{\Theta}, T_{\eta}\}]}$$

where T_{η} is the hitting time of state $\eta \in \Sigma$, i.e. the first time that the Markov chain visits the state, $P_{\eta}[\cdot]$ denotes the probability of an event given that the chain starts in state $\eta \in \Sigma$, σ_{opt} is the region corresponding to the unique global optimum, and $\Sigma_1 = \{\eta \in \Sigma \setminus \{\sigma_{opt}\} \mid \sigma_{opt} \subseteq \eta\}$, $\Sigma_2 = \{\eta \in \Sigma \mid \sigma_{opt} \not\subseteq \eta\}$ and $\Sigma = \{\sigma_{opt}\} \cup \Sigma_1 \cup \Sigma_2$ are disjoint state spaces.

NP/SA introduces SA into the NP algorithm, which increases the probability of obtaining the global optima in the sampled regions and further increases the probability of the state of the Markov chain changes in the correct direction. Consequently, probability $P_{\eta}[T_{\sigma_{opt}} < T_{\eta}]$ at time $\eta \in \Sigma_1$, probability $P_{\eta}[T_{\Theta} < T_{\eta}]$ at time $\eta \in \Sigma_2$, and probability $P_{\Theta}[T_{\sigma_{opt}} < \min\{T_{\Theta}, T_{\eta}\}]$ are increased while probability $P_{\Theta}[T_{\eta} < \min\{T_{\Theta}, T_{\sigma_{opt}}\}]$ at time $\eta \in \Sigma_2$ is decreased. The combined effect of these factors reduces the expected number of nested partitioning when the NP algorithm converges to the global optima, and thus speeds up the convergence of the algorithm.

4 A Numerical Example

In this section we consider a numerical example to illustrate the combined NP/SA method. In order to prove the optimization efficiency of the NP/SA method, we respectively implement the NP algorithm using the traditional random sampling as well as the NP/SA method for the minimization problem of Schaffer's f_6 function. Then we represent numerical results that compare the computation efficiency of the NP/SA method to a pure NP implementation.

The Schaffer's f_6 function is designed to have its global optimum at 0, surrounded by circular "valleys" designed to trap methods based on local search, see Fig. 2 [13].

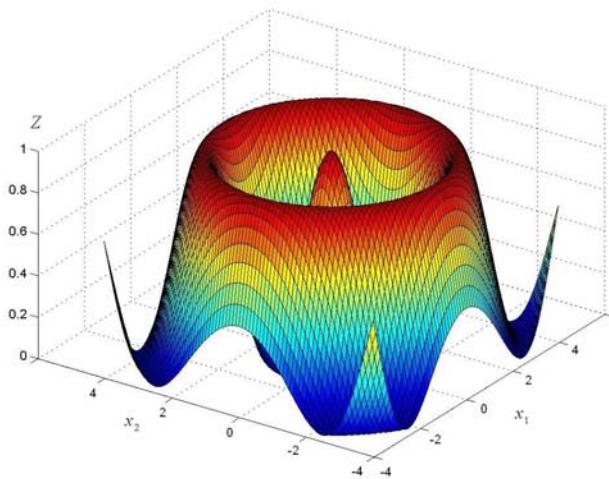


Fig. 2. The Schaffer's f_6 function

The function is given by

$$f(x_1, x_2) = 0.5 - \frac{(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5}{[1.0 + 0.001(x_1^2 + x_2^2)]^2}$$

To calculate its optimal solution, firstly we implement the pure NP method with the traditional random sampling operator. The following scheme is used. In each iteration the most promising region is partitioned into nine subregions. 30 points in each subregion is randomly sampled. The algorithm terminates at the tenth iteration. Secondly, we implement the NP/SA method and use the same nested partitioning scheme as the pure NP method. Moreover, the initial annealing temperature is 10 and the final annealing temperature is 0.0001. The comparison between the results of the two methods is shown in Table 1.

Table 1. Comparison of NP and NP/SA Performance

Algorithm	Result	Number of Times Backtracking Occurs
NP	0	6
NP/SA	0	0

As a result, after the adoption of the NP/SA method, the saved number of times we need calculate the performance function values is $\Delta C = 2N(M_{\sigma(k)} + 1)H = 2 \times 30 \times (9 + 1) \times 6 = 3600$

where H is the reduced number of times backtracking occurs after NP/SA is adopted.

These results give a strong indication that the NP/SA method obtained by introducing SA into the NP algorithm is very useful in combining the global optimum search capability of the NP algorithm and the local search capability of the SA algorithm, reducing the number of times backtracking occurs in

the nested partitioning, and making great improvements in calculation efficiency.

5 Conclusions

We have presented a new optimization algorithm that combines the NP algorithm and the SA algorithm. The resulting algorithm NP/SA retains the benefits of both algorithms, i.e., the global perspective and convergence of the NP algorithm and the powerful local search capabilities of the SA.

Since the random sampling operator of the NP algorithm is changed and the probability of obtaining the global optimal solution in each region is increased, the convergence is sped up, the number of times backtracking occurs in the nested partitioning is reduced, and hence the optimization efficiency is improved.

However, further theoretical and empirical development is needed for the algorithm. The NP/SA algorithm can be enhanced in several aspects. For example, we can use more elaborate partitioning, sampling and backtracking schemes if we have more knowledge of the specific decision problem. If we know that some solutions with certain properties are better than other solutions, we can add more weights on the regions containing these points. Future work will also focus on more numerical experiments and implementing the algorithm for complex decision problems in many fields to improve the current solving methods.

Acknowledgements

This research work is supported by the Natural Science Fund of China (**serial No.70501022**)

References:

- [1] Petr Wolf, Fuzzy logic in decision – making process, *Proceedings of the 3rd WSEAS/IASME International Conference on Dynamical Systems and Control*, 2007, pp. 282-287
- [2] Jihong Han, Yu Zhao, Yadi Wang, Zhiyong Zhou, A Problem Solving Mechanism for Formal Analysis of Cryptographic Protocols, *Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization*, 2007, pp.95-98
- [3] L. Shi, S. Ólafsson, Nested partitions method for global optimization, *Operations Research*, Vol. 4008, No. 3, 2000, pp. 390-407.
- [4] L. Shi, S. Ólafsson, N. Sun, New parallel randomized algorithms for the traveling

- salesman problem, *Computers & Operations Research*, 26, 1999, pp. 371-394.
- [5] L. Shi, S. Men, Optimal buffer allocation in production lines, *IIE Transactions*, Vol.35, No.1, 2003, pp. 1-10.
- [6] L. Shi, S. Olafsson, Q. Chen, A new hybrid optimization algorithm, *Computers & Industrial Engineering*, Vol.36, No.2, 1999, pp. 409-426.
- [7] L. Shi, S. Ólafsson, Q. Chen, An optimization framework for product design, *Management Science*, Vol. 47, No. 2, 2001, pp. 1681-1692.
- [8] S. Ólafsson, L. Shi, A method for scheduling in parallel manufacturing systems with flexible resources, *IIE Transactions*, Vol.32, No.2, 2000, pp. 135-146.
- [9] S. Ólafsson, N. Gopinath, Optimal selection probability in the two-stage nested partition method for simulation-based optimization, *Proceedings of the 2000 Winter Simulation Conference*, 2000, pp. 736-742.
- [10] S. Kirkpatrick, C. Gelatt, Jr., and M. Vecchi, Optimization by simulated annealing, *Science*, Vol. 220, No. 4598, 1983, pp. 671-680.
- [11] R.P. Barretto, L. Chwif, T. Eldabi, et al., Simulation optimization with the linear move and exchange move optimization algorithm, *Proceedings of the 1999 Winter Simulation Conference*, 1999, pp. 806-811.
- [12] M. A. Ahmed, T. M. Alkhamis, Simulation-based optimization using Simulated Annealing with ranking and selection, *Computers & Operations Research*, Vol.29, No.4, 2002, pp. 387-402.
- [13] Roberto Battiti, Mauro Brunato and Srinivas Pasupuleti, Do not be afraid of local minima: Affine Shaker and Practicle Swarm, *Technical Report # DIT-05-049*, Department of Computer Science and Telecommunications, University of Trento, Italy, 2005.