ASSESSMENT OF PARALLEL SIMULATED ANNEALING PERFORMANCE WITH THE NEXUS/ANC9 CORE DESIGN CODE SYSTEM

Boyan D. Ivanov[1], and David J. Kropaczek[2]

[1]Westinghouse Electric Company
1000 Westinghouse Dr, Cranberry Township, PA 16066, USA

[2]Oak Ridge National Laboratory
One Bethel Valley Road, Oak Ridge, TN 37830, USA

ivanovbd@westinghouse.com, kropaczekdj@ornl.gov

ABSTRACT

The method of parallel simulated annealing is being considered as a loading pattern optimization method to be used within the framework of the latest Westinghouse core design code system NEXUS/ANC9. A prototype version of NEXUS/ANC9 that incorporates the parallel simulated annealing method was developed. The prototype version was evaluated in terms of robustness, performance and results. The prototype code was used to optimize LPs for several plants and cycles, including 2-loop, 3-loop and 4-loop Westinghouse plants. Different fuel assembly lattices with IFBA, WABA and Gadolinium burnable absorbers were also exercised in these cores. Different strategies were evaluated using different options in the code. Special attention was paid to the robustness and performance when different number of parallel processes were used with different size of Markov chain.

KEYWORDS: Parallel Simulated Annealing, Loading Pattern Optimization, NEXUS/ANC9

1. INTRODUCTION

Westinghouse recently developed and released its latest core design code system NEXUS/ANC9 [1]. This new core design code system was significantly upgraded from its predecessor, to such an extent, that the existing Loading Pattern (LP) optimization tools could not be used with the new system. A project was initiated to identify the best LP optimization methodology that can seamlessly work with NEXUS/ANC9. The first step was to investigate the existing Westinghouse LP optimization codes ALPS [2] and PEARLS [3] as possible candidates to be updated to work with the latest core design system. The ALPS code uses reactivity matching, groups the assemblies by reactivity, and performs binary exchanges to generate initial LPs that are further improved by performing more binary exchanges based on systematic sequence during cycle depletion [2]. Of the final LP candidates, a few are selected by the designer and furthermore explicit calculations are performed using the ANC8 core design code system with additional manual assembly transitions and rotations to arrive at the final LP. ALPS uses its own simplified cross-section library and neutronic solver which results in somewhat different results when an ALPS pattern is used directly in ANC.

The PEARLS code resembles the ALPS methodology in the sense that it also groups the assemblies in coarse batches by reactivity to limit the search domain. In contrast to ALPS, instead of using reactivity matching and binary exchanges, PEARLS uses heuristics to further limit the search domain and then performs mathematical optimization to find the best pattern within the selected search domain, which consists of a number of coarse batch patterns. The mathematical optimization is then performed using a sensitivity matrix that approximates the core simulator solution in 2D. PEARLS is a deterministic code in the sense that the solution does not depend on pseudo-random selection processes and instead is based on the solution of a system of linear equations that explores all possible combinations within a coarse batch resulting in the optimal LP for that coarse batch. Bearing in mind that a coarse batch consists of roughly 10^{20} LPs for a 4-loop core with eighth-core symmetry, PEARLS examines a vastly larger domain than ALPS. Hundreds or even thousands of coarse batch loading patterns can be examined with Pearls very efficiently. The PEARLS method for coarse batch selection and use of sensitivity matrix perturbation theory does however mean that the global optimum LP might not be found if the best coarse batch is not selected. An advantage of PEARLS is that it uses the licensed core simulator to perform the core calculations for the final optimization, so that in principal its final LP can be used directly in the reload safety analysis.

After investigating ALPS and PEARLS it was realized that the cost for integration would be high and a decision was made to look for other methodologies like Simulated Annealing (SA), Genetic Algorithm (GA) and Artificial Intelligence (AI). Initial investigation was performed where all advantages and disadvantages of these methods were evaluated in terms of flexibility, performance, capabilities and cost for integration. After careful evaluation the method of Parallel Simulated Annealing (PSA) was selected for further evaluation. PSA is the parallel implementation of the SA method. SA is a single-solution, nature inspired, metaheuristic method. The name and development of this method is inspired by the process of annealing, which is the heating and controlled cooling of a material to increase the size of its crystals and reduce their defects and internal stresses and therefore strengthen the material. In SA, there is a “temperature” variable that simulates the heating and cooling process. Initially the temperature is set high and then it is gradually allowed to cool. While the temperature is high the search method is allowed, with higher probability, to accept changes to the LP that are worse than the current “best” LP. This allows the search method to escape any local optimum early in the search process. With the temperature decreasing the probability of accepting such a solution decreases and the search gradually stabilizes in an area of the search space which hopefully contains the global optimum. The gradual decrease of the temperature, which allows the algorithm to escape local optima, is what makes SA very effective in large problems with multiple local optima like the LP search problem.

The SA method is inherently a serial algorithm in which each successive solution depends on the previous one. Mixing of States and Markov Chains are utilized in PSA to parallelize the calculation which has demonstrated improved performance [4, 5]. Single cooling schedule is used with parallel Markov chains. The cooling temperature stays the same until all solutions within the Markov chains are completed. At each cooling step sampling from a pool of current best solutions is performed. The pool of best solution is assembled from the solutions in the Markov chains in the previous cooling step. Once all solutions within the Markov chains are completed the temperature is evaluated and decreased according to the cooling schedule. Based on the new temperature the probability of accepting solutions is changed accordingly.

The PSA method as embodied in the optimization code of reference [6] was coupled with the ANC9 code in a prototype version and was evaluated in terms of robustness, performance and results. The prototype code was used to optimize LPs for several plants and cycles, including 2-loop, 3-loop and 4-loop Westinghouse plants. Different fuel assembly lattices with IFBA, WABA and Gadolinium burnable absorbers were also exercised in these cores. The robustness and performance of the code was evaluated when different number of parallel processes were used with different size of Markov chains.

2. SENSITIVITY ANALYSIS

2.1. Test problem

The prototype version of NEXUS/ANC9 with PSA was tested on a 4-loop Westinghouse PWR core to exercise the performance and robustness of the code. This is a unique 18-month cycle having several 17x17 assembly designs including Wet Annular Burnable Absorber (WABA), Integrated Fuel Burnable Absorber (IFBA) and two enrichments of 4.6 w/o and 4.95 w/o. Several fuel assembly designs had both IFBA and WABA while others contained only IFBA or no burnable absorber at all. The number of feed assemblies was preset to 96 and the code was set to optimize on the enrichment split. The code was restricted from placing fresh assemblies on the periphery and the center bundle was reused from the previous cycle. The problem was formulated as a constraint satisfaction problem that was solved with the usual constraints on FdH, End of Cycle (EOC) boron concentration, Hot Full Power (HFP) boron concentration, and peak pin burnup. Additional constraints on batch size and batch enrichment were also imposed. Different perturbations were utilized in the search like change the batch size, change the fresh fuel type, change orientation, etc.

2.2. Results of the sensitivity analysis

The code can run with multiple Markov chains in parallel. Different size of the Markov chains can also be selected. These two input parameters control the number of solutions that the code will execute per cooling step which results in different running time, robustness of the solution and exhaustiveness of the search. The goal of this study is to find proper inputs that would lead to acceptable results in reasonable running time. In this work, the PSA method is exercised with 49 and 105 parallel Markov chains. The following sizes of the Markov chain were selected for each run 5, 10, 15 and 20. Each combination is then run ten times with different random seed to examine the mean and standard deviation for each combination allowing by that the assessment of the robustness of the solution when different number of solutions were used per cooling step.

Figure 1 below shows the evolution of the objective function as a function of the cooling steps when 49 parallel groups were used with the four different sizes of the Markov chain.

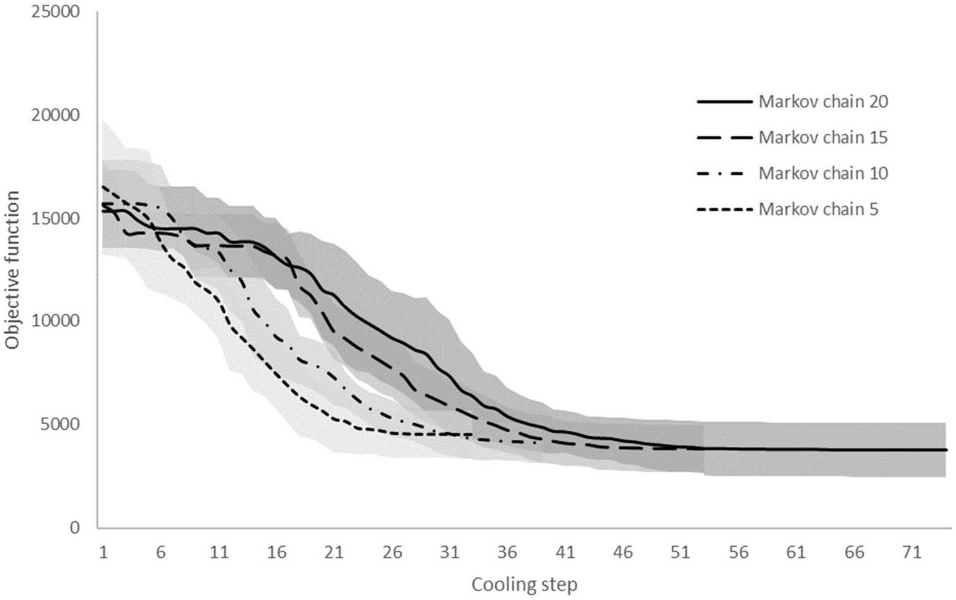

Figure 1. Objective function versus cooling step using 49 parallel groups.

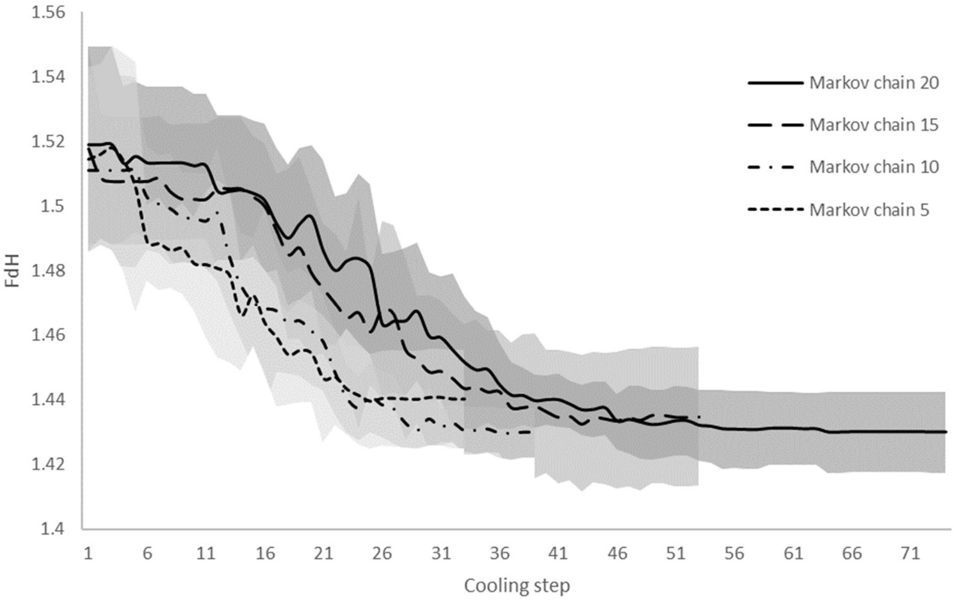

Figure 2. FdH versus cooling step using 49 parallel groups.

In addition to the objective function the FdH value is presented in Figure 2, which is the constraint with the largest weight. The shaded area represents the standard deviation. These two figures show that PSA converges faster when Markov chain is shorter. There are also less number of cooling steps needed for the code to converge for the shorter Markov chains using the same convergence criteria. However, as expected the standard deviation increases with the decrease of the Markov chain size although in some cases smaller standard deviation is observed for smaller Markov chains. This signifies that the solution is not very robust in this range of solutions per cooling step which is less than a thousand and more cases need to be run per cooling step in order to observe consistent behavior of the data.

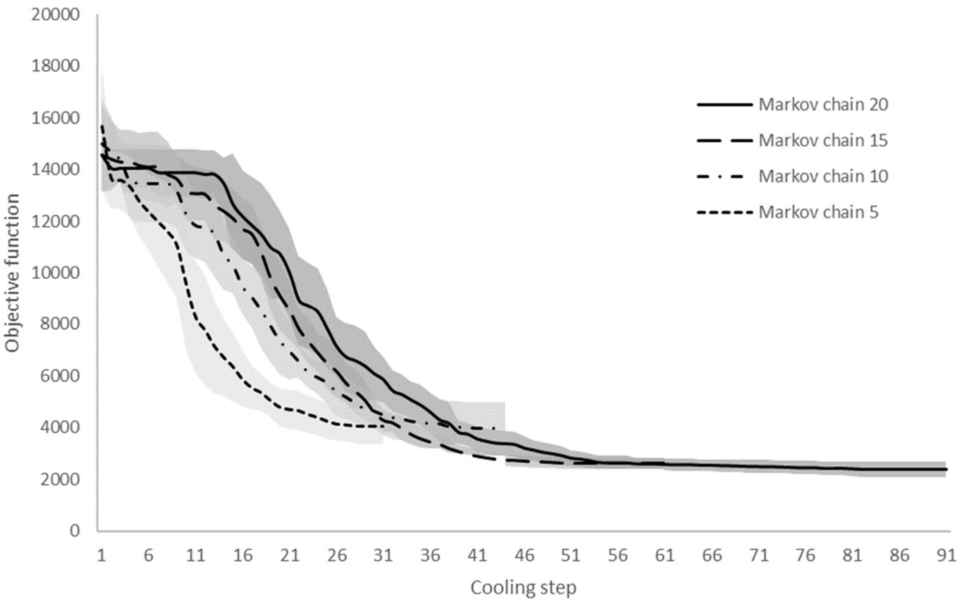

Figure 3. Objective function versus cooling step using 105 parallel groups.

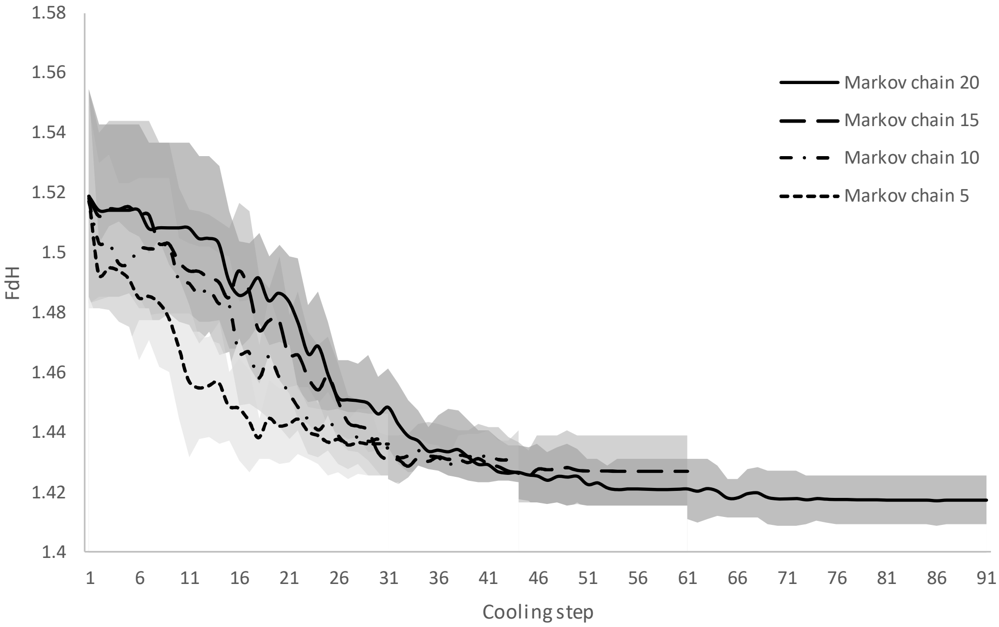

Figure 4. FdH versus cooling step using 105 parallel groups.

Figures 3 and 4 present the same parameters – the Objective function and FdH but obtained with 105 parallel groups. The results are now very consistent, and it can be observed that the standard deviation decreases with the increase of the Markov chain length. These results again show that PSA converges faster when the Markov chain is shorter. It was previously shown that the increase of number of parallel groups also leads to faster convergence when the total number of solutions per cooling step is preserved [6]. In this calculation, comparing figures 1 and 3, we observe that faster convergence is attained even when the Markov chain length is preserved which leads to a larger number of solutions per cooling step.

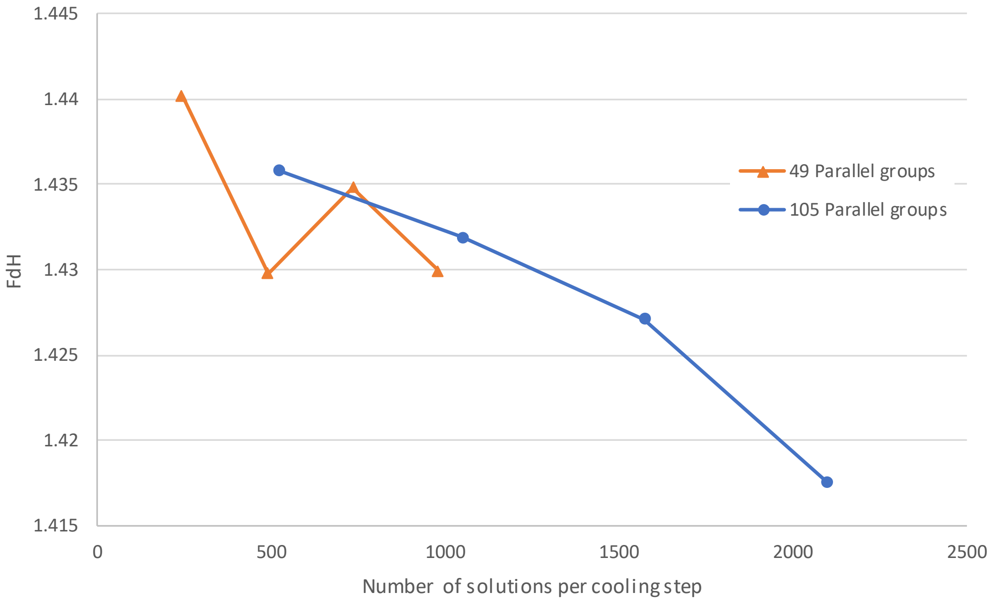

Figure 5. FdH versus total number of solutions per cooling step.

It should be noted, that the solution becomes better with the increase of the Markov chain which is expected. For example, the mean FdH for the case with Markov chain of size 20 was 1.418 while in the case of Markov chain of size 5 the mean FdH is 1.436. The same is valid for the Objective function, in the first case the Objective function decreased to 2383 while in the second case with Markov chain size of 5 it decreased to only 4051. The Objective function signifies how well the constraints were met so the lower the value the better the solution. In this calculation, the Objective function did not decrease down to 0 as unachievable FdH constraint was given to make the code run until the convergence criteria on accepted solutions is met.

Figure 5 presents FdH as a function of the number of solutions per cooling step which again shows the decrease of FdH with the increase of the number of solutions per cooling step. These results as well as observing the standard deviations in the previous figures show that about fifteen hundred or more solutions per cooling step are needed to obtain a robust solution. This will result in significant running time for the core simulator NEXUS/ANC9 which could not be acceptable. However, the running time could be significantly reduced by relaxing the convergence criteria and increasing the number of parallel groups. It can be observed that the last several cooling steps do not improve the solution significantly so relaxing the convergence criteria will reduce the running time and will not impact significantly the solution.

3. RESULTS

The PSA method with NEXUS/ANC9 was tested on several real-world plants and cycles, including 2-loop, 3-loop and 4-loop Westinghouse cores. Different fuel assembly lattices with IFBA, WABA and Gadolinium burnable absorbers were also exercised in these cases. Selection of the results are presented in Table I.

Table I. PSA/ANC9 Results.

Plant	**Number of assemblies**	**Actual LP**			**PSA/ANC9 Optimized LP**		
		FdH	EOC ppm	Max HFP ppm	FdH	EOC ppm	Max HFP ppm
A	157, 3-loop	1.463	8.0	1367.0	1.406	13.0	1374.0
B	121, 2-loop	1.492	10.0	1405.1	1.495	11.4	1412.6
C	157, 3-loop	1.456	-23.0	1327.1	1.437	-27.0	1377.6
D	157, 3-loop	1.439	8.5	1203.2	1.436	8.3	1232.4
E	193, 4-loop	1.433	11.4	1233.1	1.422	15.8	1357.4

Results in the table above show that NEXUS/ANC9 with the PSA method generated Loading Patterns that are comparable or better than the manually optimized ones. It should be noted that most of the results above were generated with Markov chain size of 5 and 105 parallel groups for the sake of running time. These results could be further improved if larger Markov chain is used or more parallel processes are utilized. Some of the results in the table show significant improvement, especially in plant A where not only the FdH decreased from 1.463 to 1.406 but also End of Cycle (EOC) boron increased from 8 ppm to 13 ppm.

4. CONCLUSIONS

The PSA method was recently adopted in NEXUS/ANC9 in a prototype version. The performance and robustness of the prototype was evaluated, and its results were assessed in comparison to manually optimized ones. The code showed robust performance once more than thousand solutions were used per cooling step. This results in long running times which can be overcome with using more parallel processes and/or appropriate convergence criteria. The code produced comparable or better results than the manually optimized ones even though relatively low number of solutions per cooling steps were used.

The future work involves implementation of Machine Learning models to partially substitute the expensive core evaluation and to perform online loading pattern safety evaluation.

ACKNOWLEDGMENTS

The authors would like to thank the colleagues in the Core Engineering and Methods Development departments for their support in the course of this work.

REFERENCES

1. L. Mayhue, et al, "Qualification of NEXUS/ANC Nuclear Design System for PWR Analysis," *Proceedings of PHYSOR 2008*, Interlaken, Switzerland, Sept. 14-19, 2008.
2. ALPS User Manual, Westinghouse Electric Company, (2009)
3. S. Si and F. D. Popa, "The Pearls Loading Pattern Search Tool," *Proceedings of M&C Conference*, Palais des Papes, Avignon, France, September 12-15, 2005.
4. Chu, K., Deng, Y. and Reinitz, J. "Parallel Simulated Annealing by Mixing of States," *J. Comp. Phys.*, **148**, 646-662 (1999).
5. David J. Kropaczek, "Concept for Multi-cycle Nuclear Fuel Optimization Based On Parallel Simulated Annealing With Mixing of States," *International Conference on the Physics of Reactors*, Interlaken, Switzerland, Sept. 14-19, 2008.
6. David J. Kropaczek and Ryan Walden, "Constraint Annealing Method for Solution of Multiconstrained Nuclear Fuel Cycle Optimization Problems," J. Nuclear Science and Engineering, 193, 506-522 (2019).