

## DEVELOPMENT AND APPLICATION OF A PENALTY-FREE METHOD FOR ADDRESSING CONSTRAINTS WITHIN FUEL CYCLE OPTIMIZATION PROBLEMS

David J. Kropaczek

*North Carolina State University, P.O. Box 7909, Raleigh NC 27695, kropaczek@ncsu.edu*

**Abstract** - A method is developed and demonstrated for addressing constraints within the context of fuel cycle optimization problems. The penalty-free method, based on constraint annealing, eliminates the use of traditional constraint penalty factors by treating each constraint as separate and concurrently solved minimization problems within a global optimization search framework. Results are demonstrated for a realistic core loading pattern design problem.

### I. INTRODUCTION

Fuel cycle design and optimization belongs to a class of NP-hard combinatorial optimization problems characterized by discrete decision variables, non-linear formulations for the objective function and constraints, and use of computationally intensive, multi-physics models for evaluation of the parameters of interest. For LWRs, the fuel cycle design problem may be generally defined as the determination of the fresh fuel design(s), exposed fuel carryover, and fresh and exposed fuel placements within a loading pattern that minimizes fuel costs while meeting constraints on cycle energy production, operational margins and safety limits.

Several approaches have proved capable for the solution of such problems, the most successful being methods based on simulated annealing [1] and evolutionary algorithms [2]. A primary implementation challenge that remains is the treatment of constraints within the optimization algorithm. Table I displays a representative, although not comprehensive, list of constraints of interest to PWRs. It is noted that numerous constraint formulations are possible and are unique to each reactor and fuel cycle under design. The optimization method must assure convergence of the search to the global optimum for the defined objective function while assuring the feasibility of the solutions obtained. For many problems, the existence of local minima and the potential for entrapment necessitates the use of techniques that allow for violation of constraints during the search. This is most readily accomplished through the use of penalty functions applied to the constraints as part of the objective function formulation. Penalty function approaches that have been employed for solution of large-scale combinatorial problems include static, dynamic, and adaptive methods [2]. Variations of these methods have been demonstrated for simulated annealing solution of the multi-cycle core loading optimization problem [3, 4].

The penalty function approach takes the form of a modified objective function as follows:

$$\bar{f} = f + \sum_{i=1}^I \gamma_i C_i \quad (1)$$

Within Eq. 1,  $f$  represents the unmodified objective function,  $\gamma_i$  is the penalty weight specific to constraint  $C_i$ , and  $I$  is the number of active constraints for the problem. It is noted that constraints are defined to have a lower zero bound representing non-violation of constraint limits while the objective function is generally treated as unbounded for purposes of the optimization. It is also clear that constraints may become objective functions through elimination of the lower zero bound and vice-versa. The penalty weights may be user input constants or may be adjusted periodically during the course of the optimization. As shown in Table I, the significant number and possible combinations of core loading design constraints poses a tremendous challenge. In general, penalty function formulations are problem dependent with penalty weights often tailored to achieve a balance between computational run time performance and quality of optimized solutions. An approach for treating penalty weights developed for solution of one design often lacks robustness when applied to a different reactor with a different set of constraints.

TABLE I. Representative list of PWR core loading design constraints

Category	Parameters
Reactivity	MTC, SDM, critical boron
Thermal Margins (unrodded/rodded)	FdH, FQ, assembly average power, quadrant power tilt, steaming rate (CIPS/CILC), AST
Fuel Exposure	Peak rod, region average
Vessel Fluence	Peripheral fuel rod power
Fuel Manufacturing	Enrichment & burnable poison distributions
Failed fuel mitigation and contingency	Peripheral fuel re-use (GTRF), step-out-bundles

### II. CONSTRAINT ANNEALING METHOD

Conceptually, the proposed penalty-free method [5] treats the objective function and constraints within the multi-constrained optimization problem as separate minimization problems to be solved for  $f$  and  $C_i$ . Of

primary importance is the manner in which the separate minimization problems are coupled and how it impacts solution evolution during the course of the optimization. An approach based on parallel simulated annealing with mixing of states is next developed, hereafter referred to as *constraint annealing*.

For purposes of clarity, and without loss of generality, the discussion that follows will focus on constraints only. Incorporation of one or several objective functions into the optimization method described is readily achieved by treating each objective function as an unbounded constraint, noting that both positive and negative values are now possible. The result is a constrained optimization problem (COP) that degenerates into a constraint satisfaction problem (CSP) when the objective functions are set equal to zero.

### 1. Adapted Simulated Annealing

Constraint annealing is based on the method of simulated annealing where random sampled, successive solutions are either accepted or rejected based on the Metropolis criterion:

$$P_{i,n}^l = \begin{cases} \exp(-(C_{i,n}^l - C_{i,n}^*)s_{i,n}) & \text{for } C_{i,n}^l > C_{i,n}^* \\ 1.0 & \text{otherwise} \end{cases} \quad (2)$$

where:

- $s_{i,n}$  = inverse temperature for constraint  $i$  at cooling step  $n$
- $P_{i,n}^l$  = acceptance probability for constraint  $i$  at  $s_{i,n}$ , Markov step  $l$
- $C_{i,n}^l$  = constraint  $i$  value at  $s_{i,n}$ , Markov step  $l$
- $C_{i,n}^*$  = current global accepted value for constraint  $i$  at cooling step  $n$

Within Eq. 2, each constraint maintains a unique cooling schedule to assess solutions generated within the Markov chain sequences. Global solution acceptance is based on satisfying the Metropolis criterion *for each constraint*. This equates to satisfying the bounding constraint (i.e. the lowest acceptance probability) for a given sampled solution. It is noted that a sampled solution may therefore satisfy a subset of non-bounding constraints (i.e. local acceptance) while being rejected as a global accepted solution. This motivates a second key aspect of constraint annealing and that is the use of local acceptance/rejection information as part of the annealing schedule.

The annealing schedule for each constraint is based on adjustment of the inverse temperature following completion of each Markov chain. For constraint annealing, the choice of a  $\lambda$ -schedule seeks to maintain a stationary probability distribution for each constraint  $i$  at cooling step  $n$ :

$$|\bar{X}(s_{i,n-1}) - \mu(s_{i,n})| \leq \lambda \sigma(s_{i,n}) \quad (3)$$

where:

- $\bar{X}(s_{i,n-1})$  = average value for constraint  $i$  at  $s_{i,n-1}$
- $\mu(s_{i,n})$  = true mean for constraint  $i$  at  $s_{i,n}$
- $\sigma(s_{i,n})$  = true standard deviation for constraint  $i$  at  $s_{i,n}$
- $\lambda$  = constant

For the proposed method, an adaptation of the Lam cooling schedule [6] is utilized for each constraint as follows:

$$s_{i,n+1} = s_{i,n} + \lambda \left( \frac{1}{\hat{\sigma}(s_{i,n})} \right) \left( \frac{1}{s_{i,n}^2 \hat{\sigma}^2(s_{i,n})} \right) G(\rho_n) \quad (4)$$

$$G(\rho) = \left( \frac{4\rho(1-\rho)^2}{(2-\rho)^2} \right) \quad (5)$$

where:

- $\hat{\sigma}(s_{i,n})$  = standard deviation estimate for constraint  $i$  at  $s_{i,n}$
- $\rho_n$  = global acceptance ratio at cooling step  $n$

The Lam cooling schedule of Eq. 4 and 5 satisfies the stationary condition of Eq. 3, with  $\lambda$  being a characteristic of the cooling schedule. A key aspect is that within constraint annealing, the global acceptance ratio and  $\lambda$  are common to each constraint annealing schedule while the standard deviation estimates and resultant inverse temperatures are calculated local to each constraint. Constraint annealing therefore combines knowledge of the individual constraint behavior with global solution acceptance as the means to inform the optimization search. It is noted that at a fixed temperature the average constraint value will converge to the true mean as the Markov chain length goes to infinity (i.e. a time-homogeneous Markov chain). Therefore, a sufficiently small  $\lambda$  can always be set that will establish quasi-equilibrium, albeit with a slower cooling rate and increased run-time.

### 2. Parallel Acceleration with Mixing of States

Parallel simulated annealing (PSA) allows for acceleration of the optimization by incorporating parallel Markov chains within the solution algorithm thus enabling solution on tens to hundreds of processors [7]. Within PSA concurrent Markov chains are initiated at the start of each temperature evaluation. Within constraint annealing, the sampling probability is calculated for each member of the population based on the *minimum* constraint probabilities for each solution and the inverse temperature as follows:

$$J_n^p = \frac{\min \{F_{1,n}^p, F_{2,n}^p, \dots, F_{l-1,n}^p, F_{l,n}^p\}}{\sum_{p=1}^N \min \{F_{1,n}^p, F_{2,n}^p, \dots, F_{l-1,n}^p, F_{l,n}^p\}} \quad (6)$$

$$F_{i,n}^p = \exp(-C_{i,n-1}^p s_{i,n}) \quad (7)$$

where:

$$J_n^p = \text{sampling probability for solution } p \text{ at start of cooling step } n$$

$$C_{i,n-1}^p = \text{constraint } i \text{ value for solution } p \text{ at end of cooling step } n-1$$

$$I = \text{number of active constraints}$$

For HPC applications, parallel Markov chains consisting of successively generated solutions may be assigned to independent processors (or groups of independent processors). As defined, PSA may be performed synchronously, where the Markov chain length is fixed for each processor, or asynchronously, where the Markov chains may be truncated to achieve optimal CPU load balance.

### 3. Algorithm

The algorithm consists of an initialization phase, where randomization of the solutions occurs and initial temperatures are established. This is followed by an optimization phase consisting of PSA and mixing of states components. Appendix A shows the pseudo-code for the algorithm. In summary, the objective function and each constraint will evolve according to a unique temperature. Each generated solution will be evaluated against the objective function and constraints according to the Metropolis criterion of Eq. 2 resulting in the set of acceptance probabilities. Individual state-transitions are then calculated and the solution accepted globally if the lowest constraint acceptance probability is satisfied. Upon completion of the set of parallel Markov chain, pooled statistics from each constraint are used with the global acceptance ratio to calculate updated temperatures according to Eq. 4 and 5. In addition, the set of current best solutions from the completed Markov chains are used to construct a probability density function according to Eq. 6 and 7 which is sampled to provide initial solutions for the subsequent Markov chains. To summarize the key points:

- a) Potential solutions are generated based on a common move generation strategy.
- b) Solutions are evaluated with respect to the individual acceptance criterion for each constraint, independent of the other constraints that might exist.
- c) Solutions are *globally* accepted or rejected based on having satisfied *all* constraint individual acceptance criteria.
- d) Local constraint and objective function statistics are accumulated based on *all* evaluated solutions within the Markov chain. This includes acceptances where a new best solution is assigned and rejections where the current best solution is maintained.

- e) Global acceptance ratios are calculated based on global accepted solutions.
- f) Starting solutions for each parallel Markov chain are obtained by sampling from the population of  $P$  current best *global solutions*, which comprises the union of solutions obtained for each Markov chain at completion of the previous temperature evaluation.

### III. RESULTS

The constraint annealing method is demonstrated within the FORMOSA-P code [8] for a 3-loop Westinghouse PWR loading pattern design formulated as a COP. The decision variables for this problem include fresh burnable poison design, fresh and exposed fuel placements, and gradient orientations of the exposed fuel bundles.

The move generation strategy is based on performing perturbations to the set of decision variables as a means of sampling a new solution. The possible perturbations include: 1) the binary swap of exposed fuel, 2) the binary swap of fresh and exposed fuel, 3) the change of fresh fuel burnable poison design in a single location, and 4) the change in exposed fuel gradient orientation in a single location.

For the current application, constraint annealing is executed synchronously, where each processor executes a Markov chain of length  $N$  to completion. The value of  $N$  is based on the number of perturbations possible for the move generation strategy for the binary swap of fresh and exposed fuel ( $N=160$  for the current problem). Convergence is based on a global acceptance ratio (set equal to 0.01) or detection of asymptotic convergence based on the objective function and constraint mean and standard deviation values. A value of  $\lambda=1.0$ , equal to one standard deviation within the stationary criteria of Eq. 3, was selected as the only 'adjustable' parameter. All problems were executed on a 24 CPU core machine (i.e. 24 parallel Markov chains).

Two cases are considered. Case A focuses on the assessment of the constraint annealing method as pertains to the robustness of the algorithm in identifying a global optimum solution. This is achieved by repeat execution of the same problem with different random number seeds. The goal is to assess performance, as measured by the evolution of key parameters during the progress of the optimization as well as the final results. For Case A the objective function is the maximization of EOC boron concentration at a target EOC exposure (i.e. minimizing the negative boron). Constraints for Case A include limits on FdH and peak rod burnup only.

For Case B the focus is on the demonstration of the constraint annealing method for a realistic core design problem, as defined by a unique objective function and constraint configuration created from the parameter list of Table I. Two objective functions are defined: 1) the maximization of EOC boron concentration at a target EOC exposure and 2) the minimization of vessel fluence.

Constraints include: 1) FdH, 2) rodDED FdH (i.e. the lead control bank at the rod insertion limit), 3) HZP MTC, 4) peak rod burnup, 5) steaming rate, and 6) alternate source term (AST). It is noted that Case B is a more restrictive subset of Case A. Table II describes the problem definition for each case.

TABLE II. Optimization problem configurations

Constraint	Case A Limit	Case B Limit
EOC Boron (ppm)	>0.0	> 0.0
Fluence	n/a	< 0.180
FdH	< 1.590	< 1.590
Rodded FdH	n/a	< 1.590
Peak Burnup (GWD/MTU)	< 54.00	< 54.00
MTC (pcm/°F)	n/a	< 0.0
Steaming Rate (lbm/ft <sup>2</sup> -hr)	n/a	< 4.00
AST (kW/ft)	n/a	= 0.00

Several items are important to note. Of the constraints listed in Table I, the fuel manufacturing and contingency constraints are readily addressed through the sampling procedure. This is particularly important for constraints such as GTRF where the exclusion of certain fuel placements will reduce the size of the decision space. Constraints such as those on enrichment and BP distributions are likewise addressed as part of the sampling process and reduce the decision space. The remainder of the constraints within Table I must be evaluated utilizing a core simulator. Most of the reactivity, thermal margin and exposure constraints are straightforward with some exceptions.

First, the fluence constraint is modeled as a weighted integral over cycle exposure of the assembly fuel rod powers (i.e. fast flux source terms), where higher weight factors indicate higher importance with respect to the overall vessel fluence critical weld locations. Fluence as defined is therefore a unitless parameter consistent with the definition of an average relative rod power. Second, the steaming rate is a core average value calculated as an integral over the core and the operating cycle of the amount of subcooled nucleate boiling vapor mass flux. It is used as a proxy for CIPS, a phenomena in PWRs where the uptake of soluble boron within the fuel crud layer can induce axial power offset swings during normal operation. Finally, AST establishes a limit on peak linear power (kW/ft) as a function of fuel rod burnup. AST addresses fuel release for postulated accidents such as loss of coolant, steamline break, and the fuel handling accident. AST can be the limiting constraint in some reactor designs due to the use of high burnup fuel in high power locations.

### 1. Case A Results

Table III displays the statistics results for the Case A problem executed 11 times utilizing different random numbers seeds. For all trials, the FdH and rod peak burnup

constraints were satisfied with an accompanying increase in EOC boron. As shown, the standard deviations for all parameters are small relative to the mean value indicating the optimization search algorithm is capable of finding essentially the same optimum solution. This is further reinforced by examination of the BOC  $k_{\infty}$  distribution (mean and standard deviation) which show similar results (Fig. 1 and 2) including the same fresh loading fuel pattern for all cases executed. The low standard deviations are also indicative of nearly identical fuel being loaded in the same core locations.

TABLE III. Summary of optimization results for COP, independent run statistics

Constraint	Limit	Mean	Std.
EOC Boron (ppm)	>0.0	+13.0	0.7
FdH	< 1.590	1.588	0.003
Peak Burnup (GWD/MTU)	< 54.00	53.45	0.08

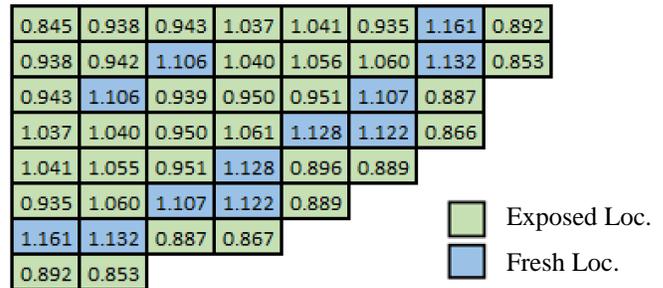


Fig. 1. BOC  $k_{\infty}$  distribution (mean value)

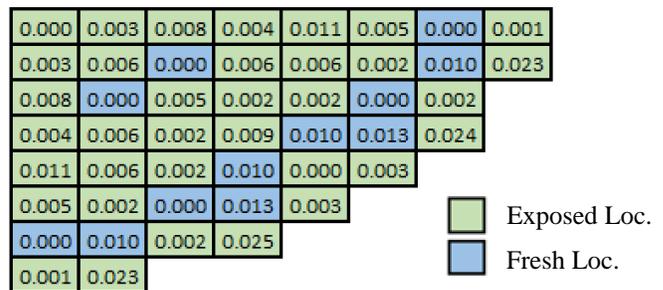


Fig. 2. BOC  $k_{\infty}$  distribution (standard deviation)

Fig. 3-5 display the algorithm performance over the course of the optimization search. The results shown are based on the average values for all trials executed. Fig. 3 shows the acceptance ratios for the FdH constraint and the EOC boron objective function as well as the global acceptance ratio. It is clear the global acceptance ratio is dominated by the satisfaction of the FdH constraint. Note that the peak rod burnup constraint is not a limiting constraint for this problem.

Fig. 4 displays the evolution of temperature during the optimization. As shown, a similar shape evolves for both the EOC boron objective function and the FdH constraint. The

exception is that the FdH temperature shows a sharp decrease towards the end of the optimization. This is consistent with the constraint violations being eliminated (i.e. a small standard deviation as appears in Eq. 4).

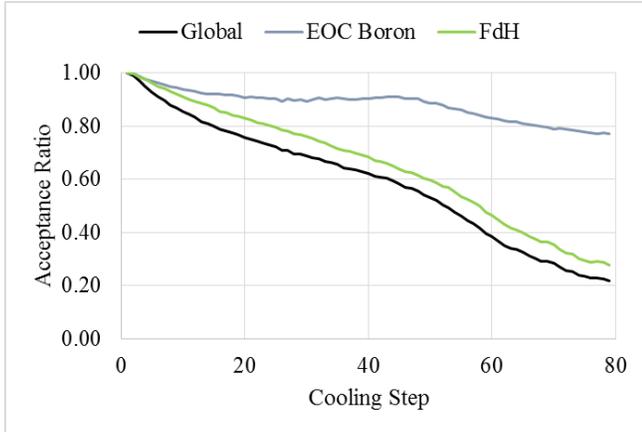


Fig. 3. Case A. Acceptance ratios (11 trial runs)

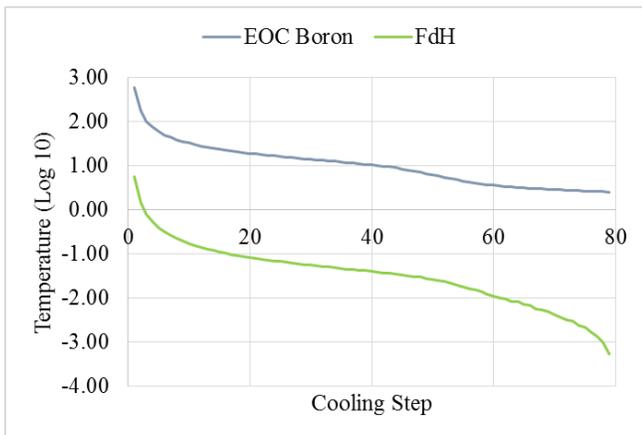


Fig. 4. Case A. Temperature evolution (average of 11 trial runs)

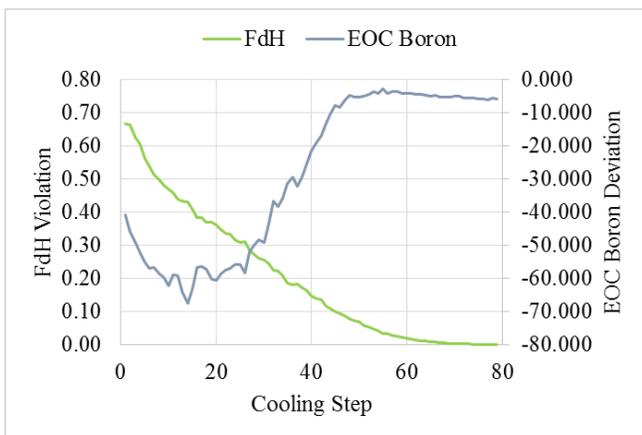


Fig. 5. Case A. Objective function and constraint evolution (11 trial runs)

Fig. 5 shows the evolution of the EOC boron and FdH (mean values) at each cooling step of the optimization. For the EOC boron objective function, the mean asymptotically approaches -5.5 (equal to +5.5 ppm in boron). This compares with a much higher value of +13.0 ppm for the optimum solution which is contained within the population. This is readily explained by Fig. 3 which shows the EOC boron objective function maintaining a high acceptance rate towards the end of the optimization search (~80%).

## 2. Case B Results

Case B incorporates an additional objective function and four additional constraints to the Case A problem specification. Fig. 6 displays the acceptance ratios for the global objective function and constraints as well as the global acceptance ratio during the optimization search. Fig. 7 similarly displays the temperature results. The flat line behavior within the temperature curves represent the point at which constraint violations have been eliminated. All constraint limits were satisfied at the completion of the search.

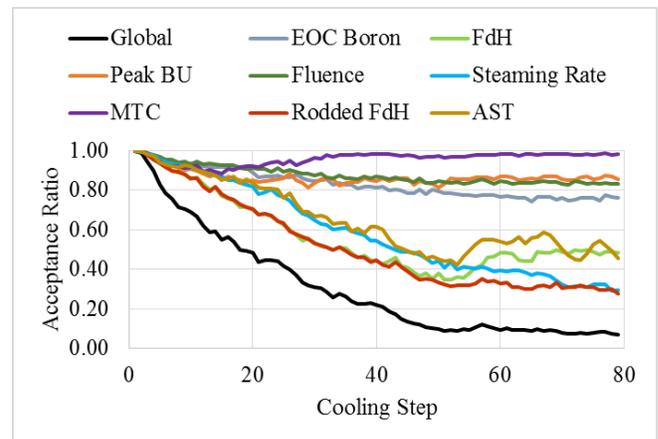


Fig. 6. Case B. Acceptance ratios

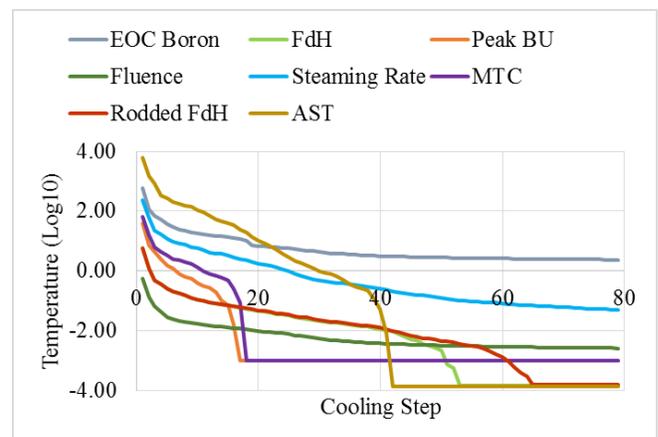


Fig. 7. Case B. Temperature evolution

In contrast to Case A, the limiting constraints for Case B in terms of dominating the global acceptance rate is more complex. Both Fdh (rodded and unrodded) are bounding during the first half of the optimization search until the constraint violations are eliminated. Steaming rate becomes dominant in the latter stages of the optimization.

Fig. 8-9 display the evolution of the objective function values during the course of the optimization. As shown, both the EOC boron and fluence exhibit similar behavior which is explained by noting that higher EOC boron is consistent with reduced core leakage and thus vessel fluence. As a result, both EOC boron and fluence are impacted greatly by the selection of fuel on the core periphery. The optimization results in improvements to both objective functions with optimum values of +4.9 ppm for EOC boron and 0.170 for fluence (representing a 5% improvement versus the target value of 0.180).

A comparison with Case A (Fig. 5) shows consistent EOC boron behavior. This is expected since the Case B specification was developed from the Case A problem definition. However, the boron improvement for Case B when compared with Case A (+13 ppm) is much reduced. A comparison with the calculated fluence for Case A (0.186) shows significant improvement. It is noted that fluence was not considered in the Case A problem definition.

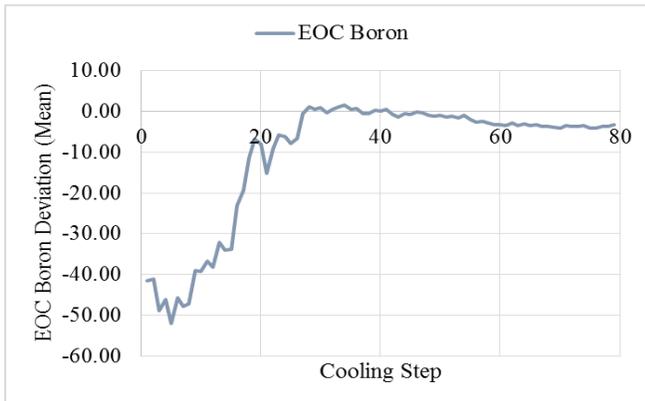


Fig. 8. Case B. EOC boron objective function evolution

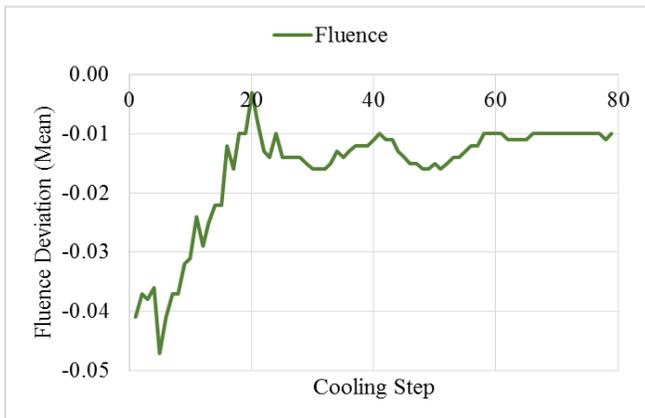


Fig. 9. Case B. Fluence objective function evolution

Fig. 10-14 display the evolution of the constraint values during the course of the optimization. The constraint violations on peak rod burnup, MTC, and AST are eliminated relatively early during the search as shown in Fig. 10, 11 and 12, respectively. It is noted that peak rod burnup and AST are impacted by the local powers within higher burnup fuel. In contrast, the MTC is impacted by the total burnable poison loading within the fresh fuel.

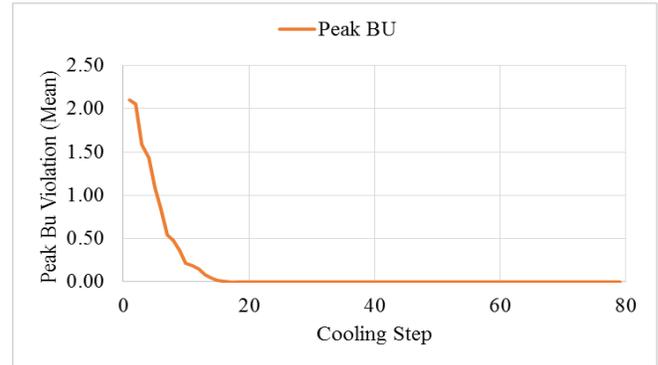


Fig. 10. Case B. Peak burnup constraint evolution

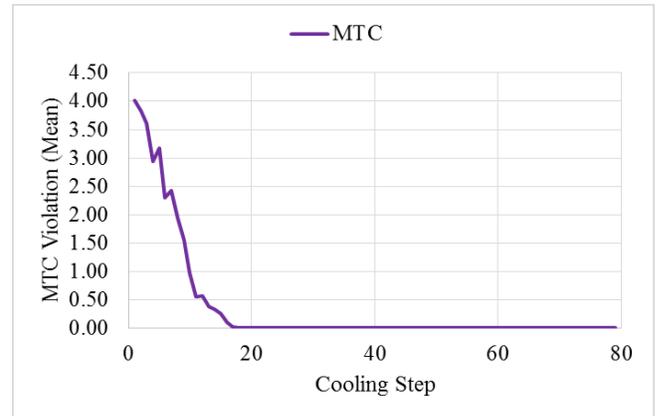


Fig. 11. Case B. MTC constraint evolution

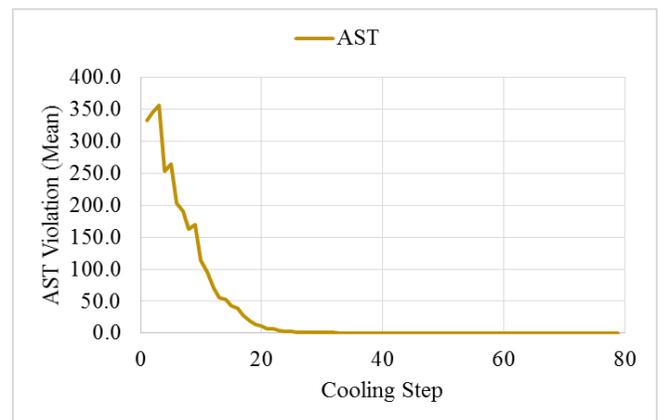


Fig. 12. Case B. AST constraint evolution

The next constraints eliminated during the search are the unrodded and rodded FdH as shown in Fig. 13 and 14,

respectively. It is noted that satisfying both FdH constraints will seek solutions where the limiting peaking factors are located in the vicinity of the lead control bank locations. Finally, Fig. 15 shows the steaming rate constraint which is eliminated during the late stages of the optimization search.

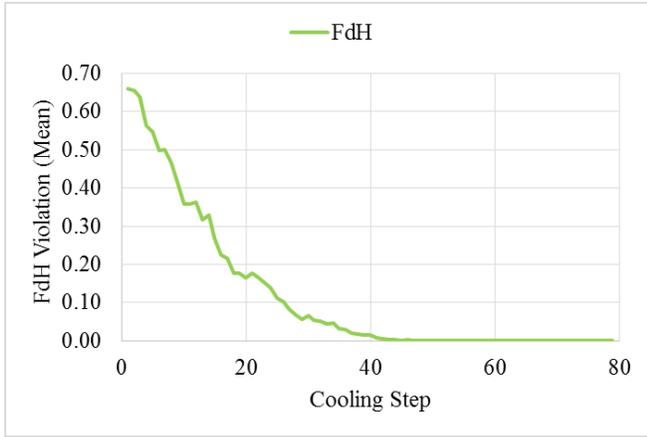


Fig. 13. Case B. FdH constraint evolution

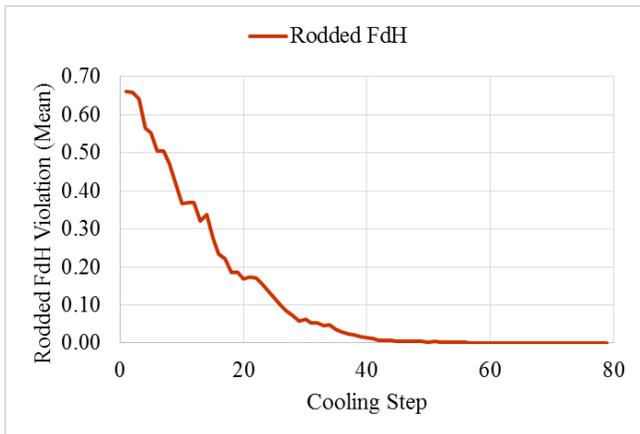


Fig. 14. Case B. Rodded FdH constraint evolution

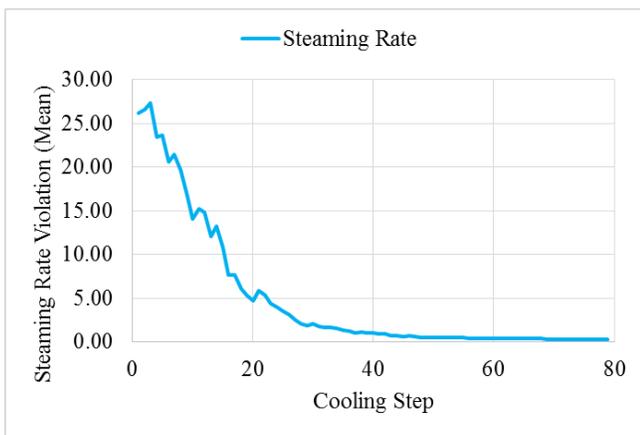


Fig. 15. Case B. Steaming rate constraint evolution

Fig. 16-17 shows a comparison of BOC peaking factor distributions (unrodded) for the Case A and Case B optimized results. Also displayed are the locations of fresh and exposed fuel as well as the limiting assembly locations in terms of their fluence contribution. The locations of the lead control rod banks are highlighted in red. Several key points are made. First, the fresh loading patterns show significant differences among both interior (groupings of 8) and axis (groupings of 4) core locations. The result is a significant reduction in peaking factors in the fluence locations for Case B. Second, Case B shows reduced peaking factors in the fresh core locations as well as significantly reduced peaking factors in the key fluence locations. Examining the locations of the lead control bank locations, it is clear that rod insertion will increase the power in the limiting locations away from the control rods. For Case A, the fresh locations with peaking of 1.58 show an increase of 0.02 with rods inserted (leading to violation). For Case B, a similar increase 0.02 maintains the peaking factor below the constraint violation limit. It is again noted that neither fluence nor rodded FdH were active constraints for Case A. Thus, the results are consistent with the problem specifications.

0.571	0.890	1.184	1.513	1.499	1.224	1.589	0.682
0.890	1.098	1.587	1.455	1.507	1.466	1.531	0.605
1.184	1.590	1.175	1.162	1.255	1.555	0.924	
1.513	1.470	1.154	1.431	1.580	1.416	0.613	
1.499	1.508	1.260	1.580	1.062	0.697		
1.224	1.469	1.555	1.415	0.697			
1.589	1.532	0.922	0.614				
0.682	0.588						

Exposed Loc.  
 Fresh Loc.  
 Fluence Loc.

Fig. 16. Case A. BOC peaking factors (unrodded)

0.568	0.879	1.201	1.410	1.135	1.410	1.145	0.538
0.879	0.999	1.518	1.435	1.206	1.569	1.525	0.499
1.201	1.506	1.106	1.417	1.218	1.503	0.943	
1.410	1.421	1.413	1.436	1.442	1.570	0.665	
1.135	1.198	1.204	1.441	1.565	0.893		
1.410	1.553	1.497	1.567	0.893			
1.145	1.514	0.941	0.670				
0.538	0.518						

Exposed Loc.  
 Fresh Loc.  
 Fluence Loc.

Fig. 17. Case B. BOC peaking factors (unrodded)

## IV. CONCLUSIONS

A penalty-free algorithm has been developed and demonstrated for solution of fuel cycle optimization problems. The method eliminates any need for penalty functions and problem specific adjustable parameters as are often encountered in multi-objective, multi-constrained optimization problems. Two case studies were presented. The first examined the robustness of the algorithm with respect to algorithm performance and the ability of the optimization search to identify global optimum solutions. The second case expanded upon the first by incorporating an additional objective function and additional constraints, including realistic core design limits on reactivity, thermal margins, and fuel exposure. The performance results obtained are shown to be consistent with not only the problem specification but the expected physical behavior of the core design.

## APPENDIX A: CONSTRAINT ANNEALING ALGORITHM

### Initialization Phase

0. Assign randomly generated solution to each processor **k**
1. For each processor **k** initiate independent Markov chains & analyze; continue until completed
  - 1a. perform sampling and evaluate solution objective and constraint values
  - 1b. accumulate statistics for each constraint **i**
  - 1c. accept solution, assigning as the new best solution
2. Communicate best solution & constraint statistics from each processor **k** back to master
3. Calculate initial constraint temperatures for each constraint **i**
  - 3a. calculate constraint value standard deviation based on Markov chain
  - 3b. calculate temperature as a multiple of standard deviation (e.g. 20x)
4. Broadcast constraint temperatures to each processor **k**

### Optimization Phase

**(perform until converged)**

#### Parallel SA

1. For each processor **k** initiate independent Markov chain & analyze; continue until all chains are completed
  - 1a. perform sampling and evaluate solution objective and constraint values
  - 1b. accumulate statistics for each constraint **i**
  - 1c. calculate constraint acceptance probabilities (Eq. 2)
  - 1d. accept/reject solution based on the minimum acceptance probability for all constraints; if accepted assign new best solution, otherwise restore previous best solution
2. Communicate best solution & constraint statistics from each processor **k** back to master processor

### Optimization Phase (continued)

#### Mixing of States

3. Calculate pooled statistics from each processor
  - 3a. calculate std. dev. for each constraint **i**
  - 3b. calculate global acceptance ratio
  - 3c. **Exit optimization** if global acceptance ratio satisfied or detection of asymptotic convergence via constraint statistics
4. Calculate updated constraint temperatures for each constraint **i** (Eq. 4 & 5)
5. Broadcast constraint temperatures to each processor **k**
6. Build pdf sampling distribution based on the set of current best solutions (Eq. 6 & 7)
7. For each processor **k**, sample from the pdf and assign a new best solution
8. Continue **Parallel SA** optimization

## NOMENCLATURE

LWR – light water reactor  
PWR – pressurized water reactor  
HWP – hot zero power  
BOC – beginning of cycle  
EOC – end of cycle  
FdH – hot channel peaking factor (2D)  
FQ – total peaking factor (3D)  
MTC – moderator temperature coefficient  
AST – alternate source term  
SDM – shutdown margin  
CIPS – crud induced power shift  
GTRF – grid to rod fretting  
CILC – crud induced localized corrosion

## REFERENCES

1. S. KIRKPATRICK, C. D. GELATT, and M. P. VECCHI, "Optimization by Simulated Annealing," *Science*, 220, 671-680 (1983).
2. T. BÄCK, D. FOGEL, & Z. MICHALEWICZ, *Handbook of Evolutionary Computation*, C5.2, IOP Publishing Ltd. Bristol, UK (1997).
3. D. KROPACZEK, "COPERNICUS: a multi-cycle optimization code for nuclear fuel based on parallel simulated annealing with mixing of states," *Prog. in Nucl. Energy*, 53, 554-561 (2011).
4. T. K. PARK, H. G. JOO., & C. H. KIM, "Multicycle Fuel Loading Pattern Optimization by Multiobjective Simulated Annealing Employing Adaptively Constrained Discontinuous Penalty Function," *Nuclear Science and Engineering*, 176, 226-239 (2014).
5. D. KROPACZEK, "Penalty-free method for multi-constrained optimization problems based on parallel simulated annealing" *Trans. of Am. Nucl. Soc.*, 115, 543-548 (2016).

6. J. LAM, "An Efficient Simulated Annealing Schedule"  
*Ph.D. Dissertation.*, Yale University, New Haven, Connecticut (1998).
7. K. CHU, Y. DENG, and J. REINITZ, "Parallel Simulated Annealing by Mixing of States," *J. Comp. Phys.*, 148, 646-662 (1999).
8. P. M. KELLER., "FORMOSA-P: Constrained Multiobjective Simulated Annealing Methodology"  
*Proc. M&C 2001*, Salt Lake City, Utah, Sep. 2001, American Nuclear Society (2001) (CD-ROM).