Advances in Nuclear Fuel Management IV (ANFM 2009)
Hilton Head Island, South Carolina, USA, April 12-15, 2009, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2009)

COPERNICUS: A Multi-Cycle Nuclear Fuel Optimization Code Based on Coupled In-core Constraints

David J. Kropaczek
Studsvik Scandpower, Inc.
1015 Ashes Drive, Suite 205
Wilmington, NC 28405
USA

Dave.Kropaczek@studsvik.com

Keywords: Multi-Cycle Optimization Fuel

ABSTRACT

COPERNICUS is the Studsvik code for performing nuclear fuel optimization over a multi-cycle planning horizon that provides for an implicit coupling between traditionally separate in-core and out-of-core fuel management decisions. These decisions include determination of: fresh fuel region size; sub-region enrichments and bundle designs; exposed fuel re-use; and core loading pattern. The COPERNICUS methodology is based on a parallel implementation of the Simulated Annealing optimization algorithm, modified by the technique of Mixing of States that allows for deployment in a processor scalable environment. COPERNICUS utilizes the 3-D licensing grade code SIMULATE for evaluation of all core loading pattern constraints, such as those involving reactivity and thermal margin requirements. Results are presented for a transition cycle design that compares performance of multi-cycle optimization to successive, single cycle optimization with regard to reducing levelized fuel costs.

1. INTRODUCTION

Fuel management for Light Water Reactors (LWRs) may be divided into out-of-core and in-core fuel management tasks. Out-of-core fuel management deals with the long range energy planning for fuel procurement while 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 (LP), and the operational strategy (i.e. control rods, flow, moderator temperature). Efforts in the nuclear industry to address the out-of-core and in-core problems as a coupled, constrained optimization problem have been quite limited. The recent enhancements to the OCEON code (Anderson, et al., 2007a, 2007b, Du, 2008) are one such example of solving the coupled problem.

Multi-Cycle Optimization (MCO) for nuclear fuel management eliminates the distinction between in-core and out-of-core fuel management tasks by redefining the problem to be solved as one of determining the set of successive, single cycle core designs for a planning horizon that minimizes levelized fuel cost subject to constraints.

MCO constraints include both out-of-core and in-core parameters, such as fuel design criteria, target energy production in each cycle, thermal and reactivity margins, and discharge exposure. MCO is desirable to address those situations encountered during fuel cycle planning where the potential exists for significant deviation from equilibrium conditions (i.e. changes in region size and/or average enrichment in successive cycles). Examples of such planning situations include changes in cycle length, extended power uprate (EPU), and introduction of advanced fuel products or poison designs. Of particular importance is the design of the *transition* fuel cycles, defined as those non-equilibrium fuel cycles that span the cycles from initial change en route to achieving a new equilibrium condition. MCO takes into consideration the fresh bundle design, exposed bundle selection and previous operating history as part of each single cycle's core design, while allowing coupled in-core and out-of-core constraints, such as those on peak discharge burnup, to be factored into the fuel cost evaluation.

2. COPERNICUS METHODOLOGY

2.1 Multi-Cycle Problem Definition

A multi-cycle fuel planning horizon consists of a number of successive, single fuel cycles, where each cycle is associated with an energy production plan and constraints. The energy production plan encompasses all items impacting power output within a given cycle including: cycle length, power level requirements, operational strategy (control rods, core flow), and coast down. Constraints include all thermal, reactivity, and fuel performance limits established to protect fuel integrity. The exposed fuel inventory at the completion of one fuel cycle is used as inputs to the design of the subsequent fuel cycles. The characteristics of the exposed fuel inventory in a given fuel cycle has a direct impact on the fresh fuel region size, fresh fuel bundle design (enrichment and poison distributions), and loading pattern. Thus, MCO may be thought of as a problem of determining the selection and distribution of exposed and fresh fuel in each cycle of a multi-cycle fuel planning horizon that minimizes fuel cost.

A fuel cycle j may therefore be defined by a beginning of cycle (BOC) fuel inventory $\mathbf{I}_j$ that is comprised of the exposed fuel bundles within the core and the discharge pool, following the completion of the previous fuel cycle j-1. A loading instruction operator, $\mathbf{O}_j$, may be defined that provides a mapping of the fuel inventory, $\mathbf{I}_j$, to the core loading pattern, $\mathbf{LP}_j$.

The operator $\mathbf{O}_j$ determines those exposed bundles and previously discharged bundles that will be re-inserted into the core. With respect to the exposed fuel, the operator $\mathbf{O}_j$ addresses the following: 1) pairings of exposed bundles by similar characteristics (for example, bundles with similar average and symmetric gradient values of reactivity or exposure); 2) ranked ordering of core locations, such as by reactivity or exposure; and 3) gradient orientations for exposed bundles as mapped to the core locations (i.e. cross-core shuffles). In addition, $\mathbf{O}_j$ provides 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 fuel 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 core and discharge pool, is a 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 (see Eq. 1):

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

The MCO problem may be stated as follows. Given:

1) A multi-cycle energy production plan
2) A set of multi-cycle constraints for cycle's *j* through $(j+n)$
3) A cycle *j* fuel inventory, $\mathbf{I}_j$
4) A set of available fresh fuel designs $\{\mathbf{F}_j, \mathbf{F}_{j+1}, ..., \mathbf{F}_{j+n}\}$,

Determine the set of operators, $\{\mathbf{O}_j, \mathbf{O}_{j+1}, ..., \mathbf{O}_{j+n}\}$, that minimizes fuel cycle cost over the planning horizon. Note that the set of operators, $\{\mathbf{O}_j, \mathbf{O}_{j+1}, ..., \mathbf{O}_{j+n}\}$, uniquely determines the set of successive, single cycle core loading patterns, $\{\mathbf{LP}_j, \mathbf{LP}_{j+1}, ..., \mathbf{LP}_{j+n}\}$, for the planning horizon.

2.2 Parallel Simulated Annealing Algorithm

Well-suited to solution of the MCO problem is the method of Simulated Annealing (SA), a rejection sampling algorithm that belongs to a class of Markov chain Monte Carlo techniques. SA optimization (Kirkpatrick, et al. 1983; Aarts and Van Laarhoven, 1985) simulates the slow cooling of a 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 objective function, *C*, which captures the objective function and constraints of the optimization problem. SA is inherently a serial algorithm, since each new solution is generated based on the current accepted solution within the Markov chain. COPERNICUS employs a parallel version of SA based on *Mixing of States* which has been shown to scale with the number of processors (Chu, 1999; Kropaczek, 2008).

2.2.1 Move Generation Strategy

The move generation strategy is critical to creating the Markov chains within the SA algorithm. Markov chains are generated stochastically and have the property that given a current state, future states are independent of all past states. Thus, each solution within SA depends only on the most recent accepted solution (i.e. the *current best* solution). The sampling strategy for generating candidate SA solutions is based on the creation of a set of perturbed loading instruction operators $\{\underline{\mathbf{O}}_j, \underline{\mathbf{O}}_{j+1}, ..., \underline{\mathbf{O}}_{j+n}\}$ from the current best set of operators $\{\mathbf{O}_j, \mathbf{O}_{j+1}, ..., \mathbf{O}_{j+n}\}$.

As discussed previously, each cycle *j* operator, $\mathbf{O}_j$, defines a set of loading operations that may be performed on the available exposed fuel inventory $\mathbf{I}_j$ and fresh designs $\mathbf{F}_j$. Perturbing $\mathbf{O}_j$ therefore involves a change in one or several loading decisions. Six categories of $\mathbf{O}_j$ perturbations, *m*, are defined and are given by:

1) Assignment of exposed fuel core locations and the ranked ordering of those locations, such as by reactivity or exposure
2) Assignment of gradient orientations to the exposed fuel core locations
3) Assignment of fresh fuel core locations
4) Fresh fuel region (and sub-region) sizes
5) Fresh fuel sub-region enrichments
6) Fresh fuel bundle design (including burnable poison)

It is noted that for the current application, the pairing of exposed bundles within $\mathbf{I}_j$ is performed via strict rules and is not included in the set of $\mathbf{O}_j$ perturbations. For example, a pairing might be based on a reactivity matching of quadrant or octant in mirror or rotational gradient symmetry.

Within the current MCO algorithm, a random sampling process is employed to select the cycle, *j*, and type of perturbation, *m*, to be applied to $\{\mathbf{O}_j, \mathbf{O}_{j+1}, ..., \mathbf{O}_{j+n}\}$ in the creation of $\{\underline{\mathbf{O}}_j, \underline{\mathbf{O}}_{j+1}, ..., \underline{\mathbf{O}}_{j+n}\}$. For example, modifying $\mathbf{O}_j$ to reflect a change in the ranked ordering of exposed fuel 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 assigned to exposed fuel to a fresh fuel location would change the fresh fuel region size. Through changes to $\{\mathbf{O}_j, \mathbf{O}_{j+1}, ..., \mathbf{O}_{j+n}\}$, it is possible to manifest a broad range of changes to $\{\mathbf{LP}_j, \mathbf{LP}_{j+1}, ..., \mathbf{LP}_{j+n}\}$, including changes in region size, enrichment, bundle design, and fresh and exposed fuel placements over successive cycles.

2.2.2 MCO Objective Function

The objective function, C, within COPERNICUS is defined as a cumulative objective function as follows (see Eq. 2):

$$C = \lambda(FCC - FCC_{targ}) + \sum_{j=1}^{n} x_j \sum_{l=1}^{q_j} w_{j,l} P_{j,l} \tag{2}$$

Where,

FCC = levelized fuel cycle cost (subscript $targ$ designates target value)
λ = fuel cost importance weight
x_j = importance weight applicable to cycle j
n = number of cycles in planning horizon
q_j = number of active constraints in cycle j
$w_{j,l}$ = importance weight applicable to constraint l in cycle j
$P_{j,l}$ = violation measure for constraint l in cycle j

The levelized fuel cycle cost, FCC, represents the average cost per unit of electricity produced for the planning horizon and includes: uranium ore, conversion, separative work, fabrication, back-end costs and carrying charges (pre-operation and in-core). In-core carrying charges are calculated based on a linear energy allocation model. It is important to note that alternative formulations for FCC are possible, such as the levelized fuel cost for a specific cycle.

The importance weight, $w_{j,l}$, is a measure of the emphasis the MCO will place on satisfying a given constraint l in cycle j. In a similar manner, x_j is a measure of the emphasis the MCO will place on a given cycle j. Through preferential weighting, it is possible to enforce strict satisfaction of constraints for the earliest cycle(s) under design in the planning horizon with relaxation of certain constraints for those cycles farther out in the future. This is an important consideration from a practical design viewpoint since the uncertainties associated with future cycle planning increase the farther out one analyzes, due to such variables as energy plan, uranium prices, operations, and outage scheduling.

For a Pressurized Water Reactor (PWR), examples of the active constraints, $P_{j,l}$, include (but are not limited to) enthalpy rise hot channel peaking factor ($F_{\Delta H}$), target end of cycle boron, peak discharge burnup, and maximum boron over the cycle.

2.2.3 Cooling Schedule

One of the key concepts in SA is the *cooling schedule*, which is the algorithm for control of the *temperature* over the course of the optimization. Within SA, temperature is periodically adjusted following completion of a Markov chain and directly impacts the acceptance probability for solutions generated as follows:

$$p_{l,k} = \begin{cases} \exp(-(C_{l,k} - C^*)/T_k) & \text{for} \quad C_{l,k} > C^* \\ 1.0 & \text{otherwise} \end{cases} \tag{3}$$

Where,

- $p_{l,k}$ = acceptance probability for solution generated at Markov step l, cooling step k
- T_k = temperature at cooling step k
- $C_{l,k}$ = objective function at Markov step l, cooling step k
- C^* = objective function for most recent accepted solution (i.e. *current best*)

COPERNICUS utilizes an adaptive, exponential cooling schedule developed by Lam and Delosme, (1988a, 1988b), that is based on optimizing the rate at which the temperature, T_k, 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 cooling schedule is described by Eqs. 4 through 6 as:

$$T_0 = \alpha\ \sigma^* \tag{4}$$

$$s_{k+1} = s_k + \lambda\left(\frac{1}{\sigma(s_k)}\right)\left(\frac{1}{s_k^2\sigma^2(s_k)}\right)G(\rho(s_k)) \tag{5}$$

$$G(\rho) = \left(\frac{4\rho(1-\rho)^2}{(2-\rho)^2}\right) \tag{6}$$

Where,

σ^* = standard deviation of C during initialization
α = initialization parameter, (typical values $1.0 < \alpha < 2.0$)
s_k = $1/T_k$
$\sigma(s_k)$ = standard deviation of C for accepted solutions at s_k
$\rho(s_k)$ = move acceptance ratio at s_k
λ = quality factor, (typical values $1.0 < \lambda < 2.0$)

Temperature initialization is performed according to Eq. 4, by evaluating an initial Markov chain with acceptance probability set equal to 1. For subsequent cooling steps, Eqs. 5 and 6 apply. Note that both the standard deviation of the objective function, $\sigma(s_k)$ (for accepted solutions only) and the acceptance ratio for generated moves, $\rho(s_k)$, are used in the calculation of T_k. The quality factor, λ, provides for enhancement or dampening of the cooling rate.

2.2.4 Parallel SA with Mixing of States

Within parallel SA, a single cooling schedule is maintained with the initiation of concurrent Markov chains, defined by the process p, at the start of each k_{th} temperature evaluation. Starting solutions for each Markov chain are obtained by sampling from the population of N current best solutions, which is comprised of the union of solutions obtained from each Markov chain of the previous $(k\text{-}1)_{th}$ temperature evaluation. The sampling probability is calculated for each member of the population based on the member objective function for each solution and the temperature, T_k:

$$J_{s,k} = \frac{\exp(-C^*_{s,k-1} / T_k)}{\sum_{p=1}^{N}\exp(-C^*_{p,k-1} / T_k)} \tag{7}$$

Where,

$J_{s,k}$ = sample probability for solution s at the start of cooling step k

$C^*_{p,k}$ = objective function for solution of parallel Markov chain p, at completion of cooling step k

Each temperature evaluation begins with a re-initialization of the Markov chain for each parallel process, p. Eq. 7 defines the probability distribution from which the starting solutions are sampled. As is evident, lower objective function solutions are sampled with higher probability. This leads to natural biasing of the search towards higher quality solutions while not completely excluding poor solutions from the search space.

For the cooling schedule described by Eqs. 4 through 6, the standard deviation of the objective function, $\sigma(s_k)$, is based on the pooled statistics from the parallel Markov chains at the k_{th} temperature evaluation. Similarly, the acceptance ratio, $\rho(s_k)$, is likewise calculated based on pooled statistics. In this manner, information from the parallel Markov chains, which are independent, is exchanged. Note that at the completion of each $(k\text{-}1)_{th}$ cooling step, the pooled statistics for *all* solutions accepted during the cooling step form the basis of the subsequent T_k calculation. Convergence of the algorithm is detected when $\rho(s_k)$, falls below a certain threshold (~10% for the current application).

2.2.5 Parallelization Considerations

CPU run-time performance for solution of MCO problems using parallel SA with Mixing of States has been shown to scale with the number of processors (Kropaczek, 2008) without degradation in the quality of optimized results. To this end, COPERNICUS code structuring was developed to accommodate a massively parallel, distributed computing environment based on MPICH2 (Gropp, et. al., 2007). MPICH2 is a high-performance and portable implementation of the Message Passing Interface (MPI) standard that supports different computation and communication platforms. Coding in MPI, however, requires careful consideration of data flow between the master and slave processes as well as synchronization. To maximize efficiency, as much of the CPU work should be delegated to the slave processes as possible, especially the processing of all licensing code inputs and output results.

Within COPERNICUS, the master process is therefore responsible only for those portions of the algorithm which are serial, such as processing of COPERNICUS input options, temperature calculation, the calculation of sampling probabilities (i.e. mixing of states) for re-initialization of the Markov chains, and periodic archiving of overall best solutions. All additional calculations are performed in parallel, with each slave process assigned a Markov chain that includes generation of the perturbed loading instruction operators $\{\underline{\mathbf{O}}_j, \underline{\mathbf{O}}_{j+1}, ..., \underline{\mathbf{O}}_{j+n}\}$, creation and spawning of the successive, single cycle licensing simulator depletion cases, and processing of results with respect to solution acceptance as per Eq. 3.

3. RESULTS

3.1 Test Problem

A 3-loop PWR, 17x17 lattice, transition cycle design, beginning in Cycle 3 was selected for analysis. This problem was selected because it is especially challenging from a design perspective. The transition is characterized by a change in cycle energy from 300 EFPD (effective full power days) to 570 EFPD that is accompanied by a new fuel product introduction, beginning in Cycle 3, that allows for an increase in design $F_{\Delta H}$. Also introduced in the transition are 6 inch top and bottom axial blankets at 2.0 w/o. The Cycle 1 and 2 LP designs are based on an 'Out-In' loading strategy, whereby the most reactive fuel is placed towards the core periphery (resulting in higher core leakage). Cycle 2 contains a fresh region size of 52 bundles at 4.4 w/o. Table 1 describes the transition cycle details, including the Cycles 1 and 2 fuel cycle description. Gadolinium (i.e. Gd_2O_3) at 4 w/o and 6 w/o is used as the integral burnable absorber (BA) for all cycles and is in configurations ranging from 4 to 24 rods within the fuel lattice.

Active constraints for this problem include $F_{\Delta H}$ (1.480), maximum boron (1400 ppm), peak discharge exposure (60. GWd/t), and end-of-cycle boron (+/- 5 ppm). For Cycles 3 - 7, either one or two enrichment sub-regions may be used in each cycle. A total of thirteen gadolinium patterns are available for each enrichment sub-region, with a minimum spacing of 0.2 w/o required between enrichment sub-regions. No restrictions are placed on the LP strategy for Cycles 3-7, although the natural result based on optimization will be towards low-leakage since this is most consistent with minimizing fuel cost. For purposes of MCO, reinserts from the fuel pool are allowed for all cycles, although this is an option within COPERNICUS. Thus, carryover fuel from Cycle 2 as well as the earlier discharged fuel from Cycle 1 is available to load, beginning in Cycle 3.

Cycle	Cycle Energy (GWd/t)	Enrichment Sub-regions	Enrichment (w/o)	BA Designs	$F_{\Delta H}$ Limit	LP Strategy
1	13.8	3	2.0 / 2.4 / 3.0	4	1.435	Out-In
2	11.6	1	4.4	2	1.435	Out-In
3	14.0	1 or 2	3.0 - 5.0, Δ=0.2	13	1.480	Low-Leakage
4	18.0	1 or 2	3.0 - 5.0, Δ=0.2	13	1.480	Low-Leakage
5	22.0	1 or 2	3.0 - 5.0, Δ=0.2	13	1.480	Low-Leakage
6	22.0	1 or 2	3.0 - 5.0, Δ=0.2	13	1.480	Low-Leakage
7	22.0	1 or 2	3.0 - 5.0, Δ=0.2	13	1.480	Low-Leakage

Table 1. Transition Details (Cycles 3-7)

3.2 Multi-Cycle Optimization Performance

The COPERNICUS code was executed for the above MCO problem with the objective of minimizing levelized fuel cost over the planning horizon. A separate set of successive, single cycle optimization (SCO) cases were executed with COPERNICUS in order to provide a basis for comparison.

All depletions were performed with the SIMULATE core simulator (Rempe, et. al. 1989) using quarter core symmetry, 24 axial mesh, 1 node/assembly (with intranodal cross sections), pin power reconstruction, and between 12 and 14 depletion steps per cycle depletion. A 40 CPU dedicated Linux cluster employing MPICH2 was used as the computational platform for the analysis. Execution was completed in approximately 12 hrs.

COPERNICUS performance as executed on 40 CPUs is shown in Figure 1. The objective function represents the mean and standard deviation for all solutions accepted during a cooling step. The objective function is constructed such that positive values represent constraint violations in any core loading pattern in the planning horizon, including *FCC* exceeding the target fuel cost value. Negative values represent satisfaction of all in-core constraint limits while reducing the levelized fuel cost ($/MWh) below the target value for the planning horizon. The acceptance ratio is the fraction of accepted solutions to total solutions sampled during a cooling step.

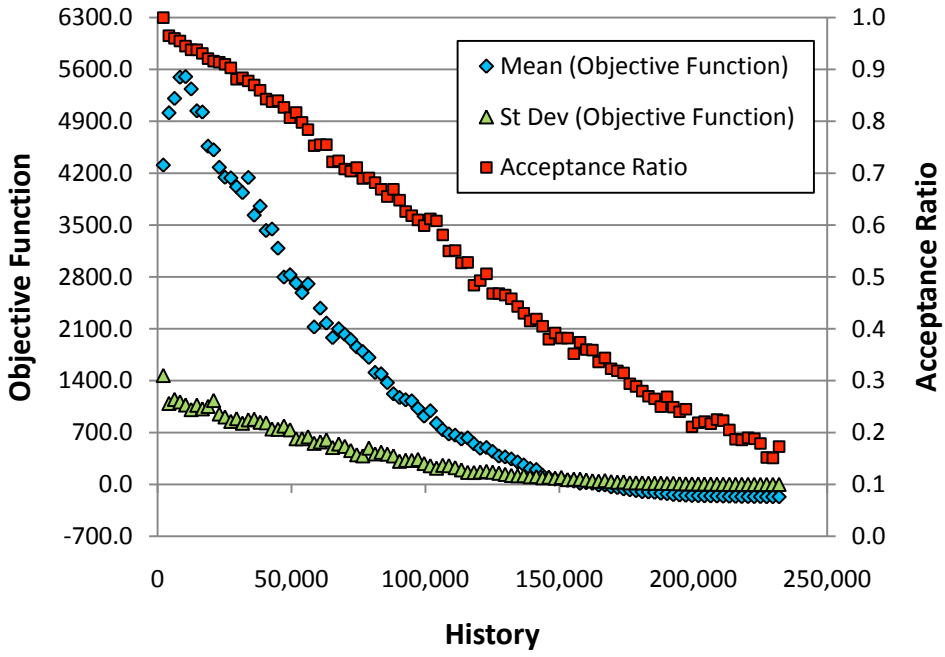

Figure 1. MCO Objective Function Performance (40 CPUs)

As shown in Figure 1, the observed exponential behavior of the mean and standard deviation of the objective function versus history is consistent with the adaptive cooling schedule for SA temperature as described by Eqs. 4 through 6. The acceptance ratio exhibits fairly smooth, linear behavior, without showing abrupt changes. This is an important aspect for estimating time to completion as convergence is detected based on a minimum acceptance ratio. Note that the results shown represent the pooled statistics across all 40 CPUs employed during the MCO.

Figure 2 displays the behavior of the COPERNICUS optimization with respect to acceptances by perturbation type. Perturbations are defined by changes to: *Region Size*, *Enrichmen*t, *Fresh LP* (bundle placement), *Fresh Design*, *Exposed LP* (bundle selection and placement), and *Orientation* (exposed fuel gradient).

Within Figure 2, region size and enrichment perturbation types have the lowest frequency of acceptance. Acceptance rates for these perturbations tend toward zero at a much faster rate than the other perturbation types. This is consistent with the experience that changes to region size and sub-region enrichments will have the greatest impact on levelized fuel costs through the fuel cycle cost component of Eq. 2. Although the number of acceptances for region size and enrichment decrease to zero by history 180,000 (at approximately 75% completion), it is important to note that the out-of-core and in-core problems remain tightly coupled.

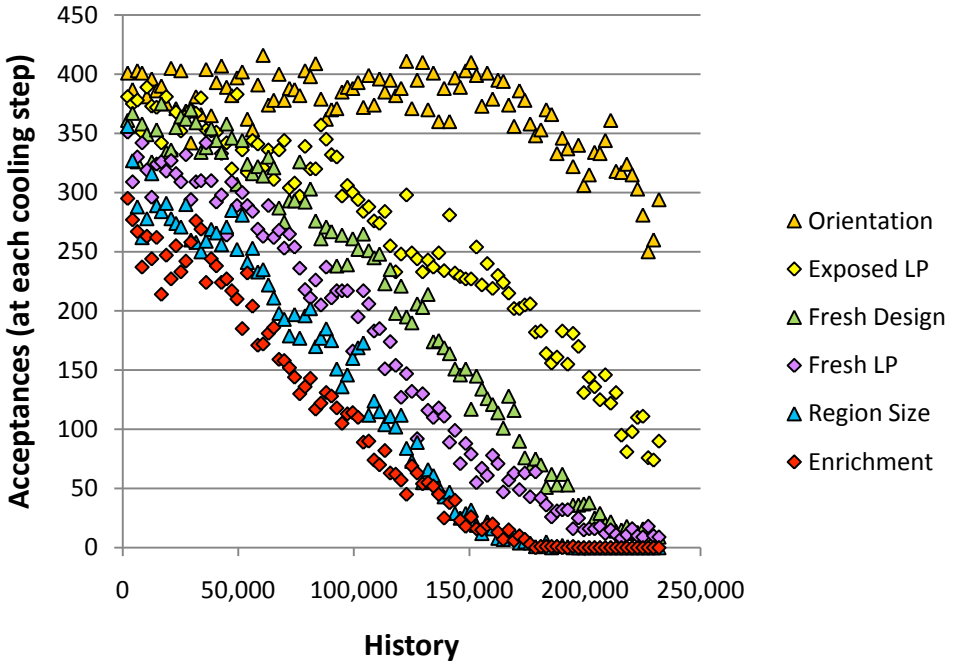

Figure 2. Acceptances by Perturbation Type (MCO)

The remaining perturbation types of Figure 2 determine the detailed behavior of the single cycle core LPs in each of the cycles of the planning horizon (i.e. Fresh LP, Fresh Design, Exposed LP, and Orientation). The acceptance frequency for each of the perturbation types are consistent with the impact that each has on regional power sharings, assembly power, local power peaking and core reactivity. The more localized the perturbation, the higher the acceptance frequency, and vice-versa. For example, fresh LP changes impact primarily the gross core leakage and regional power sharings. Thus, fresh LP changes would be expected to have a lower acceptance rate than exposed LP changes, which affect primarily assembly power and, to a lesser extent, core reactivity (due to reactivity carryover and detailed core leakage). Similarly, changes in fresh design determine the gadolinium residual reactivity and exhibit a lower acceptance rate versus changes in orientation, which impacts primarily local power peaking. MCO results are thus shown to be consistent with experience.

3.3 Comparison with Successive, Single Cycle Optimization

SCO cases were executed by running COPERNICUS for each individual fuel cycle in the planning horizon, with the results from Cycle N-1 being used as input to Cycle N. Note that in SCO mode, calculation of levelized fuel cost ($/MWh) for the planning horizon cannot be performed until all cycles have been designed. Thus, levelized fuel cost is replaced by the fresh region fuel cost for SCO (with an estimate of carrying charges).

Figure 3 displays a comparison of MCO and SCO region fuel costs for each cycle in the planning horizon. Also shown is the levelized fuel cost calculated for each strategy. Overall, the MCO levelized fuel cost results are reduced by 1.0% when compared with the SCO results. The fuel cost results for Cycles 3-7 mirror the region size and enrichment loading requirements of the transition cycles. The Cycle 3 initial design fuel cost results for MCO are *higher* when compared to the SCO results. This is followed by significantly lower MCO costs in Cycles 4 with a converging of results in Cycles 5-7.

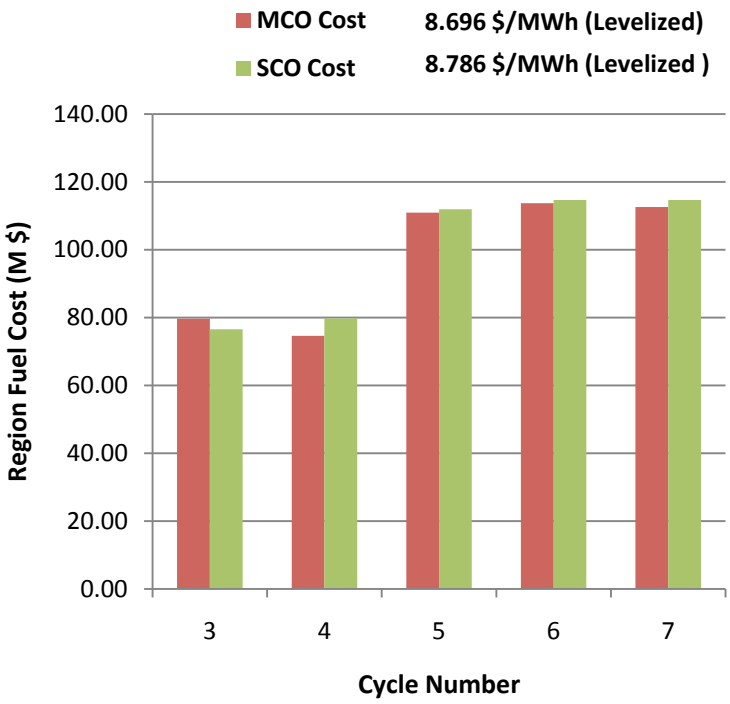

Figure 3. Region Fuel Cost Comparison

Figure 4 displays a comparison of results for region size and average enrichment for MCO and SCO. As shown the MCO region size shows a large reduction in Cycle 4 versus SCO, and is directly responsible for the reduced region fuel cost. Average enrichment behaves quite similarly for both MCO and SCO.

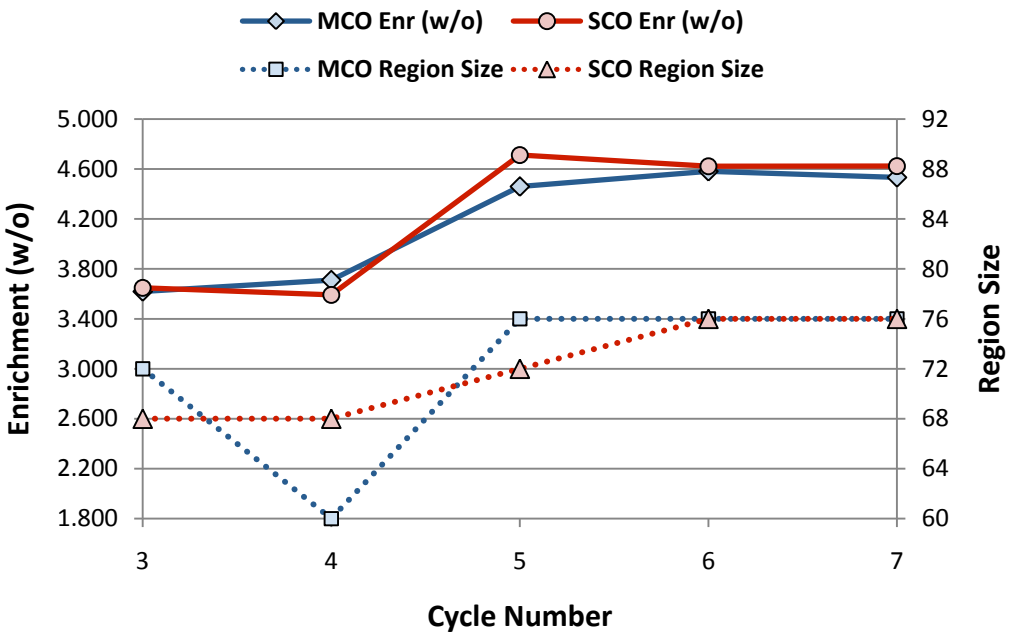

Figure 4. Region Size and Enrichment Comparison

Figure 5 displays the constraint behavior for maximum boron and peak discharge burnup for MCO and SCO. As shown, the constraint limit on maximum boron is challenged in Cycles 5-7. Peak discharge burnup is challenged beginning in Cycle 7.

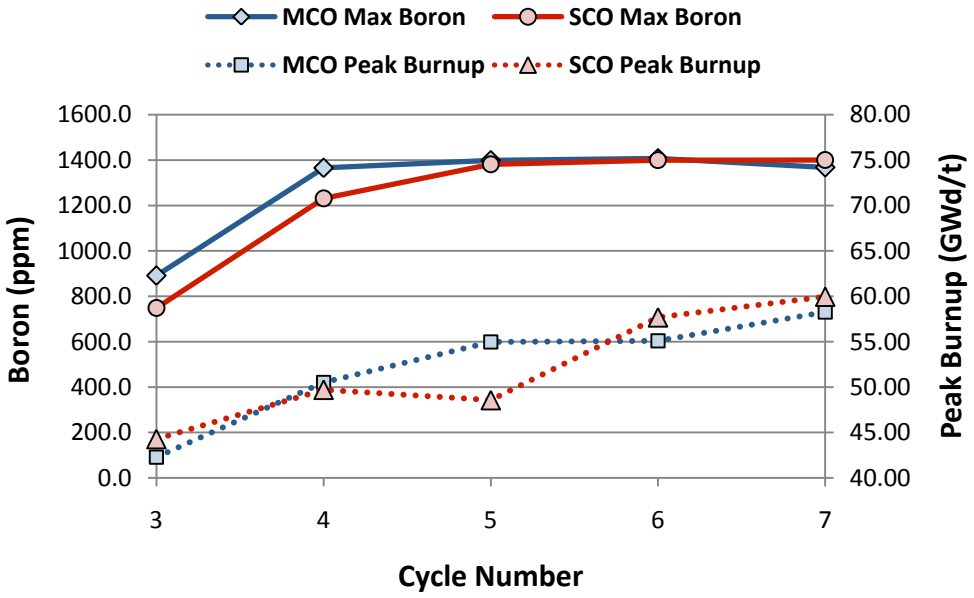

Figure 5. Maximum Boron and Peak Burnup Comparison

4. CONCLUSIONS

MCO incorporating coupled in-core and out-of-core constraints is effective in addressing constraints currently omitted from approaches that employ SCO. As demonstrated for a transition cycle, MCO can reduce levelized fuel cost versus the use of successive SCO. In addition, MCO can address constraints, such as peak discharge burnup, effectively by incorporating the bundle exposure history over the planning horizon as part of the optimization.

The COPERNICUS code, using a licensing grade simulator, is demonstrated to be feasible for use in a production environment without resorting to approximation. This is particularly important since many constraints, especially those limits involving fuel performance, can only be evaluated in the context of a multi-cycle analysis using a 3-D simulator.

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.

Du, Shuang, "Implementation of Genetic Algorithms and Parallel Simulated Annealing in OCEON-P," M.Sc. Thesis, N.C. State University, 2008.

Gropp, W., et al. "MPICH2 User's Guide," Version 1.0.6, Mathematics and Computer Science Division, Argonne National Laboratory, Sept. 14, 2007.

Kirkpatrick, S., Gelatt, C.D. and M. P. Vecchi M.P., "Optimization by Simulated Annealing," *Science* 220, 1983, 671.

Kropaczek, David J., "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, 2008.

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.

Rempe, K.R., et al., "SIMULATE-3 Pin Power Reconstruction: Methodology and Benchmarking," *Nucl. Sci. Eng.*, 103, 1989, 334.