

Concept for Multi-cycle Nuclear Fuel Optimization Based On Parallel Simulated Annealing With Mixing of States

David J. Kropaczek *

Studsvik Scandpower, Inc., Wilmington, NC, USA

Abstract

A new concept for performing nuclear fuel optimization over a multi-cycle planning horizon is presented. The method provides for an implicit coupling between traditionally separate in-core and out-of-core fuel management decisions including determination of: fresh fuel batch size, enrichment and bundle design; exposed fuel reuse; and core loading pattern. The algorithm uses simulated annealing optimization, modified with a technique called mixing of states that allows for deployment in a scalable parallel environment. Analysis of algorithm performance for a transition cycle design (i.e. a PWR 6 month cycle length extension) demonstrates the feasibility of the approach as a production tool for fuel procurement and multi-cycle core design.

1. Introduction

Fuel management for Light Water Reactors (LWRs) has historically been divided into out-of-core and in-core fuel management tasks. Out-of-core fuel management deals with the long range planning horizon for uranium procurement and involves the consideration of many years of plant operation over several energy cycles. In-core fuel management addresses the constraints of the fuel design, including enrichment and burnable poison distributions, placements of the fresh and reload bundles within the core loading pattern, and the operational strategy (i.e. control rods, flow, moderator temperature), that maximize fuel utilization while meeting single cycle energy requirements. Out-of-core analyses are typically concerned with mitigating the risks associated with long range planning projections, such as variability in the wholesale energy market or uranium cost,

while in-core analyses are concerned with assuring the compliance of the core design with plant licensing and operating technical specifications. For today's LWR cores, with reload batch fractions in the range of 20%-50%, the strong coupling between out-of-core and in-core analyses is evident when one considers that fuel will reside in the core between 2 and 5 cycles (i.e. 4 to 6 years) of operation.

A primary reason for the historical divide between out-of-core and in-core fuel management can be attributed to the large computational burden required to perform the fuel cycle analyses. In-core fuel management requires the use of licensing grade core simulators (Rempe, et al. 1989) with advanced neutronic features to model the complex geometries and materials of today's fuel designs (e.g. part length fuel rods, mixed oxide fuel, etc.). Computational resources are challenged, however, when such simulators are combined with automation tools to facilitate their use in design. Nevertheless,

* Corresponding author, dave.kropaczek@studsvik.com
Tel: +1 (910) 509 2059; Fax: +1 (910) 509 7436.

great strides have been made in the area of single cycle optimization for both Pressurized Water Reactors (PWRs) (Kropaczek and Turinsky, 1991; Stevens, et al. 1995; Parks, 1996) and Boiling Water Reactors (BWRs) (Karve and Turinsky, 2000; Kropaczek and Russell, 2003) over the last decade.

Despite progress, a criticism of single cycle optimization tools that has limited their potential has been the lack of coupling between successive refueling cycles. As is well-known, use of single cycle optimization results applied successively across cycles, without proper construction of the objective function, can lead to divergent multi-cycle behavior such as batch size or enrichment oscillations between cycles (Kropaczek, et al. 1993). Out-of-core analyses have therefore traditionally focused on the use of simple zero-dimensional models and, more recently, on the use of fast running three-dimensional GUI-enabled simulators that facilitate rapid manual development of a multi-cycle loading strategy that avoids divergent behavior. Such models have the same accuracy as the licensing simulator (Stevens and Rempe, 2003) and provide improved consistency with the in-core analysis.

Efforts in the nuclear industry to address the coupled out-of-core and in-core fuel management problem for LWRs as a constrained optimization problem have been quite limited. The recent enhancements to the OCEON code (Anderson, et al., 2007a, 2007b), an out-of-core optimization code for determining the fresh batch cycling schemes (i.e. enrichments, batch sizes) over a multi-cycle planning horizon, is recognition of the evolving need. This need can be traced to technological improvements that have allowed for increases in capacity factor, longer cycle lengths, extended power uprate (EPU), and advanced fuel designs. The result is the aggressive challenging of formerly benign design constraint limits, such as those on peak pellet exposure, and the introduction of new core design limits, such as those on steaming rate in PWRs and channel distortion in BWRs. To this end, a concept for multi-cycle fuel optimization was developed.

2. Algorithm for Multi-Cycle Optimization

The first step in developing a multi-cycle optimization algorithm is to recast the problem as

one that considers the actual coupling between out-of-core and in-core constraints.

2.1. Problem Definition

Each fuel cycle has associated with it an energy production plan and constraints that determines the exposed fuel characteristics at the completion of the cycle. The energy production plan encompasses all items impacting power output including: cycle length, power level requirements, operational strategy (control rods, core flow), and coastdown. Constraints include all thermal, reactivity, and fuel performance limits established to protect fuel integrity.

A cycle j is defined by an available fuel inventory \mathbf{I}_j which exists at beginning of cycle, derived from the exposed fuel bundles existing within the core as well as the discharge pool following the completion of the previous cycle. A loading instruction operator \mathbf{O}_j may be defined that provides a mapping of the fuel inventory of \mathbf{I}_j to the core loading pattern, \mathbf{LP}_j . The operator \mathbf{O}_j determines those exposed bundles and previously discharged bundles that will be reinserted into the core. In addition, \mathbf{O}_j may be extended to provide a mapping of available fresh fuel designs \mathbf{F}_j to the core loading pattern \mathbf{LP}_j . \mathbf{F}_j defines the permissible set of fuel bundle mechanical designs as well as enrichment and burnable poison distributions (radial and axial) established by manufacturing considerations for cycle j . Thus, given \mathbf{I}_j , \mathbf{F}_j and the operator \mathbf{O}_j , the loading pattern \mathbf{LP}_j for cycle j is uniquely determined. It is noted that within the context of single cycle optimization, \mathbf{LP}_j comprises the entire set of decision variables for a single cycle j .

For a multi-cycle analysis, the characteristics of the available fuel inventory \mathbf{I}_j in any given cycle j , which includes those bundles in the discharge pool and the core, is a direct function of the previous cycle core loading pattern, \mathbf{LP}_{j-1} , and available fuel inventory, \mathbf{I}_{j-1} . Therefore, for any cycle $j+n$ in the planning horizon, which begins with cycle j , the following functional relationship exists for the fuel inventory available for loading (see Eq. 1):

$$\mathbf{I}_{j+n} = f(\mathbf{LP}_{j+n-1}, \mathbf{LP}_{j+n-2}, \dots, \mathbf{LP}_j, \mathbf{I}_j) \quad (1)$$

The multi-cycle optimization problem may therefore be restated as follows. Given a multi-cycle energy plan and constraints for cycles j through $j+n$,

an available fuel inventory I_j , and set of available fresh fuel designs $\{F_j, F_{j+1}, \dots, F_{j+n}\}$, determine the set of operators $\{O_j, O_{j+1}, \dots, O_{j+n}\}$ that result in the optimal set of loading patterns $\{LP_j, LP_{j+1}, \dots, LP_{j+n}\}$ that minimizes fuel cycle cost over the planning horizon.

2.2. Simulated Annealing Optimization

Determination of the set of operators $\{O_j, O_{j+1}, \dots, O_{j+n}\}$ belongs to the class of large-scale combinatorial optimization problems that are characterized by discreteness, nonconvexity, and nonlinearity. In recent years, solutions to such problems have focused on stochastic methods such as Simulated Annealing (SA) (Stevens and Rempe, 1995) and Genetic Algorithms (GAs) (Parks, 1996), which most likely can be attributed to the advances in computational processing speed.

Simulated annealing is based on the well-known Metropolis algorithm (Metropolis, et al., 1953), a rejection sampling algorithm belonging to a class of Markov chain Monte Carlo methods developed for the purpose of sampling probability distributions. The Metropolis algorithm is specific to the Boltzmann probability distribution. A Markov chain is defined as a sequence of samples obtained from a probability distribution, in this case the Boltzmann distribution at a given system temperature. Markov chains are generated stochastically and have the property that given a current state, future states are independent of all past states. The state of the Markov chain after a sufficient number of samples is considered to be in equilibrium and representative of the sampled Boltzmann distribution.

SA optimization (Kirkpatrick, et al. 1983; Aarts and Van Laarhoven, 1985) generalizes the Metropolis algorithm by seeking to achieve a system state corresponding to one of minimum energy. This is performed by simulating the slow cooling of the system from high to low temperature according to a cooling schedule with equilibrium maintained at each successive temperature value. Within SA, the system energy is merely an analogy for a mathematical cost function, C , which captures the objective function and constraints of the optimization problem.

The SA process for generating Markov chains proceeds as follows. A current system state, C_{old} , is changed to a new state, C_{new} , via a system perturbation that may involve changes to one or

several independent variables. For $C_{new} < C_{old}$, the perturbation or "move" is always accepted. For $C_{new} > C_{old}$ the move is accepted with probability proportional to $\exp\{-(C_{new}-C_{old})/T\}$, where T is the system temperature. For moves not accepted the system remains in the old state. Note that a sufficiently high temperature will accept all moves (i.e. both increases and decreases to C) while a low temperature will tend towards accepting only those moves that decrease C .

The goal of SA is to determine the set of decision variables defining the system state that minimizes the cost function, which will occur at the minimum system temperature. It is clear, however, that SA algorithm performance is directly tied to the cooling schedule employed. Too slow a cooling rate will result in prohibitive computational time while too high a cooling rate will result in a lack of representative states (analogous to "quenching" the system).

2.3. Cooling Schedule

While global proofs of convergence exist for logarithmic cooling schedules, practical constraints on computation time require cooling schedules that are both computationally efficient and robust with respect to identifying the highest quality solutions. The Lam cooling schedule (Lam and Delosme, 1988a, 1988b) is an adaptive exponential schedule that is based on optimizing the rate at which the temperature can be decreased subject to the constraint of maintaining quasi-equilibrium. As described by Chu, et al. (1999), it differs from other adaptive schedules in that it explicitly takes into account the move generation strategy and provides a mechanism for its control.

The Lam schedule is given by Eqs. 2 and 3 as:

$$s_{k+1} = s_k + \lambda \left(\frac{1}{\sigma(s_k)} \right) \left(\frac{1}{s_k^2 \sigma^2(s_k)} \right) G(\rho(s_k)) \quad (2)$$

$$G(\rho) = \left(\frac{4\rho(1-\rho)^2}{(2-\rho)^2} \right) \quad (3)$$

where,

$1/s_k = T_k$ = temperature at k_{th} evaluation of C

$\sigma(s_k)$ = standard deviation of C at s_k

$\rho(s_k)$ = move acceptance ratio at s_k

λ = quality factor, ($1.0 < \lambda < 2.0$)

Within Eq. 2, the quality factor provides a global adjustment to the cooling rate where decreases to λ slow the cooling rate, and vice versa. The term $(1/\sigma(s))$ is a measure of the distance of the system from quasi-equilibrium. Larger values of $\sigma(s)$ will likewise decrease the cooling rate. The term $G(\rho)$, shown in Fig. 1, is a measure of the variance of the average change in C during a move. It affects Eq. 2 by providing an accelerated cooling for values of ρ in the range of 0.3 to 0.6 with decelerated cooling as ρ decreases from 0.3 to 0.0 and as ρ increases from 0.6 to 1.0.

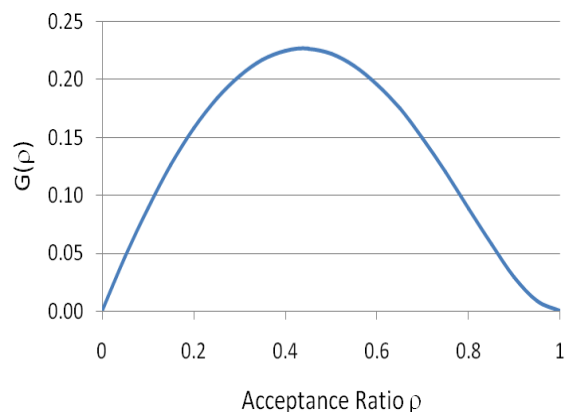


Figure 1. Function $G(\rho)$

The initial temperature is estimated by evaluating a starting Markov chain where all solutions are accepted with probability equal to 1.0, and then calculating a standard deviation of the cost function for all solutions sampled. The initial temperature, T_0 , is then determined as follows in Eq. 4:

$$T_0 = \alpha \sigma^* \quad (4)$$

where,

σ^* = standard deviation of C during initialization

α = initialization parameter, $(1.0 < \alpha < 2.0)$

2.4. Move Generation Strategy

The sampling strategy for creating multi-cycle loading and depletion results is based on the generation of a set of perturbed operators $\{\mathbf{O}_j, \mathbf{O}_{j+1}, \dots, \mathbf{O}_{j+n}\}$ given a current best solution defined by the set of operators $\{\mathbf{O}_j, \mathbf{O}_{j+1}, \dots, \mathbf{O}_{j+n}\}$. Each cycle j loading instruction operator \mathbf{O}_j can be broken down by the subset of loading operations that may be performed on the available inventory \mathbf{I}_j and fresh designs \mathbf{F}_j . These loading operations include: 1) symmetric pairing of core locations (e.g. mirror, rotational); 2) ranked ordering of core locations with respect to \mathbf{I}_j ; 3) orientations of core locations with \mathbf{I}_j (e.g. cross quadrant/octant shuffles); and 4) \mathbf{F}_j assignment by core locations. Perturbing the operator \mathbf{O}_j therefore involves a change in one or several subsets of loading operations. For example, modifying \mathbf{O}_j to reflect a change in the ranked ordering of core locations with respect to \mathbf{I}_j would correspond to one or several shuffles within the resultant loading pattern, \mathbf{LP}_j . Converting a core location that was rank ordered by \mathbf{I}_j to one that was subject to assignment by \mathbf{F}_j would effectively change the fresh fuel batch size. Through changes to the operator \mathbf{O}_j it is possible to span the range of potential loading patterns \mathbf{LP}_j , including changes in batch size, enrichment, and exposed fuel inventory.

The move generation strategy is critical to creating the Markov chains within the SA algorithm. Each solution generated during the process depends only on the current best multi-cycle solution. As encoded within the current algorithm, a random sampling process is employed to select the cycle j and type of perturbation to be applied to operator \mathbf{O}_j in the creation of new solutions. For the current application of PWR multi-cycle optimization, six categories of perturbations to \mathbf{O}_j are encoded: 1) batch (and sub-batch) size; 2) enrichment (and enrichment split among sub-batches); 3) fresh loading pattern; 4) fresh bundle design (including burnable poison); 5) exposed fuel selection and loading pattern; and 6) exposed fuel gradient orientation.

2.5. Parallel Simulated Annealing by Mixing of States

The large computational demands of multi-cycle optimization makes parallelism a necessity, particularly when one considers that a licensing

grade simulator depletion calculation must be performed over successive cycles in order to evaluate the cost function. The serial nature of SA makes parallelism a challenge although algorithms such as Temperature Parallel Simulated Annealing (TPSA) (Yamamoto and Hashimoto, 2000) have been successfully applied to the single cycle loading pattern optimization problem. Within TPSA, independent annealing lines are followed with an exchange of solutions that occurs periodically between lines.

An alternative approach to TPSA, is parallel SA based on mixing of states (Chu, et al., 1999). In this method, a single cooling schedule is maintained which, for the current application, is the Lam cooling schedule described by Eq. 2. Parallel Markov chains of length L are initiated at the start of each k_{th} temperature evaluation by sampling from a population of n current best solutions. The population of current best solutions is comprised of the union of solutions obtained from each Markov chain of the previous $(k-1)_{th}$ temperature evaluation. The sampling probability, p_i , for each solution i in the population is related to its cost function C_i as follows in Eq. 5:

$$p_i = \frac{\exp(-C_i / T_k)}{\sum_{j=1}^n \exp(-C_j / T_k)} \quad (5)$$

For the Lam cooling schedule, the standard deviation of the cost function, σ , is based on the pooled statistics of the parallel Markov chains at the k_{th} temperature evaluation. Similarly, the acceptance ratio, ρ , is likewise calculated based on pooled statistics. In this manner, information from the independent Markov chains is exchanged at the completion of each temperature evaluation with the pooled statistics forming the basis of the subsequent temperature adjustment. Convergence of the algorithm is detected when ρ falls below a certain threshold (~10% for the current application).

2.6. Cost Function

The cost function, C , for multi-cycle optimization is defined as a cumulative cost function as follows (see Eq. 6):

$$C = \sum_{j=1}^n x_j c_j \quad (6)$$

where,

- c_j = cost function for a given cycle j
- n = number of cycles in the planning horizon
- x_j = importance weight applicable to cycle j

A traditional penalty function approach (Kropaczek and Turinsky, 1991) is used in defining c_j as equal to a fuel cost component plus an additional penalty function that captures the constraint violations of the loading pattern, \mathbf{LP}_j , for the cycle. This may be expressed as follows in Eq. 7:

$$c_j = FCC_j + \sum_{l=1}^m w_l P_{j,l} \quad (7)$$

where,

- FCC_j = fuel cycle cost component for cycle j
- $P_{j,l}$ = violation measure for constraint l in cycle j
- w_l = importance weight applicable to constraint l

The fuel cycle cost component, FCC_j , is based on the total cost of the fuel batch including: uranium ore, conversion, separative work, fabrication, back-end costs and carrying charges. It is important to note that FCC_j , while specific to the fresh batch loaded in cycle j , will nevertheless depend on the future cycles (i.e. the history of how the fuel exposure is accumulated prior to final discharge). Active constraints, $P_{j,l}$, considered include enthalpy rise hot channel peaking factor ($F_{\Delta H}$), target end of cycle boron, peak discharge burnup, and moderator temperature coefficient (MTC).

Note that the cost function formulation of Eq. 6 allows, through preferential weighting, additional emphasis to be placed on the earliest cycle(s) under design in the planning horizon with lesser emphasis on cycles farther out in the future. This is an important consideration from a practical design viewpoint. For example, satisfaction of all design constraints, $P_{j,l}$, several cycles into the future might be of lesser importance in contrast to the upcoming cycle under design. This may arise due to future cycle uncertainties associated with such parameters as energy plan, operations, or outage scheduling.

3. Results

The multi-cycle optimization concept was applied to a 3-loop Westinghouse PWR core, having completed Cycle 2 of a 12 month cycle of operation. The problem created to study the characteristics of the algorithm presented is a transition design involving a cycle length change from 12 to 18 months in Cycles 3, 4 and 5, with an accompanying change in bundle design, from Wet Annular Burnable Absorber (WABA) to Gadolinium (Gad). Up to 24 unique Gad loading and Gad rod configurations were considered as part of the fresh bundle design palette. Up to two fresh batches with a minimum U^{235} enrichment split of 0.4 w/o are also allowed in each of the 3 cycles. All bundles within the Cycle 2 core as well as the spent fuel pool, which included the discharge bundles from Cycle 1, were considered candidates for re-use.

Early life, short cycle cores are particularly challenging since they tend to be designed at lower enrichments with the potential for large numbers of bundles to be prematurely discharged to the fuel pool. An assessment of algorithm performance is presented for this problem.

3.1. Parallel SA Performance

Figs. 2 and 3 present the parallel SA algorithm performance on $n=4$ and $n=16$ processors, respectively. Presented within the figures are the following: 1) temperature, 2) average cost, 3) standard deviation of cost, and 4) acceptance ratio, as a function of evaluated number of multi-cycle solution histories. Each history represents a solution which is comprised of 3 successive cycles of SIMULATE-3 (Rempe, et al., 1999) evaluated depletion results. All results, with the exception of the acceptance ratio, have been normalized to a value of 1.0 for purposes of comparison. It is noted that for consistency, the same normalization factors are used in both graphs. In the current example, a cost function value of zero satisfies all active constraints while satisfying a minimum threshold for total fuel cycle cost.

In general, similar behavior is exhibited for the results of parallel scaling from $n=4$ to $n=16$. The acceptance ratio shows general linear behavior over time while temperature follows the expected exponential cooling behavior. Evident, however, is a compression in the evaluated histories for $n=16$ which coincides with a slightly more rapid cooling

rate. For the current problem, final cost function values are indistinguishable although it is noted that the set of loading patterns obtained $\{\mathbf{LP}_j, \mathbf{LP}_{j+1}, \dots, \mathbf{LP}_{j+n}\}$ are quite different, particularly in the placement of fresh fuel and bundle design. This is not surprising considering the existence of multiple minima that are characteristic of the core loading problem. More interesting is the apparent improvement in optimization performance with scaling as n increases. One possible explanation is that the independent Markov chains initiated during each temperature evaluation are providing an efficiency improvement via a more thorough search of the local neighborhood.

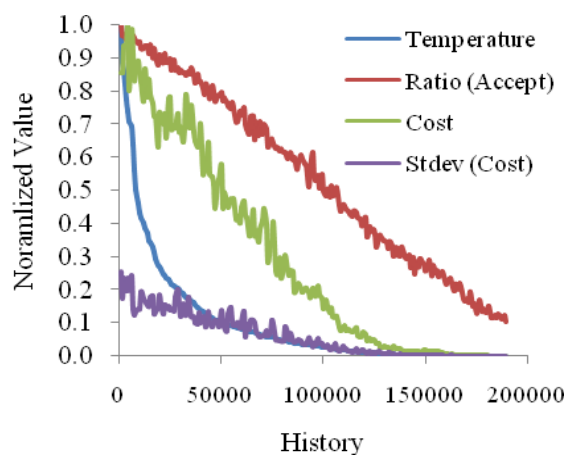


Figure 2. SA Performance ($n=4$ Processors)

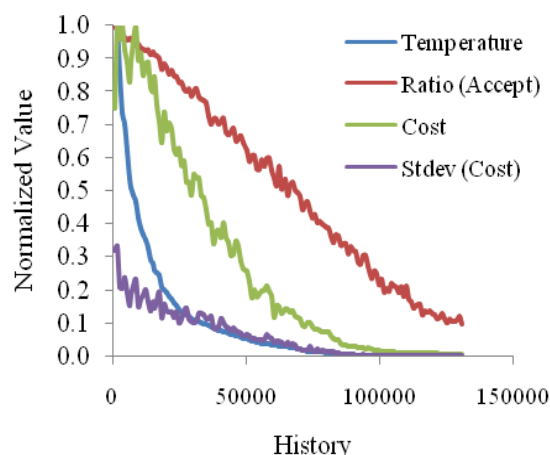


Figure 3. SA Performance ($n=16$ Processors)

3.2. Movement Generation Assessment

Fig. 4 examines the acceptance ratio as a function of temperature step for each of the individual operator perturbations applicable to the current problem. These include perturbations to: fresh batch size, enrichment, fresh fuel loading pattern (LP), bundle design, exposed fuel loading pattern, and exposed fuel gradient orientation. The sum of all operator perturbation acceptance ratios at the k_{th} temperature step is equal to the move acceptance ratio, $\rho(s_k)$, of Eq. 2.

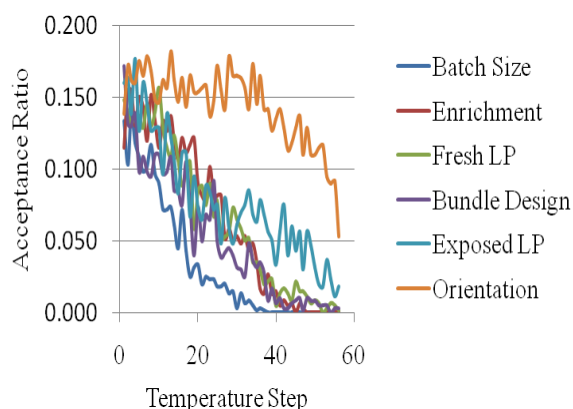


Figure 4. Operator Perturbation Acceptance Ratios

An analysis of the operator perturbation acceptance ratios as a function of temperature step provides insight into the behaviour of the optimization algorithm as temperature evolves over time. As shown in Fig. 4, operator perturbations in enrichment, fresh LP, and fresh bundle design show similar behaviour, progressing from ~ 0.15 to < 0.01 at the point of convergence. The batch size operator perturbation demonstrates a much more dramatic decreasing slope, progressing from ~ 0.15 to 0.0 at temperature step 40 (a point lying at $2/3$ of algorithm completion). The exposed fuel LP acceptance ratio shows a less steep slope and, in fact, lessens more beyond temperature step 30. The exposed fuel gradient orientation acceptance ratio displays little structure until very late in the algorithm, beyond temperature step 40, where a negative slope becomes apparent.

Conclusions that can be drawn are as follows. Batch size changes, which are the most disruptive (i.e. have the greatest impact) in terms of fuel cost

and design constraints, are significant in the earlier stages of the algorithm. Conversely, the latter stages of the algorithm tend to focus on the exposed fuel LP and orientation perturbations, which are the least significant in terms of fuel cost, although they can greatly impact the design constraints. The remaining operator perturbations: enrichment; fresh LP; and bundle design; remain significant throughout the optimization, although with decreasing influence as convergence is reached. This result is not surprising since these operator perturbations address combined core characteristics that impact both fuel cost as well as design constraints.

3.3. CPU Considerations

For the problem analyzed, a SIMULATE-3 multi-cycle depletion model was created that exhibited an average solution evolution time of 12.3 sec. (3 cycles). The model was based on 16 axial mesh, 1 node per assembly, and 12 depletion steps per cycle. Pin power reconstruction was active. An enrichment interpolation cross section library was generated based on 0.2 w/o increment in the range of 2.0 w/o to 5.0 w/o U^{235} for each unique Gad configuration. The computational environment is a distributed network of HP 8400 Quad core Xeon processors (3.0 GHz, 4 GB RAM) running Windows x64 with MPICH2 (Gropp, et al., 2007).

Referring back to Fig. 3, a wall clock time of 32 hrs for the evaluation of 150,000 multi-cycle solutions is required on a network with $n=16$ CPUs. Scaling to $n=80$ would reduce the wall clock time to 6.4 hrs, a number well within reach of daily production requirements.

4. Summary

A concept for performing multi-cycle nuclear fuel optimization has been presented and shown to be a viable approach from the perspective of algorithm performance, particularly as it relates to scalability in a parallel computing environment. The results show all decision variables of interest to fuel procurement and core design may be addressed implicitly by the algorithm presented.

5. References

- Aarts, E. and Van Laarhoven, P., "Statistical Cooling Algorithm: A General Approach to Combinatorial Optimization Problems," *Philips J. Res.* **40**, 1985, 193.
- Anderson, K.A., Turinsky, P.J. and Keller, P.M., "Improvement of OCEON-P Optimization Capabilities," *Trans. Am. Nucl. Soc.*, **96**, 2007, 201.
- Anderson, K.A., et al. "OCEON-P Linkage with SIMULATE3," *Trans. Am. Nucl. Soc.*, **96**, 2007, 203.
- Chu, K., Deng, Y. and Reinitz, J. "Parallel Simulated Annealing by Mixing of States," *J. Comp. Phys.*, **148**, 1999, 646-662.
- Gropp, W., et al. "MPICH2 User's Guide," Version 1.0.6, Mathematics and Computer Science Division, Argonne National Laboratory, Sept. 14, 2007.
- Karve, A.A. and Turinsky, P.J., "FORMOSA-B: A Boiling Water Reactor In-Core Fuel Management Optimization Package II," *Nucl. Technol.*, **131**, 2000, 48.
- Kirkpatrick, S., Gelatt, C.D. and M. P. Vecchi M.P., "Optimization by Simulated Annealing," *Science* **220**, 1983, 671.
- Kropaczek, D.J. and Turinsky, P.J., "In-Core Fuel Management Optimization for Pressurized Water Reactors Utilizing Simulated Annealing," *Nucl. Technol.*, **95**, 1991, 9.
- Kropaczek, D.J. and Russell, W.E., "Method for Optimization of BWR Fuel Management and Plant Operations," *Top. Mtg. Adv. Nucl. Fuel Mgt. III*, Hilton Head, 2003, II-117.
- Kropaczek, D.J., McElroy, J., and Turinsky, P.J., "Validity of Single Cycle Objective Functions for Multi-Cycle Reload Design Optimization," *Trans. Am. Nucl. Soc.*, **69** 1993.
- Lam, J. and Delosme, J.-M., *An Efficient Simulated Annealing Schedule: Derivation*, Technical Report 8816, Electrical Engineering Department, Yale, New Haven, CT, September 1988.
- Lam, J. and Delosme, J.-M., *An Efficient Simulated Annealing Schedule: Implementation and Evaluation*, Technical Report 8817, Electrical Engineering Department, Yale, New Haven, CT, September 1988.
- Metropolis, N., et al., "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.*, **21**, 1953, 1087.
- Parks, G.T., "Multiobjective Pressurized Water Reactor Reload Core Design by Non-dominated Genetic Algorithm Search," *Nucl. Sci. Eng.*, **124**, 1996, 178.
- Rempe, K.R., et al., "SIMULATE-3 Pin Power Reconstruction: Methodology and Benchmarking," *Nucl. Sci. Eng.*, **103**, 1989, 334.
- Stevens, J.G., Smith, K.S., and Rempe, K.R., "Optimization of Pressurized Water Reactor Shuffling by Simulated Annealing with Heuristics," *Nucl. Sci. Eng.*, **121**, 1995, 67.
- Stevens, J.G. and Rempe, K.R., "Equilibrium Cycle Analysis with XIMAGE/SIMAN," *Top. Mtg. Adv. Nucl. Fuel Mgt. III*, Hilton Head, 2003, II-43.
- Yamamoto A. and Hashimoto, H., "Application of Temperature Parallel Simulated Annealing to Loading Pattern Optimizations of Pressurized Water Reactors," *Nucl. Sci. Eng.*, **136**, 2000, 247.