PARALLEL COMPUTING ADAPTIVE SIMULATED ANNEALING SCHEME FOR FUEL ASSEMBLY LOADING PATTERN OPTIMIZATION IN PWR's

Hyun Chul Lee, Hyung Jin Shim, and Chang Hyo Kim
Nuclear Engineering Department
Seoul National University
Seoul, Korea
lhc@tornado9.snu.ac.kr ; shj@tornado3.snu.ac.kr ; kchyo@gong.snu.ac.kr

ABSTRACT

An adaptive control scheme of simulated annealing(SA) parameters derived from polynomial-time cooling schedule is presented in terms of the efficiency enhancement of the SA algorithm. The parallel computing adaptive SA optimization scheme which incorporates the optimization-layer-by-layer(OLL) neutronics evaluation model is then applied to determining the optimum fuel assembly(FA) loading pattern(LP) in Korea Nuclear Unit 11(KNU 11) PWR using seven Pentium personal computers(three Pentium II 266 MHz and four Pentium Pro 200 MHz). It is shown that the parallel scheme enhances the efficiency of the SA optimization computation significantly but that it can get trapped in local optimum LP more frequently than the single processor SA scheme unless one takes preventive steps. As a way to prevent trapping of the parallel scheme in local optimum, we proposed using multiple seed LP's instead of a single LP with which the individual processors start each stage and discussed how to determine the multiple seed LP's. Because of high efficiency of the parallel scheme, acceptability of hybrid neutronics evaluation model which is slower but more accurate than OLL model into parallel optimization calculation is examined from the standpoint of the computing time. By demonstrating that the FA LP optimization calculation for the equilibrium cycle core of the KNU 11 PWR can be completed in less than an hour on seven Pentiums, we justified the routine utilization of the hybrid model in the parallel SA optimization scheme.

1. INTRODUCTION

The simulated annealing(SA) algorithm has presented a powerful tool for the in-core fuel management optimization computations.[1,2] But it has a disadvantage of lengthy turnaround time by requiring neutronics evaluation of several tens of thousands of trial loading patterns(LP's) in the optimization process. As a way to overcome this, Parks et al.[3] examined earlier the parallel computing SA capability with the parallel implementation of FORMOSA on an IBM 3090 vector machine with four processors. Recently we presented personal computer(PC)-based parallel computing SA fuel assembly(FA) LP optimization schemes[4] which incorporate a fast-running optimization-layer-by-layer(OLL) neural networks neutronics evaluation model (OLL model hereafter).[5] We demonstrated that they are very effective in reducing the turnaround time of the FA LP optimization computations in PWR but that they have a drawback to get trapped in the local optimum LP more frequently than the single processor SA scheme. The objectives of this paper is to reinvestigate the effectiveness of parallel computing SA optimization schemes by introducing the adaptive control of the SA parameters, to present a way to reduce the probability of any given parallel optimization run getting trapped in the local optimum LP, and to examine acceptability of incorporating more accurate but more time-consuming neutronics evaluation model than the OLL model from the standpoint of computing time.

The self-adaptive control of annealing parameters is prerequisite to minimize the total number of LP sampling and thus reduce the computing time of the SA optimization calculations. The FORMOSA code is known to use the self-adaptive control scheme proposed by Huang et al..[6,7] Except for reference 6 where the adaptive SA scheme of the FORMOSA code was first discussed, the details on it are rarely available in open literature. Here we introduce a slightly different adaptive scheme derived from polynomial-time cooling schedule[8], in setting the initial temperature, the temperature decrement ratio, and the stopping criteria of the optimization calculation and discuss its application to, and its effectiveness in, the FA LP optimization calculations.

In order to take advantage of the efficiency of the parallel scheme, one has to reduce the probability that it gets trapped in the local optimum LP.[4] We will discuss the cause for this. Through numerical experiments we will show that the probability for trapping in the local optimum LP depends on the way how one decides the seed LP's with which individual processors start at the beginning of each stage. Then we will suggest using multiple seed LP's to avoid trapping in the local optimum and present how to determine them.

The OLL model is an extremely fast neutronics evaluation tool. To use it, one has to spend the extra computing time for training all the necessary OLL networks for neutronics evaluation and sacrifice a certain degree of computational accuracy as well. In addition, the trained OLL model has a limited utility because they can be applicable only to the specific cycle core that has provided the training set of neutronics data. In order to overcome these disadvantages of the OLL model, we devised a hybrid neutronics model which employs the nodal method for global core calculation and utilizes the OLL networks only for prediction of the pin power peaking factors(PPPF's) of the individual FA's. The hybrid model is more accurate than, and free from the above-mentioned disadvantages of, the OLL model. Even though it is more time-consuming than the OLL model, we will demonstrate that the FA LP optimization can be carried out in less than an hour using the seven Pentium PC's(three Pentium II 266 MHz and four Pentium Pro 200 MHz) available at our graduate research laboratory and therefore the routine utilization of the hybrid model is justified.

2. STATEMENT OF THE OPTIMIZATION PROBLEM

The FA LP optimization problem here is to determine the fuel loading matrix X that will maximize the end-of-cycle(EOC) critical soluble boron concentration, $sb(X)$, under the radial pin power(F_{DH}) and maximum FA discharge burnup(B_{max}) constraints using the augmented cost function[2], $f(X)$,

$$f(X) = -sb(X) + \boldsymbol{q}_1 \sqrt{\sum_{l=0}^{L}\sum_{m=1}^{NA} \max\left[\left(P_m^l(X) - F_{\Delta H}\right),0\right]^2} + \boldsymbol{q}_2 \sqrt{\sum_{m=1}^{NA} \max\left[\left(B_m(X) - B_{\max}\right),0\right]^2} \qquad (1)$$

3. PARALLEL COMPUTING SA OPTIMIZATION SCHEME

The SA optimization algorithm consists of generation of a new LP(X_{new}) from the current LP(X_{cur}), evaluation and decision making on the acceptance of the X_{new} according to the acceptance probability,

$$P = \begin{cases} e^{-\Delta f / c}, & \text{if } \Delta f = f(X_{new}) - f(X_{cur}) > 0 \\ 1, & \text{otherwise} \end{cases} \qquad (2)$$

In parallel computing SA scheme, N(>1) processors are employed simultaneously to follow the SA

optimization algorithm. As in the case of the single processor SA scheme, the whole process is divided into annealing stages in each of which the preset number of LP's are subject to SA sampling with the constant temperature parameter c set by the specified annealing schedule. A given stage usually ends either when the total number of the LP generation reaches the preset limit per stage(N_{stg}) or when the number of the accepted LP's reaches the preset limit(N_{acp}).

In reference 4, we presented three different parallel schemes. But here we consider only one parallel scheme designated as the scheme(II). In this scheme, all the N processors engage simultaneously in SA optimization process and one master processor communicates with all the other N-1 slave processors to count the total number of LP's that all the N processors have sampled and accepted. Thus all the processors complete a given stage at the same time by the stage end criteria(N_{stg} or N_{acp}) recorded on the master processor and start the next stage with the X_{cur} provided by the master processor.

4. SELF-ADAPTIVE CONTROL OF THE SA PARAMETERS

The SA optimization algorithm requires setting the initial temperature, c_0, temperature decrement ratio, a_k, and the stopping criteria. For the self-adaptive control of these parameters, we use a polynomial-time cooling schedule presented in reference 8. In this schedule, the initial temperature is determined by

$$c_0 = \frac{\overline{\Delta f}^{(+)}}{\ln\left(\dfrac{m_2}{m_2 \chi - m_1(1-\chi)}\right)} \tag{3}$$

where the χ is the initial acceptance ratio, and m_1 and m_2 are the number of cost increasing and cost decreasing LP's, respectively. The $\overline{\Delta f}^{(+)}$ is the average of the cost difference over the m_2 cost increasing LP changes.

The temperature decrement ratio a_k is given by

$$\alpha_k = \frac{1}{1 + \dfrac{c_k \ln(1+\delta)}{3\sigma_k}}, \tag{4}$$

where the c_k and σ_k are the temperature and the standard deviation of the cost function at the stage k, respectively. The distance parameter, d, is some small positive constant close to zero, and is a measure of closeness of equilibrium distributions of two Markov chains at two successive cooling stage.

To end the optimization computation, we use the stopping criteria derived as follows :

$$\frac{\langle f \rangle_k - f^{opt}}{\langle f \rangle_\infty - f^{opt}} \approx \frac{1}{\langle f \rangle_\infty - f^{opt}} \cdot c_k \left.\frac{\partial \langle f \rangle_c}{\partial c}\right|_{c=c_k} \approx \frac{(\sigma_k)^2}{\left(\langle f \rangle_\infty - f_k^{best}\right) c_k} < \varepsilon_s, \tag{5}$$

where

$\langle f \rangle_\infty$ = the mean cost function at initial hot temperature,

$\langle f \rangle_k$ = the mean cost function at stage k,

f^{opt} = the cost function of the optimum LP,

f_k^{best} = the cost function of the best LP recorded till the stage k,

ε_s = the stopping parameter.

Equation (5) states that the distance between the $\langle f \rangle_k$ and f^{opt} is a small fraction of the distance between the $\langle f \rangle_\infty$ and f^{opt}. In driving Eq. (5), we assumed that $\langle f \rangle$ varies linearly with c and used $\frac{\partial \langle f \rangle}{\partial c} = \frac{\sigma^2}{c^2}$ along with an approximation, $f^{opt} \approx f_k^{best}$.

The initial temperature and stopping criteria are different from those presented by Huang et al.[7] But the temperature decrement ratio of Eq.(4) can be shown to be similar to the one presented by Huang et al. and adopted by FORMOSA code. In this conjunction, let us note $\lambda c_k/s_k << 1$ with $\lambda = \ln(1+\delta)/3$. Then

$$\alpha_k = \frac{1}{1+\frac{\lambda c_k}{\sigma_k}} \approx 1-\frac{\lambda c_k}{\sigma_k} \approx exp\left(-\frac{\lambda c_k}{\sigma_k}\right) \tag{6}$$

This is the temperature decrement ratio used by the FORMOSA code.

The standard deviation may fluctuate a lot stage by stage. This may cause large changes in the temperature decrement ratio, α_k, which in turn lead to unwanted premature annealing. As suggested by reference 9, we avoided this using the smoothed s_k defined by[10]

$$s^s_{k+1} = (1-w)\, s_{k+1} + w\, s^s_k \frac{c_{k+1}}{c_k} \tag{7}$$

The weighting factor w is taken as 0.9.

5. HYBRID NEUTRONICS EVALUATION MODEL

The hybrid neutronics evaluation model consists of the nodal diffusion theory method for global core neutronics calculation and the OLL neural network for the prediction of PPPF. As shown in Fig.1a, the NEMSNAP[11], the nodal expansion method code of Seoul National University, computes global core neutronics characteristics such as normalized power and burnup of individual FA's, net currents and group fluxes at the nodal interfaces, and critical soluble boron concentration. The OLL networks are used to predict the homogeneous PPPF ($PPPF_{hom}$) of individual FA's inputting the nodal powers and thermal group nodal interface fluxes and net currents(cf. Fig.2). Thus the OLL networks are equivalent to the pin power reconstruction subprogram in the nodal code. For the purpose of comparison, Fig.1b shows the OLL model. In this model the OLL networks are used to compute even the global core design parameters. Because the outputs of the OLL model are confined to normalized powers and burnups of individual FA's and critical soluble boron concentration, the PPPF predictions by the OLL networks in this model rely only on the normalized power and the assembly discontinuity factors(ADF's) of the FA of interest and surrounding eight FA's. Thus the PPPF prediction becomes less accurate than the hybrid model, as will be shown later.

6. NUMERICAL RESULTS AND DISCUSSION

The parallel computing adaptive SA scheme is applied to the optimum FA LP search of the equilibrium cycle core of the KNU 11 PWR networking seven Pentiums with the help of message passage interface(MPI).[12] In the following we present how to determine free parameters of the adaptive SA algorithm, a way to avoid trapping in the local optimum in the parallel SA calculations and its effectiveness.

6.1 THE ADAPTIVE CONTROL OF THE SA PARAMETERS

The adaptive SA scheme has some free parameters; c, l, e_s, N_{acp}, and N_{stg}. Because their optimum values are a priori unknown, they must be determined through numerical tests. The c determines the initial hot temperature condition, which must be hot enough for a satisfactory annealing result. The closer to 1 the c is, the hotter the c_0 becomes. We observed that the c greater than 0.99 gives the c_0 which is about ten times the standard deviation of the cost functions at the initial hot temperature condition and thus $c \geq 0.99$ is considered acceptable. The value l is problem dependent. The less the l is, the slower the annealing and the higher the probability of getting near optimal LP is. And vice versa. Therefore the total number of LP sampling and the success probability that any given SA run reaches the near optimal LP depends on the selection of l value. In our problem, we observed that the optimum values of l lie in 0.1 ± 0.02. The stopping parameter e_s is a measure of how far the $<f>_k$ is off the optimal cost. The e_s=0.005 is the reasonably good choice from the standpoint of the total number of LP sampling and the success rate of getting near optimum per optimization run. The N_{stg} is related to the length of the Markov chains while the N_{acp} to the establishment of the equilibrium condition. We observed that they are problem-dependent and that N_{stg}=500 and N_{acp}=50 are reasonably good in our LP optimization problem.

Table I compares performance of the non-adaptive versus adaptive SA algorithms. The total number of the LP sampling and the number of stages are reduced considerably by the adaptive control of the temperature decrement ratio. They are also affected by the stopping criteria. The typical stopping criteria for termination of the annealing when the number of accepted LP's is less than 2 requires more LP sampling than otherwise. The stopping criteria proposed by Huang et al.[7] appears inferior to Eq. (5) because the former occasionally leads to premature termination of the annealing process, as shown in the average of the optimum cost functions in the forth and fifth columns of Table I. The standard deviation of the cost function fluctuate a lot while the smoothed standard deviation changes monotonically with the advancing stage number(cf. Fig. 3). Because of this, the use of the unsmoothed standard deviation in the stopping criteria, Eq. (5), leads to premature termination of the annealing process in most of the optimization runs as noted in the sixth column and fails to ensure near optimum LP. Therefore, the use of the smoothed s_k^s is recommended for the adaptive SA algorithm.

6.2 PARALLEL COMPUTING LP OPOPTIMIZATION WITH OLL MODEL

The parallel computing SA scheme is apt to get trapped in local optimum LP more frequently than the single processor SA scheme. Table II shows this by comparing the performance of the single processor SA versus the parallel computing SA optimization computations. The average of the optimum cost functions in the parallel optimization calculations with the same SA parameter as the single processor SA calculation(l=0.1, N_{acp}=50, N_{stg}=500) are much larger than that of single processor scheme, e.g., -427.7 because the former has the tendency to stop the optimization computations at much earlier stages(k=46) than the latter(k=64). Consequently, it frequently ends up with a local optimum LP instead of a global optimum LP. The adjustment of the SA parameters such as l, N_{stg}, and N_{acp} is tried. But it failed to prevent the trapping of the local optimum as noted in Table II. In this conjunction, any LP with the EOC SB concentration less than 430 ppm is taken as the local optimum LP.

In N-processor parallel SA scheme, each processor samples and evaluates roughly N_{stg}/N LP's every stage. If all the N processors start a given stage with the same seed LP provided by the master processor, the LP search space that N processors cover is apt to be much limited than that the single processor SA scheme covers. In an extreme case that $N=N_{stg}$, the LP space that the N processors cover will consist of those LP's generated from a single binary exchange of two FA's from the same seed LP, which must be smaller space than that sequentially generated in a single processor SA optimization. This may cause relatively easy trapping in the local optimum LP in the parallel scheme. Figure 4 explains this in terms of the stagewise standard deviation of the cost functions of the LP's accepted at each stage. It shows that the s_k^s of the parallel scheme approaches more rapidly to the limiting value and triggering the stopping criteria at much earlier stage, which causes the parallel scheme to end up with the local optimum very easily. To avoid this, the parallel scheme needs to cover the LP search space as wide as the single processor scheme. One way to achieve this is to let the parallel scheme start each stage with multiple seed LP's by allowing each processor to have its own seed LP which is determined in the following way. At the end of a stage, each processor has its own current LP. Thus the N-processor parallel scheme has N current LP's. We number them in the predetermined order. Starting with its own current LP, each processor compares the N–1 LP's of the other processors successively in the cyclic order to decide its seed LP for the ensuing stage according to the acceptance probability;

$$P = \begin{cases} e^{-\frac{\Delta f}{pc_k}}, & if\ \Delta f = f_i - f_{cur} > 0 \\ 1, & otherwise \end{cases} \qquad (i = 1,2,\cdots N-1) \tag{8}$$

The LP which is last accepted becomes the seed LP of the processor. The parameter p(>1) in Eq. (8) is chosen to create hotter condition, so that a given processor can accept the current LP of other processors with high probability. By determining the seed LP's of N processors at the beginning of each stage this way, one can avoid starting the stages with only a single seed LP and thus hopefully reduce the tendency of the parallel scheme getting trapped in the local optimum. Figure 5 depicts the stagewise standard deviation of the parallel scheme with the multiple seed LP's at each stage.. The stagewise standard deviation of the parallel scheme now varies in a very similar way as that of the single processor SA scheme. The higher the p is, the slower the change in the standard variation becomes and thus the number of the total LP sampling increases. Table III shows performance of the parallel adaptive SA scheme with multiple seed LP's. We observed the p=3 is reasonable choice from the efficiency standpoint.

6.3 PARALLEL SA OPTIMIZATION WITH HYBRID NEUTRONICS MODEL

The hybrid model is designed to overcome major disadvantages of the OLL model at the expense of the computing time. Table IV shows a comparison of the PPPF prediction accuracy of the hybrid and OLL models. The hybrid model can predict the PPPF with the mean and maximum relative errors less than 1 and 5 % with respect to 4 N/FA nodal calculation, respectively. In comparison, the OLL model shows much larger mean and maximum errors. Needless to mention, the higher PPPF prediction accuracy of the hybrid model is ascribed to the fact that the OLL networks in the hybrid model utilizes more informations than those in the OLL model. As for the computing time, the hybrid model takes about one second on Pentium II 266 MHz for a cycle burnup analysis of a given LP of the KNU 11 PWR on 10 burnup steps, while the OLL model takes only 50 milliseconds.

Table V shows the performance of parallel SA optimization computations with the incorporation of the hybrid model on seven Pentiums. For a reasonable statistics, the 30 parallel optimization runs are made. The total number of LP sampling is about 15,000 on the average. The turnaround time of each optimization run is mostly less than an hour. The parallel scheme has produced near optimum LP's as frequently as the single processor scheme. The six to seven out of ten optimization runs has reached the near optimum LP.

CONCLUSIONS

The adaptive control scheme of SA parameters introduced herein is slightly different from that of the FORMOSA code. The initial temperature setting, the stopping criteria, and use of the smoothed standard deviation in the temperature decrement ratio and the stopping criteria are new features that are straightforwardly implemented and contribute to efficiency enhancement of the SA optimization computations. The parallel SA scheme can overcome the major disadvantage of the lengthy turnaround time of the single processor SA optimization scheme. In order to avoid easy trapping of the parallel scheme in the local minimum LP, however, one must take the preventative scheme like use of the multiple seed LP's at each annealing stage that we suggested in this paper. The high efficiency of the parallel scheme may allow to incorporate the slower but more accurate neutronics model like the hybrid model than the OLL model into SA algorithm. We justified the incorporation of the hybrid model into the parallel SA scheme by demonstrating that the FA LP optimization of the KNU 11 PWR can be carried out in less than one hour with seven Pentiums.

In conclusion, the parallel computing adaptive SA optimization scheme is very efficient and it can be carried out with little or no extra cost on computer facilities in the PC-affluent working environment of any fuel management groups nowadays. The acceptable turnaround time of less than an hour on seven Pentiums as well as the high success rate of getting near optimum LP per optimization run warrant the routine utilization of the parallel computing adaptive SA optimization scheme on multiple personal computers.

ACKNOWLEDGEMENTS

This work was supported partly by Korea Nuclear Fuel Company and partly by Korea Science and Engineering Foundation.

REFERENCES

1. D. J. Kropaczek and P. J. Turinsky, In-Core Fuel Management Optimization for Pressurized Water Reactors Using Simulated Annealing, Nucl. Tech., 95, 9 (1991).

2. J. G. Stevens, K. S. Smith, K. R. Rempe, and T. J. Downer, Optimization of Pressurized Water Reactor Shuffling by Simulated Annealing with Heuristics, Nucl. Sci. Eng., 121, 67 (1995).

3. G. T. Parks, P. J. Turinsky, and G. I. Maldonado, Scientific Excellence in Supercomputing, The IBM 1990 Contest Prize Papers, Vol. I, Balwin Press, Athens, GA (1992).

4. J. H. Lee, H. C. Lee, and C. H. Kim, Parallel Computing Fuel Assembly Loading Pattern Optimization by Simulated Annealing Method, Trans. Am. Nucl. Soc. 80, 228 (1999).

5. J. H. Lee, H. J. Shim, C. S. Jang, and C. H. Kim, Incorporation of Neural Networks into Simulated Annealing Algorithm for Fuel Assembly Loading Pattern Optimization in a PWR, Proc. Int. Conf. on Phys. of Nuclear Science and Technology, Oct 5-8, 1998, Long Island, NY, USA, Vol. 1, p. 75.

6. D. J. Kropaczek, P. J. Turinsky, G. T. Parks, and G. I. Maldonado, The Efficiency and Fidelity of The In-Core Nuclear Fuel Management Code FORMOSA-P, Reactor Physics and Reactor Computations, Israel Nuclear Society and European Nuclear Society, Tel-Aviv (1994).

7. M. D. Huang, F. Romeo, and A. Sangiovanni-Vincentelli, An Efficient General Cooling Schedule for Simulated Annealing, IEEE Int. Conf. On Computer Aided Design, pp. 381-384 (1986)

8. E. Aarts and J. Korst, Simulated Annealing and Boltzmann Machines, John Wiley & Sons, New York, NY, USA (1989).

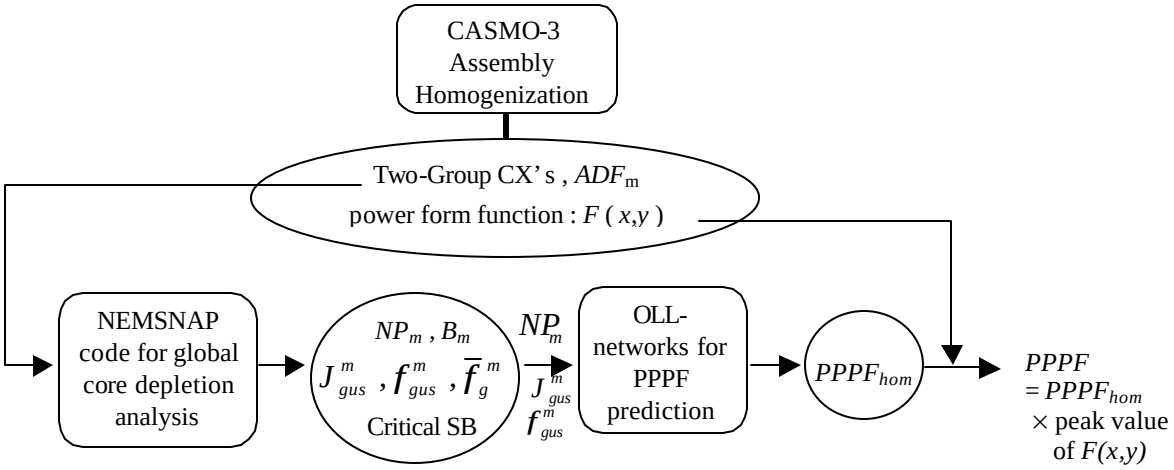
9. R. Diekmann, R. Lüling, and J. Simon, Problem Independent Distributed Simulated Annealing and Its Application, Technical Report PC², TR -003-93 (1993).

10. R. Otten and L. van Ginneken, The Annealing Algorithm, Kluwer Academic Publishers (1989).

11. T. K. Kim and C. H. Kim, Verification of the Pressurized Water Reactor Analysis Codes : CASMO/NEMSNAP, Proc. Korean Nucl. Soc., Spring Meeting, Pohang, Korea, May 27-28 (1994).

12. P. S. Pacheo, Parallel Programming with MPI, Morgan Kautmann Publisher Inc., San Fransisco, USA (1997).

(a) Hybrid Neutronics Evaluation Model.



(b) OLL Networks-Based Neutronics Evaluation Model.

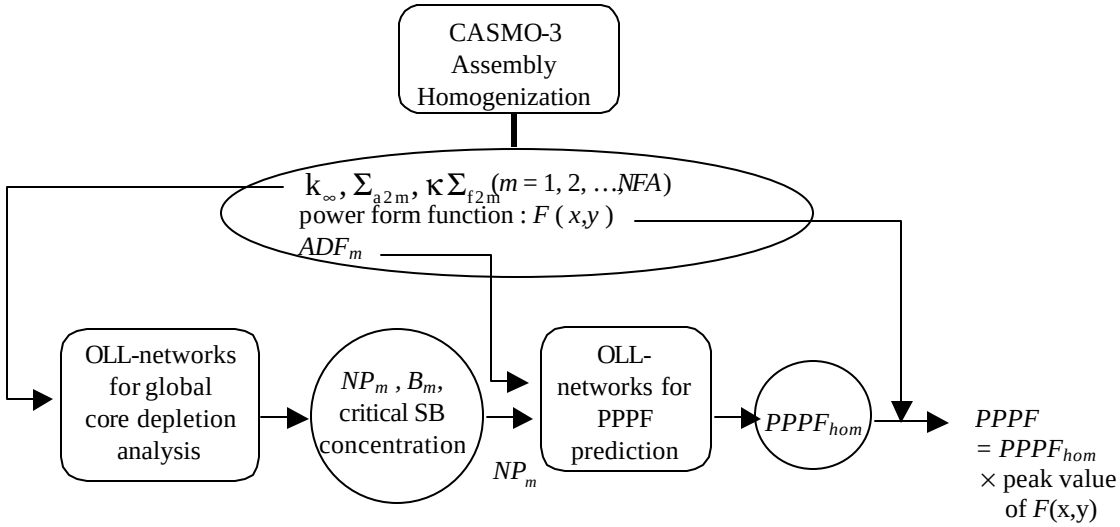
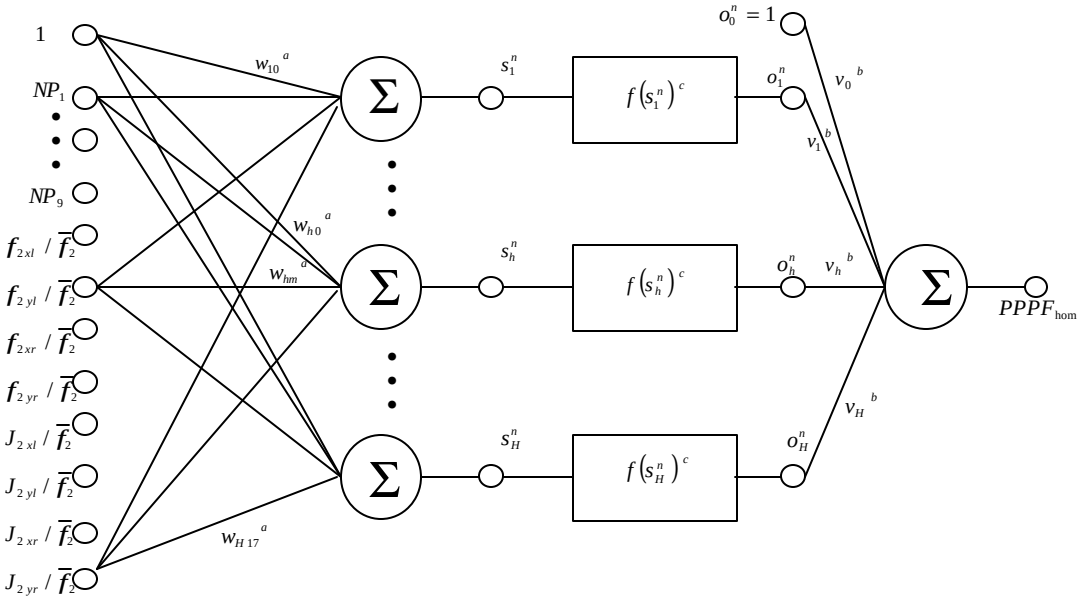


Figure 1. Hybrid and OLL Networks-Based Neutronics Evaluation Model.



[a] w_{hm} (m=0,1, … ,7, h = 0,1, .., H) = weights for hidden neurons
[b] v_h (h = 0, 1, … H) = weights for output neurons
[c] $f(s)$ = sigmoidal function defined by $(1+e^{-s})^{-1}$

Figure 2. OLL Networks for Prediction of $PPPF_{hom}$ in Hybrid Neutronics Evaluation Model.

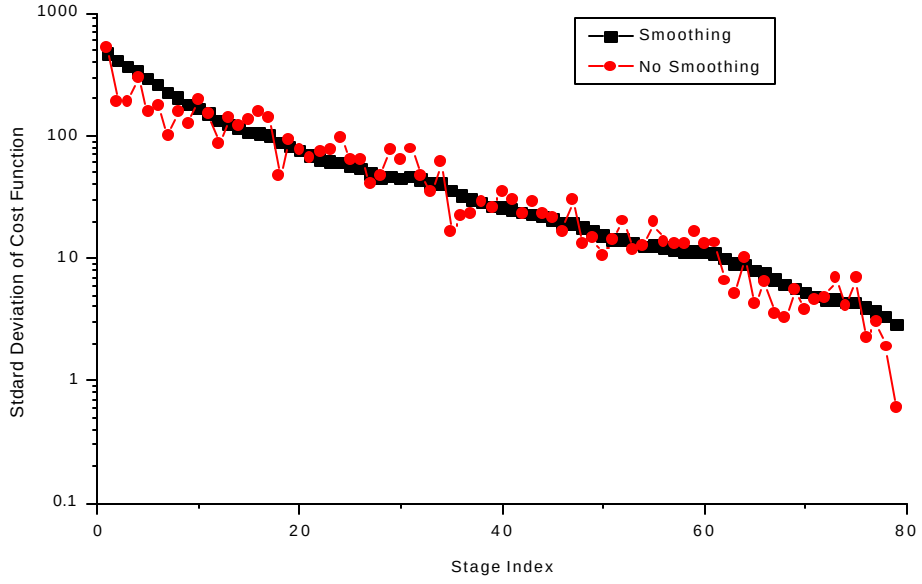


Figure 3. Effect of Smoothing on Satgewise Standard Deviation of Cost Function

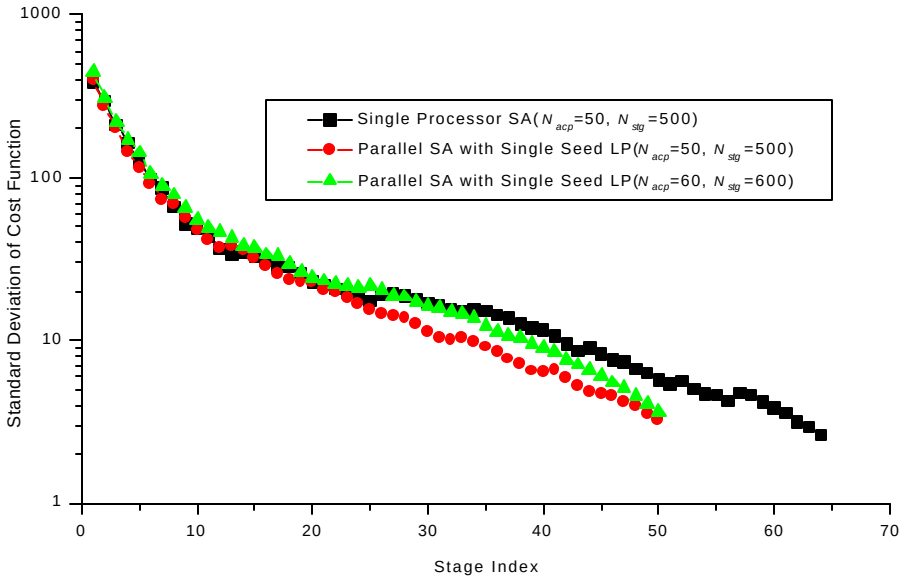


Figure 4. Stagewise Standard Deviation of Cost Function.

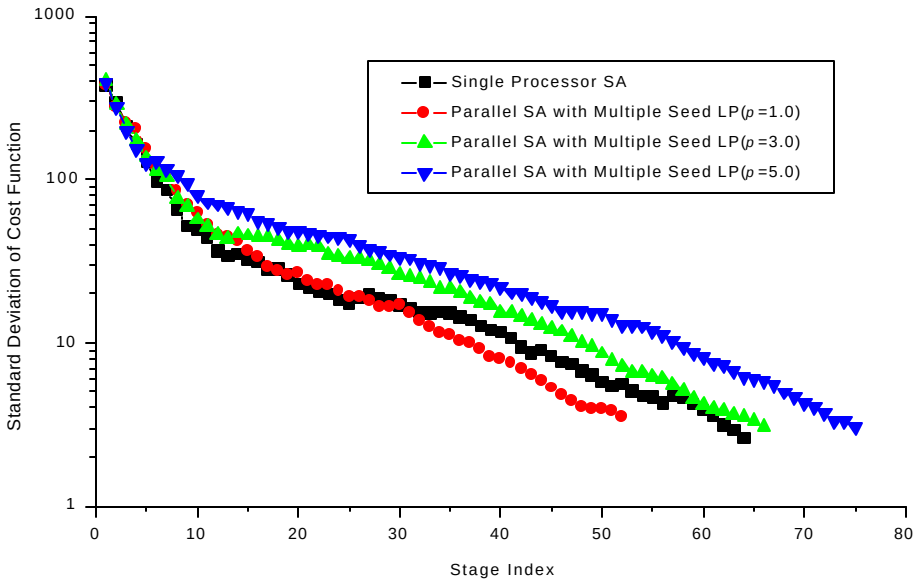


Figure 5. Stagewise Standard Deviation of Cost Function.

Table I. Performance Comparison of Nonadaptive versus Adaptive SA Algorithms*
(N_{acp}=50 , N_{stg}=500 , c=0.99 , l=0.1 , e_s=0.005)

SA Algorithm	Nonadaptive SA	Adaptive SA (I)	Adaptive SA (II)	Adaptive SA (III)	Adaptive SA (IV)	Adaptive SA (V)
Initial Temperature c_0	Eq. (3)	Eq. (3)	Eq. (3)	Eq. (3)	Eq. (3)	Eq. (3)
Cooling Parameter a_k	Constant (=0.95)	Eq. (6) with σ_k^s	Eq. (6) with σ_k^s	Eq. (6) with σ_k^s	Eq. (6) with σ_k	Eq. (6)) with σ_k^s
Stopping Criteria	Number of Accepted LP 's < 2	Number of Accepted LP's < 2	$\frac{(\Delta f)_k^{\max}}{f_k^{\max} - f_k^{\min}} > 0.95$ [a]	$\frac{(\Delta f)_k^{\max}}{f_k^{\max} - f_k^{\min}} > 0.98$ [a]	Eq. (5) with σ_k	Eq. (5) with σ_k^s
Mean Number of LP Sampling $\pm s$ [b]	48059±4582	29480±1867	11213±11528	20963±10697	12343±2387	16975±1680
Mean Number of Stages $\pm s$ [b]	193±9	101±5	47±36	70±28	66±6	64±4
$\langle f^{opt} \rangle \pm s$ [b]	-434.2±10.3	-431.0±11.1	-380±50.4	-408.9±53	-418.4±16.4	-427.7±12.0

* Numerical results are derived from 30 independent optimization runs.
[a] Reference 7.
[b] s : standard deviation.

Table II. Performance of Parallel Adaptive SA Scheme with Single Seed LP.* (c=0.99 , e_s=0.005)

Parameters		Number of Stages	Total Number of LP Sampling	EOC Soluble Boron (ppm)	PPPF	Cost Function	Elapsed Time [a] (sec)
N_{acp}=50 N_{stg}=500	λ=0.1	46±6	10504±2907	401±31	1.461±0.012	-395.0±29.4	89±22
	λ=0.07	67±11	15819±3948	409±23	1.460±0.011	-405.6±21.6	134±32
	λ=0.05	92±20	20880±8043	417±27	1.462±0.010	-412.1±25.0	177±64
N_{acp}=60 N_{stg}=600	λ=0.1	48±4	13825±2605	410±20	1.458±0.012	-407.0±17.8	119±20
N_{acp}=70 N_{stg}=700	λ=0.1	52±4	18209±2859	413±26	1.458±0.008	408.9±24.0	148±18
N_{acp}=80 N_{stg}=800	λ=0.1	55±5	22572±3641	419±22	1.461±0.009	-414.2±20.2	185±30

* Numerical results are derived from 20 independent optimization runs.
[a] On seven Pentium(Three Pentium II 266MHz and Four Pentium Pro 200MHz)

Table III. Performance of Parallel Adaptive SA Scheme with Multiple Seed LP.*
(N_{acp} =50 , N_{stg}=500 , c=0.99 , l=0.1 , e_s=0.005)

Parameter p	Number of Stages	Total Number of LP Sampling	EOC Soluble Boron (ppm)	PPPF	Cost Function	Elapsed Time [a] (sec)
1.0	51±4	12547±1969	414±23	1.455±0.008	-411.2±22.2	113±18
3.0	67±5	17951±2170	430±14	1.458±0.008	-427.8±14.4	150±19
5.0	75±4	20674±1601	434±12	1.461±0.008	-429.3±12.9	174±16
7.0	82±7	23696±2680	436±11	1.457±0.008	-433.5±10.9	201±22

* Numerical results are derived from 20 independent optimization runs.
[a] On seven Pentium(Three Pentium II 266MHz and Four Pentium Pro 200MHz)

Table IV. Comparison of the PPPF prediction accuracy of the hybrid and OLL models.

Core Burnup State	Hybrid Model		OLL Model	
	ε_{ave} [a] (%)	ε_{max} [b] (%)	ε_{ave} [a] (%)	ε_{max} [b] (%)
BOC (0 MWD/T)	0.91	4.77	1.60	11.26
MOC (7000 MWD/T)	0.74	3.48	1.20	8.70
EOC (14000 MWD/T)	0.69	2.90	0.97	4.80

[a] ε_{ave} = average relative error in PPPF prediction
[b] ε_{max} = maximum relative error in PPPF prediction.

Table V. Performance of Parallel Adaptive SA Algorithm with Hybrid Neutronics Evaluation Model*
(N_{acp} =50 , N_{stg}=500 , c=0.99 , l=0.1 , e_s=0.005 , p=3.0)

	Number of Stages	Number of LP's Sampled	CPU Time (min)	Elapsed Time [a] (min)	EOC Soluble boron (ppm)	PPPF	Cost Function
Mean value ±s [b]	65±3	15107±1863	368±44	55±6	432±15	1.458±0.010	-430.7±15.1

* Numerical results are derived from 30 independent optimization runs.
[a] On seven Pentium(Three Pentium II 266MHz and Four Pentium Pro 200MHz)
[b] s : Standard deviation.