Development of an Application Program for Candidate Analysis and Fine-Tuning of Optimized Loading Patterns

Byeongmun Ahn [a*], Tongkyu Park [a], Sung-Kyun Zee [a]
[a]*FNC Technology Co., Ltd., 13 Heungdeok 1-ro, Giheung-gu, Yongin-si, Gyeonggi-do, 16954, Korea*
[*]*Corresponding author: dinsslug@fnctech.com*

Keywords :reactor core design, visualization, flexible operation, loading pattern optimization, innovative small modular reactor (i-SMR)

1. Introduction

For reactor cores that require flexible operation on a cycle-by-cycle basis, such as the innovative Small Modular Reactor (iSMR), trade-offs between competing design objectives, such as the cycle length and power peaking factors, must be considered when selecting the core loading pattern (LP). An LP optimization code based on the simulated annealing (SA) algorithm generates a set of candidate LPs rather than a single solution; however, examining the optimization history and the candidates solely using text-based outputs is inefficient, and the final selection is still dependent on the designer's engineering judgement.

In previous studies, a support program integrating the optimization code[1] and a related visualization tool for the flexible operation evaluation of the iSMR core[2] were developed. This study extends the support program with result analysis features, including the Pareto front visualization and candidate solution comparison, as well as a fine-tuning editor that combines the automatic optimization with the designer's manual adjustments.

2. Program Features

The program runs on the Windows OS and supports the whole process of core loading pattern (LP) optimization by integrating MOSAIC, a simulated annealing (SA) based LP optimization code, with the ASTRA core analysis code[3]. The program includes input preparation, optimization execution, result analysis, and user-driven fine-tuning all in a single graphical environment.

2.1 Optimization Code Integration and Execution

The program provides editing interfaces for the core analysis code's base input and MOSAIC's optimization input, which define control variables such as batch and shuffle configurations, objective functions with associated weights and evaluation points, and SA parameters. The optimization run is launched and managed within the program, and multiple optimization sessions can be created for each cycle and structured accordingly.

2.2 Optimization Result Analysis

When an optimization run is completed, the program loads and parses the MOSAIC output file to reconstruct the optimization history. The stage-by-stage progress of the SA process, including temperature, samples generated and accepted, and acceptance ratio, is visualized as charts allowing for intuitive examination of the convergence behavior of optimization.

The parsed solutions are organized into two complementary lists. The Top Candidates list ranks solutions based on cost, which is defined as the weighted sum of normalized objective values, whereas the Pareto front list maintains the set of non-dominated solutions over the competing objectives. Since a solution that excels in a single objective may be excluded from the cost-based ranking, the Pareto front retains such solutions while allowing the designer to consider trade-offs between the objectives. The candidates are also displayed on a scatter chart over the objective space, with the Pareto front and top candidates highlighted, as shown in Fig. 1.

In addition, the solution comparison feature compares the core maps of selected candidates side by side in terms of assembly power, assembly burnup, and pin peaking factor (F_{xy}), together with the reference LP taken from the base core analysis input of the optimization, so that the differences among candidates can be identified at the assembly level, as shown in Fig. 2.

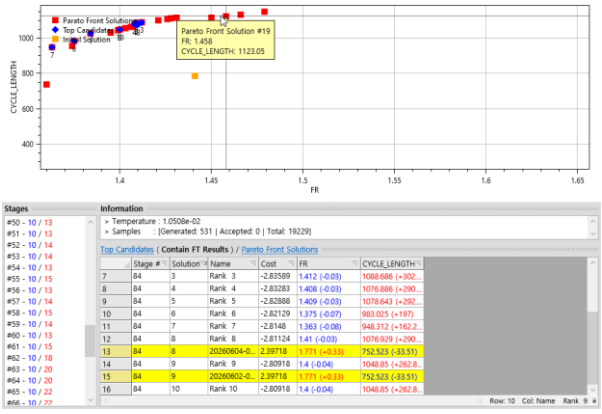


Fig. 1. Scatter chart of the candidate solutions over the two-objective space, in which the Pareto front solutions, the top candidates, and the initial solution are distinguished, together with the candidate list of the session.

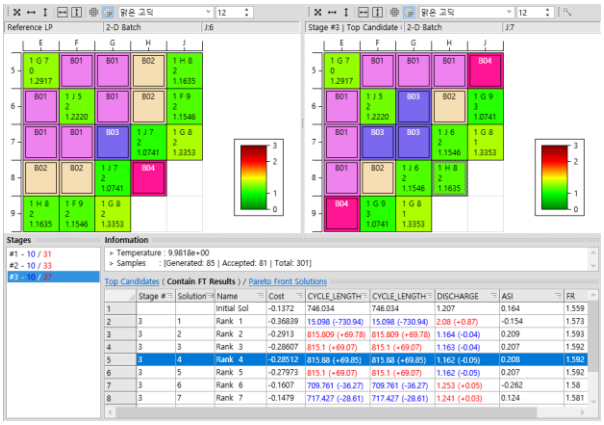


Fig. 2. Comparison of the assembly-wise core maps between the reference LP and the selected candidate solutions.

2.3 Fine-Tuning Editor

The fine-tuning editor allows the designer to manually modify the LP of a selected candidate on an interactive core map. The editor supports position swaps, allocation of fresh and spent assemblies, and rotation of assemblies within the fuel inventory defined in the core analysis input. When an evaluation is requested, the modified LP is calculated by the core analysis code, and the output is processed through the same evaluation module as in the optimization process to generate a solution entry. This ensures that fine-tuned solutions are evaluated with the same fidelity as the automatically generated candidates, and that the saved results are listed and compared together with the optimization candidates. Fig. 3 shows an example of modifying the LP of a candidate solution in the fine-tuning editor, together with the evaluation result of the modified LP.

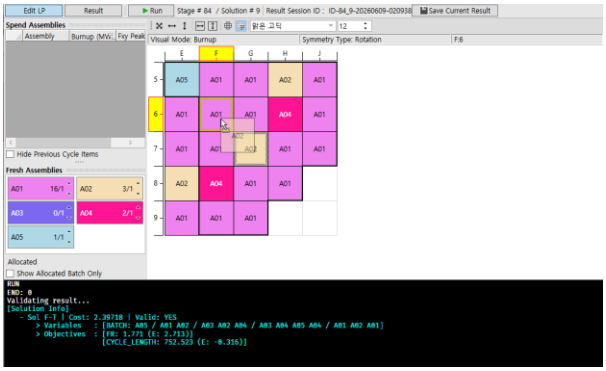


Fig. 3. Example of modifying the LP of a candidate solution in the fine-tuning editor and the evaluation result of the modified LP.

3. Application and Demonstration

To demonstrate the developed features, a sample LP optimization was performed for the initial core of an iSMR core model, and the resulting session was examined within the program. The optimization conditions were set as follows.

- The batch configuration of the initial core was selected as the control variable, with five fresh fuel assembly types under octant symmetry, and two fixed positions as constraints.
- Two equal-weighted objective functions were defined: minimize the maximum pin peaking factor (Fr) and maximize the cycle length.
- The SA process was executed in parallel with 32 threads; the run was terminated after 83 annealing stages, during which approximately 19,000 candidate samples were generated and evaluated in total.

The stage-wise plots demonstrated the typical convergence behavior of the SA process, with the temperature and acceptance ratio gradually decreasing as the stages progressed. The parsed candidates were examined based on the Top Candidates and Pareto front lists; the Pareto front increased from 4 to 23 solutions during the optimization, and the scatter chart over the two objectives revealed the trade-off between the peaking factor and the cycle length at a glance. The differences between the selected candidates were then compared with the reference LP at the assembly level using the solution comparison feature, and a candidate selected from the Pareto front was transferred to the fine-tuning editor, modified, and re-evaluated. This process confirmed that the analysis workflow from the optimization output to the user-adjusted solution works as expected.

4. Conclusions

In this study, the support program for core LP optimization was enhanced with result analysis and fine-tuning capabilities. The program parses the output of the SA-based optimization code and organizes the candidate solutions into cost-based top candidates and the Pareto front, allowing the trade-offs between competing objectives to be examined intuitively together with the stage-wise convergence behavior of the optimization. The fine-tuning editor enables the designer to modify a candidate LP directly on the core map, and the modified LP is evaluated using the same evaluation path as the optimization process, ensuring a consistent comparison of the fine-tuned and automatically generated candidates. The demonstration of a sample LP optimization of the iSMR core verified that the analysis workflow from the optimization output to the user-adjusted solution operates as intended.

The application intends to enable user-defined objective functions and incorporate an AI-based acceleration module. With these extensions, the program is anticipated to be utilized as a practical design tool in setting LPs of reactor cores that require flexible operation, including the iSMR.

Acknowledgements

This work was supported by the Innovative Small Modular Reactor Development Agency grant funded by

the Korea Government (MCEE) (No. RS-2024-00407975).

REFERENCES

[1] B. H. Ha, T. K. Park, and S. K. Zee, Development of Multi-objective Optimization Program for Reactor Core Design Using Simulated Annealing, Transactions of Korean Nuclear Society Autumn Meeting, Changwon, Korea, October 24-25, 2024.
[2] B. M. Ahn, T. K. Park, and S. K. Zee, Development of a 3D Nuclear Reactor Core Design System for iSMR, Transactions of Korean Nuclear Society Autumn Meeting, Changwon, Korea, October 24-25, 2024.
[3] KEPCO Nuclear Fuel Co. (KEPCO NF). (2021). ASTRA User's Manual: KNF-SQA-CDT-19018, Rev.01.