VHTR CORE SHUFFLING ALGORITHM

USING PARTICLE SWARM OPTIMIZATION - ReloPSO-3D

A Thesis

by

SATHISH KUMAR LAKSHMIPATHY

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2012

Major Subject: Nuclear Engineering

VHTR Core Shuffling Algorithm Using Particle Swarm Optimization ReloPSO-3D

VHTR CORE SHUFFLING ALGORITHM

USING PARTICLE SWARM OPTIMIZATION - ReloPSO-3D

A Thesis

by

SATHISH KUMAR LAKSHMIPATHY

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Pavel V. Tsvetkov
Committee Members,	Yassin A. Hassan
	Guy L. Curry
Head of Department,	Yassin A. Hassan

May 2012

Major Subject: Nuclear Engineering

ABSTRACT

VHTR Core Shuffling Algorithm Using Particle Swarm Optimization

ReloPSO-3D. (May 2012)

Sathish Kumar Lakshmipathy, B.E., Anna University

Chair of Advisory Committee: Dr. Pavel V. Tsvetkov

Improving core performance by reshuffling/reloading the fuel blocks within the core is one of the in-core fuel management methods with two major benefits: a possibility to improve core life and increase core safety. VHTR is a hexagonal annular core reactor with reflectors in the center and outside the fuel rings (3-rings). With the block type fuel assemblies, there is an opportunity for muti-dimensional fuel bocks movement within the core during scheduled reactor refueling operations.

As the core is symmetric, by optimizing the shuffle operation of $1/6^{th}$ of the core, the same process can be repeated through the remaining $5/6^{th}$ of the core. VHTR has 170 fuel blocks in the core of which 50 are control rod blocks and are not movable to regular fuel block locations. The reshuffling problem now is to find the best combination of 120 fuel blocks that has a minimized power peaking and/or increased core life under safety constraints among the 120! combinations.

For evaluating each LP during the shuffling, a fitness function that is developed from the parameters affecting the power peaking and core life is required. Calculating the power

peaking at each step using Monte Carlo simulations on a whole core exact geometry model is a time consuming process and not feasible. A parameter is developed from the definitions of reactivity and power peaking factor called the localized reactivity potential that can be estimated for every block movement based on the reaction rates and atom densities of the initial core burnup at the time of shuffling.

The algorithm (ReloPSO) is based on Particle Swarm Optimization algorithm the search process by improving towards the optimum from a set of random LPs based on the fitness function developed with the reactivity potential parameter. The algorithm works as expected and the output obtained has a flatter reactivity profile than the input. The core criticality is found to increase when shuffled closer to end of life. Detailed analysis on the burn runs after shuffling at different time of core operation is required to correlate the estimated and actual values of the reactivity parameter and to optimize the time of shuffle.

DEDICATION

I would like to dedicate this work to my parents, teachers and friends who have all played an important role throughout my life and have helped me reach here. Special thanks to Dr.Pavel Tsvetkov who guided me throughout this effort and encouraged me to carry on. Thanks to my friends and co-workers Mike Leimon, Matt Johnson, Jesse Johns and Ahmad Al Rashdan who helped me to get to the finish line. And finally thanks to my friends Prabhu Selvaraj, Raajesh Khumar, Sai Prasad, Manoj Prasad, Nickalaus Satish and Marie Cuvelier for keeping me sane.

ACKNOWLEDGEMENTS

I would like to convey my deep gratitude to Dr.Pavel Tsvetkov who gave me this opportunity and support to carry out this work successfully. He has always been supportive and helped pick myself up and get on with the work during times of distress. I would also like to thank Dr.Guy Curry and Dr.Yassin Hassan for their interest and guidance without which this work could not have been complete. I would like to thank the professors of the Nuclear Engineering department who have all helped me acquire invaluable subject knowledge and experience.

I would like to thank the members of the Nuclear Engineering department for helping me at various situations and continue my research work without administrative hassles. I would also like to thank the Texas A&M University for this opportunity and financial support.

NOMENCLATURE

3D	3 Dimensional
BOL	Beginning Of Life
DB	Deep Burn
FP	Fission Products
Gen IV	Generation IV
HTGR	High Temperature Gas-cooled Reactor
LEU	Low Enriched Uranium
LWR	Light Water Reactor
MA	Minor Actinides
PSO	Particle Swarm Optimization
Pu	Plutonium
PWR	Pressurized Water Reactor
ReloPSO-3D	Reloading PSO-3D
SNF	Spent Nuclear Fuel
TRISO	Tri-structural Isotropic
TRU	Transuranics
VHTR	Very High Temperature Reactor

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

Nuclear waste management is as important as the nuclear power generation because of the potential short-term and long-term effects on mankind and environment. Compounding these effects are the possible security issues with diversion of fissile material from nuclear wastes that can be converted into weapon grade material. The reactors in US produce about 2000 to 2300 metric tons of spent fuel and are stored onsite. A repository such as the Yucca Mountain can store approximately 63,000 tons of spent fuel, which means that there would be a need for such a geologic repository for every 20 -30 years if an alternative solution were not found [1]. With the abandonment of the Yucca Mountain repository project in 2009, the need for developing more efficient methods for mitigating these issues becomes a very important research objective.

Optimization is a process of applying mathematical techniques to improve efficient utilization of resources resulting in reduction of cost and hence economic gain. In nuclear engineering field specifically, safety takes precedence to a greater extent over cost optimization. With the long-term repository no longer an option in near future, the need for minor actinides destruction becomes important for the nuclear waste management and spent fuel storage. The objective of in-core fuel management is to improve the core performance such as increase in core life and maintain fuel integrity.

This thesis follows the style of *Journal of Nuclear Technology*.

Reactor core reshuffling optimization is one of the methods of in-core fuel management that increases core life or enhances safety or both by identifying an optimal arrangement of fuel (assemblies or blocks or pebbles) within the core. It is a multi-objective, multi modal, Non-deterministic Polynomial time (NP) hard combinatorial problem requiring metaheuristic solution methods. Maximizing cycle length and maximizing TRU incineration are inter related as longer the burnup, higher is the transuranics (TRU) transmutation and destruction.

Reactors are categorized into generations I, II, III, III+ and IV based on the cost effectiveness, reactor safety, non-proliferation, grid compatibility, implementation feasibility and the fuel cycle [2]. The first two generations (Gen I and II) represent the prototype "proof of concept" reactors resulting in civilian adaptation of reactors initially developed as power supply for submarines and the evolution of commercial designs that are economical and reliable, respectively. Gen III reactors are improvements of Gen II reactors with improved thermal efficiency, safety features and fuel material improvements. Gen III+ reactors are Gen III reactors with significant safety system designs improvements. Currently, the reactors that are in operation are either Gen II or Gen III reactors with current Gen III+ reactor construction projects ongoing.

Gen IV reactors include advanced reactors operating in the fast region of the spectrum at high temperature and increased passive safety. Very High Temperature Reactors (VHTR) is a type of Gen IV thermal reactor with helium coolant and output temperature around 950^0C. Deep Burn (DB) is the process of comprehensive nuclear waste

destruction in the spent nuclear fuel by irradiation. By optimizing the reshuffling process of the fuel blocks within the core of VHTR, the waste destruction can be increased if the core life can be increased. Cost of fuel per reload (one-third of the core) for a 1000 MWe LWR is around $40 million for a 18-24 months cycle which includes cost of uranium purchase, conversion, enrichment and fabrication along transportation, storage and salvage [1]. If optimization increases core life, then the cost incurred due to reload can be reduced and as a consequence any volatility in the fuel cost can be prevented.

Next Generation Nuclear Plant (NGNP) program is undertaken by Department of Energy (DOE) to develop reactor design that would be the basis for high temperature gas cooled reactors where the additional heat is utilized to generate hydrogen for industrial and transportation energy requirements thereby reducing dependence on conventional energy sources. The VHTR model is considered for the NGNP program for creating benchmark for commercialization of high temperature reactors.

I.A. GEN-IV REACTORS

Gen-IV reactor technology consists of six reactors as per the developed roadmap for advanced reactors of which three are fast reactors and three are thermal reactors with different types of fuel cycles. Table1 gives the different types of reactors decided for Gen-IV. The goals of the Gen IV designs are sustainability, economics, safety, reliability and proliferation resistance [2]. Sustainability implies efficient utilization of uranium resource and minimal waste production. Operating in the fast spectrum improves fuel utilization and reduced used fuel. Economics include economics of design, construction,

operation and maintenance. Cogeneration of hydrogen with electricity is an added advantage. Since Gen IV reactors are based on innovative technology, licensing is a time consuming and expensive process and hence NGNP program is implemented with stress on passive safety and reliability. Non-proliferation property of the reactor is imparted by developing efficient in-core fuel management systems.

The Gen IV reactors are being designed with closed fuel cycle and operation in fast spectrum region to deplete fertile and fissile material in the spent fuel. On-site fuel cycle operations reprocess the Pu-239 obtained from the fast reactor acting as a "breeder" with U-238 and reused in the reactor. The final waste quantity will be very less and have shorter half-lives with lower radiotoxicity compared to the waste from LWRs. Though VHTR is a thermal reactor, waste management is possible because of its high operating temperature.

The NGNP program is the short-term initiative conceptualized towards the Gen IV reactors. Knowledge gained over the years about HTGRs is utilized for the development of the NGNP design. Gas Turbine Modular Helium Reactor initially developed for using weapon grade plutonium by General Atomics together with MINATOM (Ministry for Atomic Energy of Russian Federation), is the basis for VHTR design. GT-MHR has prismatic block type fuel with TRISO particles and an annular core, which are derived from FSV (Fort St.Vrain). The feasibility of HTGRs has been proved through the successful operation of reactors ranging from dragon Reactor (UK), Peach Bottom to the German Thorium High Temperature Reactors (THTR). It provides the confidence on the

fuel structure, core physics and adopted from GT-MHR. Hence it gives an ideal technology for near term deployment plan (NGNP).

Table 1 Gen-IV reactors [2]

Reactor type	Spectrum	Fuel Cycle	Temp. ^{0}C	Applications	R&D
Gas Cooled Fast Reactor (GFR)	Fast	Closed	850	Electricity, Actinide management, hydrogen	Fuels, Materials, Safety
Lead-Alloy fast reactor (LFR)	Fast	Closed	480 - 800	Electricity, Actinide management, hydrogen	Fuels, Materials compatibility
Sodium Fast Reactor	Fast	Closed	550	Electricity, Actinide management	Advanced Recycle
Very High Temperature Reactor (VHTR)	Thermal	Open	900 -1000	Electricity, Hydrogen, Process Heat	Fuels, Materials, Hydrogen production
Supercritical Water Reactor (SCWR)	Thermal, Fast	Thermal - Open, Fast - Closed	510 - 625	Electricity	Materials. Safety
Molten Salt Reactor (MSR)	Thermal	Closed	750 - 1000	Electricity, Actinide management, hydrogen	Fuel, Fuel treatment, Materials, safety and Reliability

GFRs are similar to VHTRs except that they operate in fast neutron spectrum and closed fuel cycle mainly designed for actinide management. LFRs are reactors that can range from very large size to small battery sized fast reactors. They have high reliability and proliferation resistance because of their longer life and closed fuel cycle. MSRs operate at the thermal and epithermal regions. The fuel is a liquid mixture of sodium, zirconium, uranium fluorides which is also the coolant. The heat transfer capability of the molten salt is high, requires no time for refuel and gives high waste burnup. SFRs give the greatest sustainability among the Gen IV reactors and its primary goal is actinide management. SWCR reactor with no phase change in coolant during heat transfer gives the largest power rating and the maximum efficiency of 44% among the Gen IV reactors.

VHTR is a once through graphite-moderated, helium-cooled reactor with an expected output temperature of 1000^0C. It uses a direct Brayton cycle for electricity generation and indirect cycle for hydrogen production. The fuel configuration maybe prismatic (block-type) as it was in the Fort St.Vrain design of the HTGR or pebble-bed type as in the South Africa's PBMR. The GT-MHR based VHTR design has hexagonal fuel and moderator blocks forming an annular core with reflector blocks at the center and around the fuel blocks (Fig.1). Beside the replaceable graphite elements, it also has a permanent side graphite reflector. VHTR fuel composition is similar to the HTGR fuel used in the Fort St.Vrain reactor. VHTR uses Tri-isotropic coated fuel particles (TRISO CFP) that provide structural integrity till 1600^0C, protection from fission gases and stresses from swelling. Though the core is ceramic, the high operating temperature may affect the other non-ceramic components. The low power density of the VHTR core due to low

volume fraction of fissile material and the high negative temperature coefficient of reactivity give thermal stability and reactivity control.

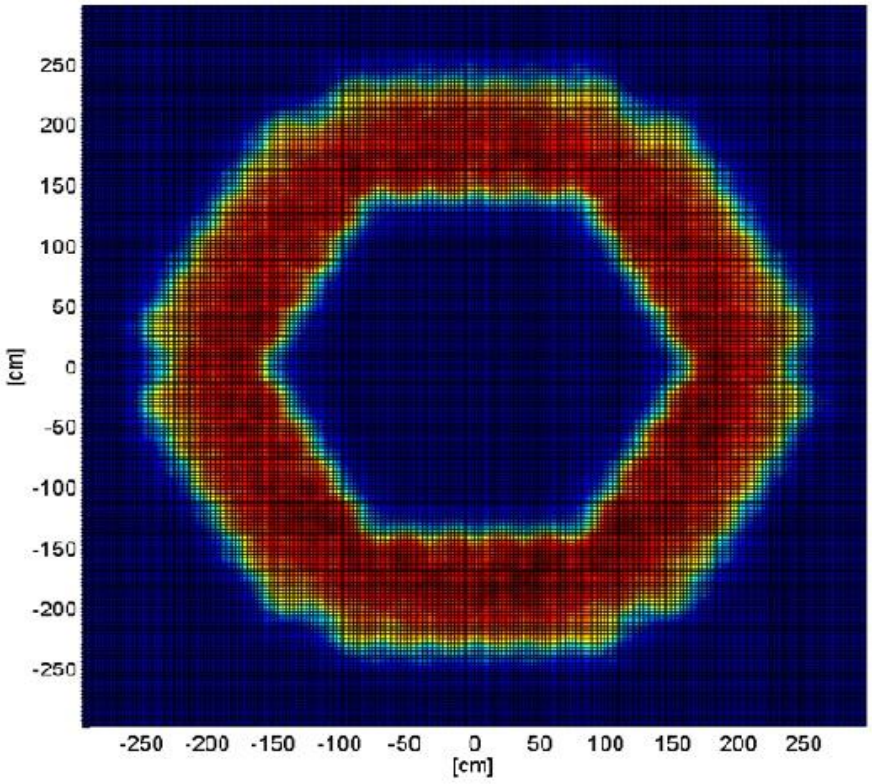


Fig.1 Prismatic annular core showing the fast flux distribution

There are two types of fuel blocks within the VHTR – fuel blocks with control rod provision and fuel blocks without them. The NGNP model has a 3-ring fuel arrangement stacked in 10 layers with inner and outer reflector blocks. Each block consists of cylindrical compacts that have TRISO particles embedded in graphite matrix stacked within a graphite block. Fig.1 shows that some of the fuel blocks have different exposure

to flux and hence power peaks within the reactor core. 3D shuffling means that the fuel block from any location can be moved to any other location within the core for reducing the power peaking. The fuel blocks with control rod can be moved only to locations that should have control rod material in them during shuffling process. In the LWRs, the fuel blocks are stacked and arranged into an assembly and hence the fuel assemblies can be moved radially only.

Both the pebble bed and prismatic arrangement allow fuel to be moved in three dimensions unlike LWRs where the fuel elements are shuffled in two dimensions. Three-dimensional reloading provides an opportunity for better core reloading and hence a chance for increasing profit by means of increase in core life. Three-dimensional shuffling of fuel blocks is better in the sense that a better mixture of blocks with higher and lower reactivity within the core.

The Fort St.Vrain VHTR design is the first of the US commercial reactors o be decommissioned due to various technological reasons that lead to economic infeasibility. A near term solution is sought after for nuclear waste management because of the current rise in the importance of spent nuclear fuel storage as an economic problem (the federal government had been collecting payments from utilities for building a repository to accept SNF by1998 or the government is liable to pay settlement to the private utilities holding the spent fuel as per the Nuclear Waste Policy Act [3]). This has increased the importance on developing VHTR with an added interest as a TRU burner. A fully functional reactor is to be constructed at the Idaho National Engineering and

Environmental Laboratory acting as a pilot for commercial and future VHTR developments.

I.B. DEEP BURN VHTR (DB-VHTR)

Deep burn concept of VHTR is developed mainly for transuranics destruction using thermal neutrons for high levels of burnup of spent fuel from LWRs. It is aimed at achieving destruction of almost entire proliferation material and roughly 90% of TRU content within the LWR used fuel. Deep Burn VHTR (DB-VHTR) is a type of high temperature reactor which uses block type fuel that could achieve a burnup up to 60%-70% of FIMA (fission per initial metal atom) by single pass, multi-cycle irradiation in VHTRs resulting in reduction of spent fuel TRUs (transuranics) in the inventory. The spent fuel from DB-VHTRs can either be stored in repository directly or can be reprocessed as fuel for fast reactors. By destroying up to 90% of the TRU in the fuel, the amount of fuel to be stored is reduced to a smaller fraction of the original high radioactive spent fuel [4].

Partition and transmutation of the Spent Nuclear Fuel (SNF) reduces the time required for the emission levels of natural uranium (~1000 years) compared to the unprocessed SNF (~1000000 years). Transmutation and destruction of Americium, Curium, Neptunium, which form the minor actinides (MA), are essential for long-term storage of SNF. Fission Products (FPs) are important sources of short-term decay heat, toxicity and radioactivity release from SNF. The SNF isotopes that are fissionable by thermal neutrons generate the neutrons for converting the thermally non-fissile part of the fuel

into fissile isotopes. This helps in extensive TRU burnup and underlines the importance of safe fuel structures required [4].

An important feature of the DB-VHTR is the use of Tristructural Isotropic (TRISO) fuel particles. With increased inherent safety due to the fuel particle structure, knowledge gained about the TRISO particles is one of the biggest outcomes of the research in VHTRs where the fuel core is surrounded by layers of Pyrolitic Carbon and Silicon Carbide increasing proliferation related safety issues and providing deep burn.

The annular DB-VHTR core is safe under transient conditions. Three characteristics are seen in VHTR that are of importance from safety point of view. They are high power peaking at the interface between the fuel and the reflectors, change in power distribution for a small change in control rod position, and very low temperature coefficients for a temperature upto $400^{o}C$ due to the presence of Pu.

To increase the burnup for high TRU transmutation and destruction, effective incineration is required which in turn emphasizes the requirement of an efficient reloading and reshuffling process. Incineration carried for a period of 5 to 6 years requires a well-developed algorithm encompassing the safety constraints and using the features of the DB-VHTR design.

I.C. RESEARCH OBJECTIVE

Objective of an optimization problem is expressed as a function of parameters that are representative of the problem. In core reloading optimization problem, the objective

function and is based on the core performance parameters such as collision rate, atom density, etc. The objective chosen is to reduce localized power peaking in the core thereby improving core safety and inherently increase core life. A method for tracking the fuel blocks during each shuffle must be developed to work with the algorithm.

Power peaking minimization is an important objective as the power spiking might occur at different locations within the core. Non-uniform power distribution or power peaking within the core produces temperature gradient and affects mass flow rate and stress on the fuel affecting fuel integrity. The process of optimization needs a set of parameters that effectively represent these objective functions in terms of measurable quantities from either simulation or actual reactor data and that could be processed further. Developing such a quantity to work with the various steps of the optimization reflecting the changes the core is predicted to undergo during the search process is necessary. A measurement of reactivity (called localized reactivity potential) of each fuel block is computed from the measured values of reaction rates from the MCNPX simulations. The reactivity potential is directly related to the power peaking and hence by minimizing the reactivity potential, power peaking can be minimized. Reactivity potential is used instead of the power peaking because during the shuffling algorithm, for every new loading pattern considered, the value of the reactivity parameter can be estimated by a relation derived based on the definition of reactivity rather than simulating the loading pattern using MCNPX. The second reason is that by extending core life can be done by optimizing the modified reactivity potential. Hence the reactivity potential provides the linking parameter connecting reducing power peaking and increasing core life.

The objective of this thesis is developing a core shuffling algorithm for VHTR. Realizing the optimization solution includes the following tasks:

(i) Identifying the parameter that can be calculated or estimated for different loading patterns without having to run MCNPX burn runs and develop the methodology to calculate or estimate the parameter

(ii) Identifying a suitable algorithm for the problem that is flexible to work with the parameter developed

(iii) Develop the code and validate the algorithm. Repeat the processes (ii) and (iii) until a suitable algorithm is identified

(iv) Apply the algorithm for the problem and analyze the results

For shuffling in LWRs, many optimization techniques ranging from multi-objective non-linear techniques to multi-objective evolutionary algorithms have been used. The difference is the type of shuffling, that is, in LWRs the shuffle is radial only due to the nature of the power peaks and the fuel structure. Due to the prismatic structure of VHTR annular core that has axial layers and radial rings with preferential hotspots as shown in the Fig.2, a method with more flexibility are required. Since the fuel blocks are arranged in layers and each fuel block is separate physically rather than as a fuel pin, there is an advantage in considering the problem in 3 dimensions and performing the optimization. On the other hand, this increases the problem complexity exponentially and hence a

metaheuristic solution method is required. The method chosen is called Particle Swarm Optimization.

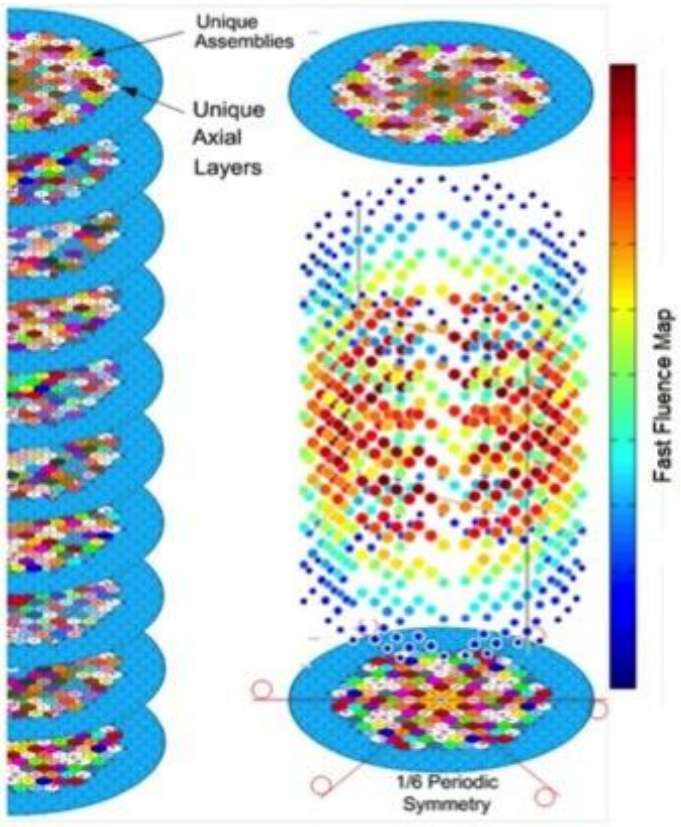


Fig.2 Whole core representation with fluence map

Particle Swarm Optimization is a stochastic optimization technique in which the search begins with a set of random solutions and each solution is improved during each of the iterations of the algorithm till the end of a predetermined number of iterations. The algorithm is run multiple times by changing algorithm parameters to get a set of optimal solutions from which the best solution can be chosen. Final project output includes

computer code for shuffling methodology with relational interfaces to physics databases and Monte Carlo codes that models whole core simulation and shuffling process.

I.D. THESIS OUTLINE

The thesis explains some concurrent optimization methods used in reactor core shuffling problem and the implementation of a suitable algorithm for 3D core shuffling of VHTR. Chapter 1 gives the general introduction on the VHTR reactor and an overall review of the shuffling problem. Chapter 2 explains the two other approaches that were tried before the eventual Particle Swarm Optimization (PSO) algorithm is found to be suitable. Chapter 3 explains why PSO is chosen over the other algorithms that are used for similar combinatorial problems. Tabu search and simulated annealing, gives a general overview of the algorithm and then describes the integration with the Physics based optimization and the corresponding data flow.

The PSO algorithm is explained in general and then its adaptation to the current problem is discussed in Chapter 4. The fuel block tracking methodology is also explained in. In Chapter 5, the developed test cases are described demonstrating the algorithm performance along with full core simulation results establishing the algorithm effectiveness. Chapter 6 provides conclusions and the recommendations. The appendices consist of detailed code description, input file to the algorithm, output interpretation and VHTR design specifications.

CHAPTER II

VHTR RELOADING/RESHUFFLING OPTIMIZATION METHODOLOGY

II.A. THE NEED FOR OPTIMIZATION

The optimization of reactor core reloading process arises from the very basic function of the VHTRs, which is, maximizing burnup of the fuel blocks burnt already. Reloading and reshuffling are two processes that can improve core life. Reloading is the process of removing the fuel blocks that cannot help sustain the reactor operation based on the fluence level and replacing them with fresh fuel blocks. Reshuffling is the process of shuffling the fuel blocks at pre-defined instances of core operation such that the process brings out a better core life than if it would have operated with fuel blocks at their original locations. With increase in core life, the economic gains and by maximizing burnup, the amount of radioactive material to be stored is reduced. Economic gain and safety are the main reasons for the need of optimization.

Implementing efficient shuffling and reloading process during a core life of 40 years can achieve reduction of fuel costs up to $35 million [1]. As the major fuel cost is fuel fabrication, by utilizing the fuel already present within the core for a longer period, the need for new fuel can be reduced. It also reduces the cost of petroleum required in the fuel fabrication process. If blocks are not shuffled axially, the high fluence levels due to exposure to continuous irradiation temperatures may compromise their structural integrity. By developing an effective shuffling process, the graphite waste from the used

fuel stream can be reduced up to 106 m^3 during a core life of 35 years and the fissionable material left in the fuel can be utilized [1].

The need for an efficient optimization of VHTR reloading process can be established from three points of interest such as the design structure of the VHTR, the reload cycle and the size of the optimization problem. The size of the optimization problem is one of the key reasons because it is found that random shuffle procedures could give a better core performance and to find a good solution from a large number of possible combinations, a systematic search algorithm is needed.

The VHTR core is a hexagonal prismatic block type core derived from the GT-MHR reference design [5]. The fuel blocks are stacked vertically and doweled as columns. This fuel block type design provides the flexibility to move the blocks in 3D rather than the fuel block type movement seen in the PWRs [6]. This increases dimensionality of fuel movement within the core during optimization and hence the need for a novel robust optimization technique. Due to the core arrangement, there will be locations with differential reaction rates and are called hotspots which are safety concerns. The hotspots are the locations in the core with power spike. They are safety concerns because the increase in temperature due to the increase in power at a particular location may cause thermal stresses. Preferential heating may cause increase in void fraction in the coolant and hence decrease performance.

The reloading cycle of the VHTR is 5-6 years with shuffling every 18 months. This in-core management approach increases the complexity of the optimization problem as it

increases the number of combinations of the core arrangement to be evaluated. As shuffling is a recurring process during the core lifetime, the number of combinations of the core configurations to be evaluated increases and hence requiring a formal search process. The core lifetime can be efficiently increased by choosing the optimal arrangement of the fuel blocks from all the possible combinations of the fuel blocks present within the core. If reloading is included, it improves the process as more fuel is brought into the core.

The hexagonal arrangement of the core means that optimization of 1/6th of the core can be repeated for each of the other $5/6^{th}$ of the core. For a typical VHTR, there are 10 layers of fuel blocks ranging from 3 rings to 8 rings of fuel and hence a total of 120 to 190 fuel blocks per $1/6^{th}$ of the core. And so the number of possible arrangements gives 120! to 190! respectively. Evaluation of each of the arrangement is impossible and hence an evolutionary optimization technique is needed. If the total life cycle of the fuel is 6 years with 4 shutdowns this number of combinations to be tested raises to $120!^4$ to $190!^4$ respectively. The deterministic codes even with a runtime of 2-5 seconds per evaluation cannot process each of the combinations in appreciable time.

II.B VHTR RELOADING/RESHUFFLING PROBLEM

The optimization search is to find the arrangement of the core with minimum power peaking, maximum TRU burnup and maximum cycle length and hence a multi-objective optimization problem [6]. Non-deterministic polynomial-time (NP) problem stands for the decision problem for which there exists a solution that can be verified using a non-

deterministic algorithm in finite computational time by a solution machine. Non-deterministic algorithm is one that has more than one resulting state or step for one particular situation and has a second part of it, which accepts one of the possible options after evaluating all of them. Polynomial time describes the complexity of the running time of the solution algorithm, which has a running time with an upper bound that is described by a polynomial of the size of the input to the algorithm. NP hard problems are those that are as hard as the hardest NP complete problems and can be a decision or optimization problem. The set of NP hard problems that have solutions that can be verified in finite computational time are called NP complete problems. VHTR reloading is one such problem because of the huge number of combinations that cannot be searched through exhaustively and hence adopts a more heuristic approach.

As the problem works on a set of possible combinations of solutions that cannot be searched exhaustively, this is a combinatorial optimization problem. The solutions are discrete or made as discrete solutions. A multi modal problem is one that does not have a single solution. It may have a set of solutions each of which maybe optimal under different boundary conditions or limitations. Reloading problem is one such problem as it may have multiple arrangements that may satisfy the required performance and safety conditions but may have a compromising attribute under different conditions. The solution set is hence a pareto-optimal front. Pareto front defines the set of possible outputs from which a suitable solution is identified. Since the solutions may not be most optimal; it is more of a weak pareto set.

Heuristics are experience-based techniques that are used for improving the search speed of combinatorial problems where extensive search is not possible. Metaheuristics is the method of finding solution to huge combinatorial problems by picking up an arbitrary solution and improve the solution at every step to get as close to an optimum solution as possible using knowledge of the system and the problem. It describes the method of finding a solution through iterative steps. The heuristics involved to solve the problem is based on the experience and knowledge gained from the neutronics analysis of the reactor and hence it fits the description of the metaheuristic solution.

For the reshuffling problem, there are a large number of combinations of the fuel blocks, for example in the 5-ring core configuration VHTR, there are 190! combinations. The optimization can be done one or more of the parameters such as the power peaking or core lifetime. It is known that changes in the core flux due to one single swap (swapping any two blocks only) are insignificant when the whole core is performance is considered. Hence it is possible for more than one combination (arrangement of core) can give similar core performance. This, along with the large number of combinations may result in multimodality. For the same reason, the solution approach is iterative, starting with a set of random solutions and improving it towards an optimal point, thereby requiring a metaheuristic approach.

Hence the nature of the problem from the discussions so far is that the shuffling optimization of the VHTR is a multi objective, NP hard, combinatorial, multi modal problem and requires a metaheuristic solution method. This shows the complexity of the

solution space and hence the difference in complexity from the LWR reloading problems.

II.C. TABU SEARCH AND SIMULATED ANNEALING

Optimization of reactor core reloading has so far been done using various optimization algorithms ranging from multi-objective non-linear programming to modern day evolutionary algorithms. As discussed above the use of evolutionary algorithms is rather unavoidable as the computing ability increased and hence the attempt to solve problems by performing a wider search towards optimum for such huge problems gathered efficiency. Two such methods that are developed in 1970's are tabu search and simulated annealing. This section deals with the steps involved and demonstrates the reason for not choosing either of them, which in turn provided further insight towards solving this problem.

Tabu search is an engineer-designed approach with no actual convergence proof [7]. It works well for many combinatorial problems. Tabu search can be looked as an advanced descent method of minimization or maximization. The algorithm is popular for combinatorial optimization problems because it is flexible in its use of the problem specific fitness function and the usage of memory that increases the possibility to cover more solution space. The following steps show the steps involved in tabu search in simple algorithm as given by [7]. Let a combinatorial problem consist of ζ solutions and the global optimum $i \epsilon \zeta$. An arbitrary solution is selected as i^* from ζ . Let k be the number of predefined iterations. Let $N(i,k)$ be the size of the neighborhood for each

feasible solution *i* during iteration *k.* Tabu search is a memory based technique that saves not just the previous best solution but also the itinerary through which each solution is achieved so as to move from local optimum and towards other regions of the solution space. Two types of memory, short-term memory and long term memory are used. Short-term memory is called the recency memory that saves the moves for the last $|T|$ moves that lead to local optimal solutions so that those solutions should not be visited again thereby moving the search to other areas. It is a restriction on the search because when the set exceeds the predetermined size of T and so multiple lists are used at the same time.

But developing such a set of solutions is uncontrollable over each i per iteration and hence each move that lead to a solution j is saved as *M(i)*, defining $j = i \oplus m$ and so neighborhood definition for the problem is:

$$N(i) = \{ j \mid \exists m \in M(i) \text{ with } j = i \oplus m \} \tag{2.1}$$

Every time a move is given a tabu status $t_r(i,m)$, the list is updated. r is just the number of tabu moves. When a new solution is found that is better than the current best solution, though it might be from the tabu list $|T|$, it might be considered and its neighborhood is analyzed. This set is called the aspiration level set. This is updated throughout the optimization process and hence known as the long-term memory. Long-term memory is called the aspirancy memory as it is used to either intensify search in certain neighborhood *N(i,k)* where solutions appear better by penalizing the solutions that are far from it or diversifying the search by penalizing the current solution region. These are

called the aspirancy conditions $a_r(i,m) \in A_r$. Here, r is the number of aspiration moves. These are the moves that are forbidden in the short-term memory but are included in the search due to their performance above a fixed aspiration limit. These techniques are the most important and attractive feature of tabu search as they help in reducing the neighborhood set or in other words choose the subset V^* on which the evaluation is done at each iteration and it varies for each iteration. Hence tabu search is also part of a class of methods called dynamic neighborhood search technique. f is the fitness function that is used to evaluate the solutions in the solution set ζ. As tabu search can use its own heuristics rules that modify the fitness function, it is a part of metaheuristic solution methods. The modification is the addition of diversification and intensification factors to the basic objective function. The algorithm steps are:

(i) Set $i^* = i$ and $k = 0$

(ii) Set $k = k+1$ and form a subset V^* solutions of the neighborhood of $N(i,k)$

(iii) Choose a best $j \in V^*$ with satisfying f or the modified fitness function $\tilde{f}$ and now $i = j$

(iv) Now $i^* = i$ if $f(i) \leq f(i^*)$

(v) Update tabu conditions and aspiration conditions

(vi) Stop algorithm at stopping condition else go to step (ii)

Stopping condition may be the exhaustion of the search space or if the maximum number of iterations reached or when there is no improvement on i^* for a predefined number of iterations or if the solution obtained is determined to be the optimal. The entire section on tabu search is based on [7].

Simulated annealing method is given its name, as it is a simulation of annealing process in which the metal placed in a heat bath is heated rapidly to a high temperature and cooled slowly so that the particles align themselves at the lowest energy levels giving the metal by heating the liquid and cooling it respectively. It is similar to tabu search except for the methods used for choosing V^* but the difference is that solutions that do not satisfy the fitness criterion are accepted with a probability derived from the Boltzmann distribution. This constant concentrates on the states with lowest energy and hence finally only those levels have non-zero probability for the material particles to settle. Simulated annealing belongs to a class of optimization solutions called threshold techniques and not the iterative improvement methods, it searches the neighborhood of a solution and compares the cost of each solution to the threshold level set and selects if the new solution is better than the current solution, if not the solution is accepted based on a criterion called the Metropolis criterion. After continuing this process for a number of iterations, the optimal point is reached (thermal equilibrium in terms of the simulated annealing process of metals).

For example, let there be a combinatorial problem (ζ,f) where the optimal is $i^* \in \zeta$ based on the cost function $f : \zeta \rightarrow \Re$. The neighborhood function for the problem is defined by N: $\zeta \rightarrow P(\zeta)$ and it maps every $i \in \zeta$ a set $N(i) \subseteq \zeta$ of neighboring solutions. From the neighborhood, different configurations are obtained for the current step and the best solution in the neighborhood is compared to the previous step best. This is analogous to getting the system to equilibrium.

The probability for the configuration to be a solution is given by the interpretation of the Boltzmann's distribution:

$$P(configuration = i) = \frac{1}{Q(c)}.\exp(-\frac{C(i)}{c}) \tag{2.2}$$

$Q(c)$ is called the partition parameter that depends on *c.* The evaluation of cost (analogous to the energy of the system) of the neighborhood best solution and the control parameter *c* in the above equation (analogous to the temperature) gives the measure for acceptance of the new solution. The perturbation is calculated as the difference between the cost before and after perturbation.

$$\Delta C_{ij} = C(j) - C(i) \tag{2.3}$$

If the above equation gives the perturbation, then the probability that j is the next better configuration is given by 1, if $\Delta C_{ij} \leq 0$ and it is given by the *Metropolis criterion*

$\exp(-\frac{C(i)}{c})$ if $\Delta C_{ij} > 0$. This is continued till the equilibrium is achieved. The control parameter *c* is initiated with a high value and is decreased during the course of the iteration for each step. Configurations around the current solution are evaluated and so on. This is done till the value of c can no more be reduced (algorithm stop condition).

Important feature of the algorithm is that the acceptance criterion for perturbations greater than 0, which is the threshold condition, is obtained randomly. The random variable X generating the threshold follows a distribution function *F*. It is chosen such that for each current solution *i*, expected value, E[X] = *i* for each step. Hence the

probability of X being at most $y \in \Re$ is given by P ($X \leq y$) = $F_i(y)$ and so a positive probability is given for values that greater than the current solution. The function F is chosen such that for solutions involving great cost increases, the probability of acceptance is lower and solutions with smaller increase are accepted with a higher probability.

Simulated annealing can also be seen as a sequence of Markov chains, which means that the current state depends on the previous state only. Here, state *j* is obtained only from a transition of state *i* during an chain and hence the whole algorithm consists of a number of Markov chains. The formation of states depends on two probabilities as discussed above as the randomly formed state in the neighborhood $N(i)$ which is accepted based on the acceptance criteria defined by the *Metropolis criteria*.

If the probability that a state is formed does not depend on the trial, then they are homogeneous Markov chains and the corresponding transition matrix is given by:

$$P_{ij}(c) = \begin{cases} G_{ij}(c).A_{ij}(c) \forall j \neq i \\ 1 - \sum_{i=1, l \neq i}^{|\mathbb{R}|} G_{il}(c) A_{il}(c) \; j = i \end{cases} \tag{2.4}$$

where $G_{ij}(c)$ is the generation probability and $A_{ij}(c)$ is the acceptance probability and they are dependent on the control parameter *c*. This gives the stochastic property for the simulated annealing algorithm as $\sum_j P_{ij}(c) = 1$ and the actual terminating condition is $P_{ij}(c) = 1$ and $\lim_{l \to \infty} c_l = 0$. The section on simulated annealing is based on [8].

II.D. OPTIMIZATION METHOD SELECTION

The main differences between tabu search and simulated annealing are the stochasticity of simulated annealing which gives its asymptotic convergence a proof, the memory based optimization of tabu search in which the next configuration depends on the previous history till then as opposed to the Markov chains of simulated annealing and the choice of subset solution spaces that is done so well in the tabu search. Both methods have their advantages and disadvantages but the choice of algorithm depends on the complexity of the combinatorial problem and that of the algorithm.

Reactor core reloading problems so far have been solved using various methods as mentioned before in including tabu search and simulated annealing. But for VHTR core reloading optimization, particle swarm optimization method is chosen. Tabu search has an inherent problem that it has no actual proof of asymptotic convergence, the high level complexity of the algorithm in itself and the large manipulation of memory that increases programming complexity. Also, modeling the problem is of high importance, as any tuning of the parameters involved will not improve the quality of the search. Another important reason for not choosing tabu search is that it is found to be working fine for some problem and not for other problems, since it does not have a convergence proof the actual reason for this behavior cannot be identified.

Simulated annealing on the other hand has its drawbacks in terms of its large time consumption and the cooling schedule that is the rate at which transition occurs from one step to other using the Metropolis criterion. Faster cooling may result in bad

convergence, which is analogous to materials cooling to metastable amorphous structure. The choice of methodology for carrying out this process is highly experience oriented and there are no strong rules. It has a lot of variations in itself, which are either proposed or borne out of experience, and hence does not help in developing a complete model. Simulated annealing also has the *performance ambivalence* that is a bad feature for any optimization algorithm. It seems to work well for some problems and fail badly for some. Though it seems to work for relatively a large variety of problems, its main advantage is its simplicity and the stochasticity built into it.

Considering the above-mentioned factors, choice of algorithm in itself is a big task for such a problem. As already mentioned, reloading optimization of DB-VHR is more complex than the usual reloading optimization of LWRs. It should be history based so as to reduce circulating back to the same optimal point. The parameters of the algorithm should be easily designable, in the sense that they could be easily obtained by running the code rather than experience. This is an important reason for choosing Particle Swarm Optimization. Particle Swarm Intelligence gives a high level of independence with respect to the system parameters. This is a major decision factor in the choice of the algorithm for reloading problems because the burnup calculations that are required for evaluating the loading patterns are very long. Hence a derived parameter, which reflects the core behavior and that can be estimated when fuel blocks are shuffled, is used in this problem. So the algorithm should have the flexibility to operate with such parameter constraints. It is simple in its structure and could easily be implemented as a computer code. Recent efforts by [9] have shown that using PSO search was faster and the results

are achieved with a wider search of the search space. Particle Swarm Optimization technique is explained in the next Chapter in context to the reloading optimization of VHTR.

CHAPTER III

OPTIMIZATION PROCESS ARCHITECTURE

III.A. OPTIMIZATION PROCESS

One of the fuel management objectives is the process of extending burnup by implementing appropriate loading schedules and is usually achieved by shuffling and reloading of fuel blocks. There are many general shuffling methods based on physics based inference such as by bringing in blocks with more fissile material left in them towards the center of the core and bringing the more burnt fuel blocks towards the periphery of the core. The reversal of the process may also be followed. Other types of management are bidirectional movement, zonal loading, graded and batch irradiation [10]. These radial shuffling methods do not utilize the blocks at the top and bottom of the core that are exposed to lesser irradiation than the ones at the center of the core. The Particle Swarm Optimization technique considers a wider search space meaning that it includes radial and axial shuffle processes and hence evaluates loading patterns that are not part of the axial or radial only shuffle methods [6].

For optimization, the core is looked as a set of domains including performance, applications and design. The design domain is further divided into geometry, non-fuel materials and the BOL fuel. Within the material domain there is the static domain, which comprises of the materials that do not change during operation and the dynamic domain that consists of the materials that change property during reactor operation. For

developing a reloading optimization algorithm, due consideration should be given to the location of the fuel blocks and their composition.

It is found that two-fuel system has better destruction rate with more than 50% during a fuel life of 12 years. The study of the two fuel system neutronics has shown that there exists a tradeoff between destruction rate and the core lifetime. The performance analysis by plotting the kernel size and mass loading against the excess localized reactivity potential for a two-fuel system showed that shorter core life and smaller fuel kernels produced higher destruction rate.

The power profile shows peaks at different locations during reactor operation and hence the need for 3D whole core exact geometry optimization. If fuel blocks with more reactivity left are moved to the peaking locations, peaks can be reduced. Placing burnable poisons at the peaking locations will also give the same effect [11]. Blocks with more reactivity are moved to locations such that the localized reactivity peaking is as close as possible to 1.

For evaluating the different arrangements during the optimization process, a compatible parameter that reflects the necessary parameters as defined by different domain characteristics is to be established. This requires subject knowledge and understanding and it is problem specific. The development of the localized reactivity potential is an important part of the optimization and helps isolate the PSO algorithm from the problem-based uncertainties that could affect the algorithm efficiency. This on the other hand rules out the use of genetic algorithm that mutates the parameters at the bit level

and sees the corresponding change in the performance. In genetic algorithm, the variables affecting the localized reactivity potential should be modified at each step towards optimization (it is an iterative procedure) which may not be implemented as the optimal combination may not be realistically achievable. It means that the factors influencing the optimization parameter is based on should be flexible enough to be changed. For example, the localized reactivity potential includes collision rates of various reactions for fuel block. Genetic mutations at bit level mean changing these collision rates and based on that, move the core towards an optimized arrangement. Practically, it is not easy to obtain desired collision densities out of the core with fuel blocks that are already present. Though this is a method that could be used for producing fuel for the VHTRs, it is not a method of optimizing the core using available fuel blocks from the LWRs. The multiple parameter change per iteration may lead to a very large number of combinations for every perturbation per iteration and hence would increase the evaluation of loading patterns even more time consuming. The heuristic algorithm for combinatorial problems, though may not search exhaustively, but performs better in looking at the number of possible arrangements when a small perturbation is introduced. Particle Swarm Optimization is one such metaheuristic solution that can be applied to the shuffling problem.

III.B. SELECTION OF VHTR PARAMETERS AND PERFORMANCE CHARACTERISTICS

The performance characteristics that are expected to change following changes in locations of fuel blocks within the core include the local flux values, the effect of each block surrounding at the previous and current location and the change in reaction rates at that location due to changes in the material composition. The measurement of the reactivity potential and flux reflect the effects of the random fuel block movements.

There is a method for predicting the reactivity potential when a block is moved from one block to the next given by the following reasoning:

If $\rho(\vec{r})$ is the localized reactivity value as a function of spatial location within the system:

$$\rho(\vec{r}) = \frac{RR_{p,MT=18,452}(\vec{r}) - \left[RR_{Leakage}(\vec{r}) + RR_{a,MT=18+MT=102}(\vec{r}) \right]}{RR_{p,MT=18,MT=452}(\vec{r})}$$

$$= \frac{RR_p - \left[RR_{Leakage} + RR_a \right]}{RR_p} = 1 - \frac{RR_{Leakage} + RR_a}{RR_p}$$

Therefore,

$$\rho(\vec{r}) = 1 - \frac{RR_{Leakage} + RR_a}{RR_p} \tag{3.1}$$

For VHTR, a measure of the reactivity using the density of nuclide i, $N_{i,k}$ for each fuel block can be calculated using the following fuel properties:

$$\frac{RR_{Reaction j, Nuclide i, Assembly k}}{N_{i,k}} = \sum_E \phi_k(E)\sigma_{i,j,k}(E) \tag{3.2}$$

for each fuel block.

Where,

$N_{i,k}$ = atom density for nuclide i, assembly k [atoms/cc].

$RR_{Reaction j, Nuclide i, Assembly k}$ = Reaction Rate for nuclide i, assembly k, reaction j [Reactions in cm^3/sec].

$\phi_k(E)$ = Flux for location of assembly k [n/cm^2.s].

$\sigma_{i,j,k}(E)$ = Microscopic cross-section for location of nuclide i, assembly k, reaction j [cm^2].

The characteristic is a dimensionless quantity, which does not reflect the amount of fuel left at any location in reference to the fissile isotopes but the reactivity potential of the individual block at that location. When a block is moved from one location to another, the reaction rates are assumed to be the same at each location. Though it is not true physically, for estimation of the reactivity potential parameter during each shuffle, it is assumed that the reactivity potential depends just on the atom densities of the blocks that are moved to that location. In reality, the reaction rates are to be obtained from deterministic burnup runs for each shuffle. The estimation adopted in this method is

explained by shuffling 3 sample blocks located at different parts of the 1/6th of the core randomly (not pair exchange) using Fig.3.

Therefore, localized reactivity is also given as (Fig.3):

$$\rho(\vec{r}) = 1 - \frac{\sum_{i=1}^{l}\left(\frac{RR_{\alpha,i,k1}}{N_{i,k1}}.N_{i,k2} + \frac{RR_{\gamma,i,k1}}{N_{i,k1}}.N_{i,k2} + ...\right)}{\sum_{i=1}^{l}\frac{RR_{\nu^* fission,i,k1}}{N_{i,k1}}.N_{i,k2}} \tag{3.3}$$

This quantity is to be computed for each block of the VHTR core. Similarly, $\rho_{potential}(\vec{r}) = 1 - \rho(\vec{r})$ is the corresponding localized reactivity potential associated with the system reactivity value:

$$\rho_{potential}(\vec{r}) = \frac{RR_{Leakage} + RR_a}{RR_p} \tag{3.4}$$

Hence excess localized reactivity potential is given by,

$$\rho_{potential}(\vec{r}) = \frac{\sum_{i=1}^{l}\left(\frac{RR_{\alpha,i,k1}}{N_{i,k1}}.N_{i,k2} + \frac{RR_{\gamma,i,k1}}{N_{i,k1}}.N_{i,k2} + ...\right)}{\sum_{i=1}^{l}\frac{RR_{\nu^* fission,i,k1}}{N_{i,k1}}.N_{i,k2}} \tag{3.5}$$

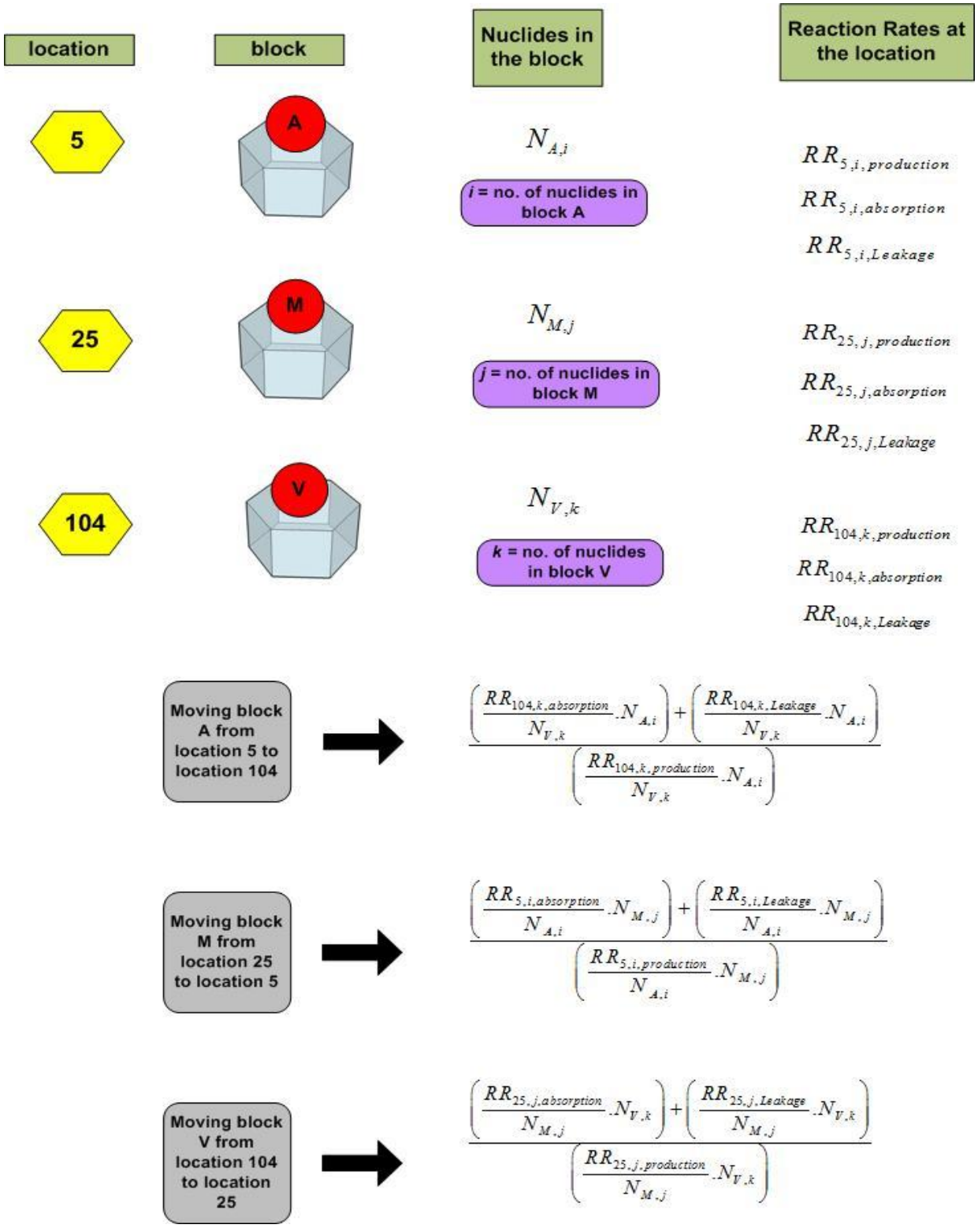


Fig.3 Reactivity parameter estimation method

Eqn.3.9 gives the reactivity potential of the block 1 when it is moved from location 1 to location 2 in $1/6^{th}$ of the core. The reaction rates are obtained as the total collision rates from the output of the initial MCNPX run and the atom densities of the material are also extracted at the end of burn run. The fitness function is developed based on the localized reactivity potential because by minimizing the reactivity potential locally, power peaking is reduced at block level within the core and the burnup will be uniform throughout the reactor resulting in a longer core life.

By optimizing towards a flat power profile, only the minimization of power peaking can be achieved as the cycle length can be increased marginally by reshuffling. Reloading of the core is necessary and by adding burnable poison to the new fuel blocks, increased core life can be obtained [12]. A corresponding penalty should be added to the fitness function to include the effect of newly added blocks in that case. The power peaking in the VHTR is such that it moves to the bottom of the core from the top and back again during the reactor operation. Such a burnup leaves a core with fuel blocks that have a wide variety of burnup levels and hence the corresponding reactivity potential left in them. This reactivity potential can be modeled for all the objective functions mentioned above. By optimizing for a uniform average reactivity potential throughout the reactor and reduced localized peaking, the power peaking can be reduced and a newer core is obtained every time shuffling is done which may give a longer lifetime than that of the original core.

The optimization architecture can be explained in three steps: generate the initial parameter using the reaction rate obtained from MCNP burn runs, estimate values of the reactivity parameter based on simplified relations and search for a optimal loading pattern and verify the reaction rates obtained by burning the final loading pattern. Fig.4 explains the stages in the optimization process.

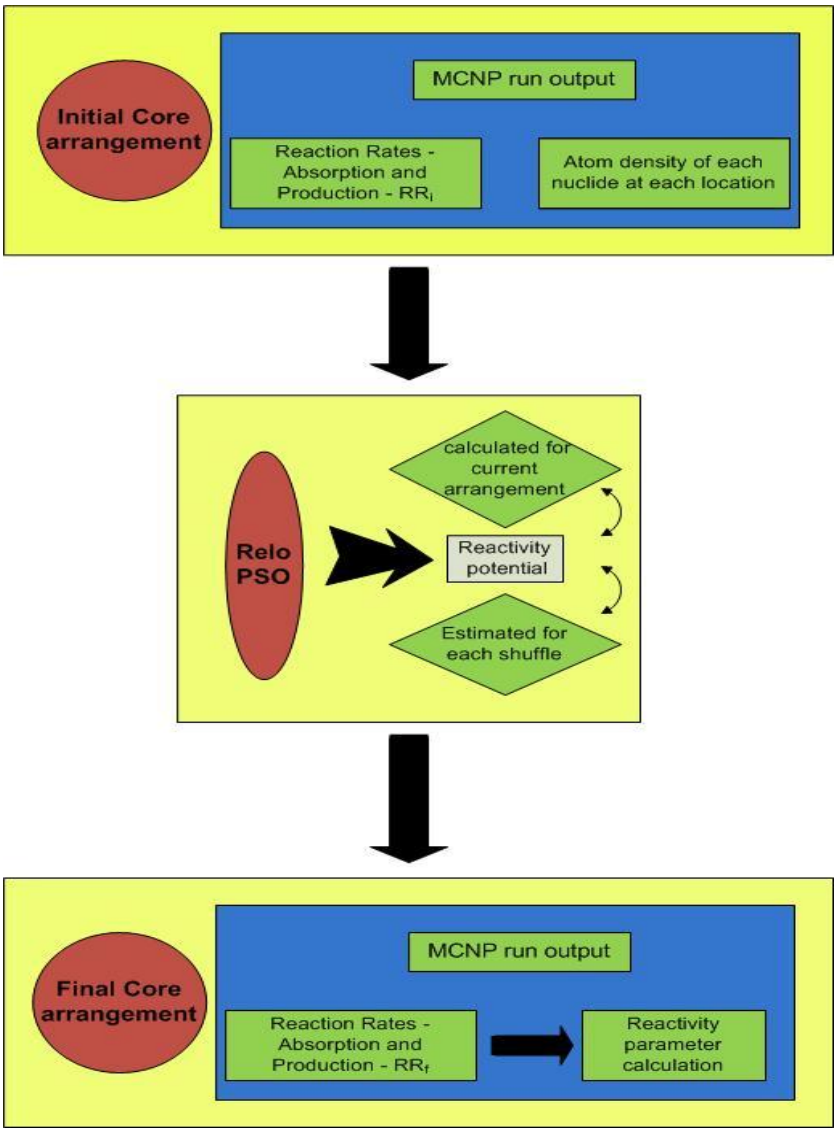


Fig.4 Algorithm and parameter

In VHTR, one sixth of the core consists of 170 blocks of which 50 have control rods that cannot be moved to a fuel block location; only be replaced by another block with arrangement for control rod to be positioned within it. The remaining 120 blocks are arranged in layers of 12 and hence forming 10 layers. The matrix of localized reactivity potentials represents the core. The 3D core data are stored in the 2D matrix. The hexagonal core with the prismatic blocks is divided into six symmetric regions and optimization is done for each region and repeated for the other region assuming that the core behavior will be similar for all the other 5 regions.

As the code can be adapted to other types of VHTR such as the deep burn VHTR that has 4, 5 or 8 ring configurations, the size of the matrix depends on the number of the rings within the core. The number of blocks in each ring from the inner ring to the outer ring is in the order 5, 6 and 6. Of these 2 blocks in the 3rd and outermost rings and one in the 2nd ring are control rod blocks. Each element of the matrix describes the reactivity potential for each block at their location. During the algorithm, the reactivity potential value changes and the blocks are tracked accordingly. A typical 3-ring core (NGNP model) is shown in Fig.5.

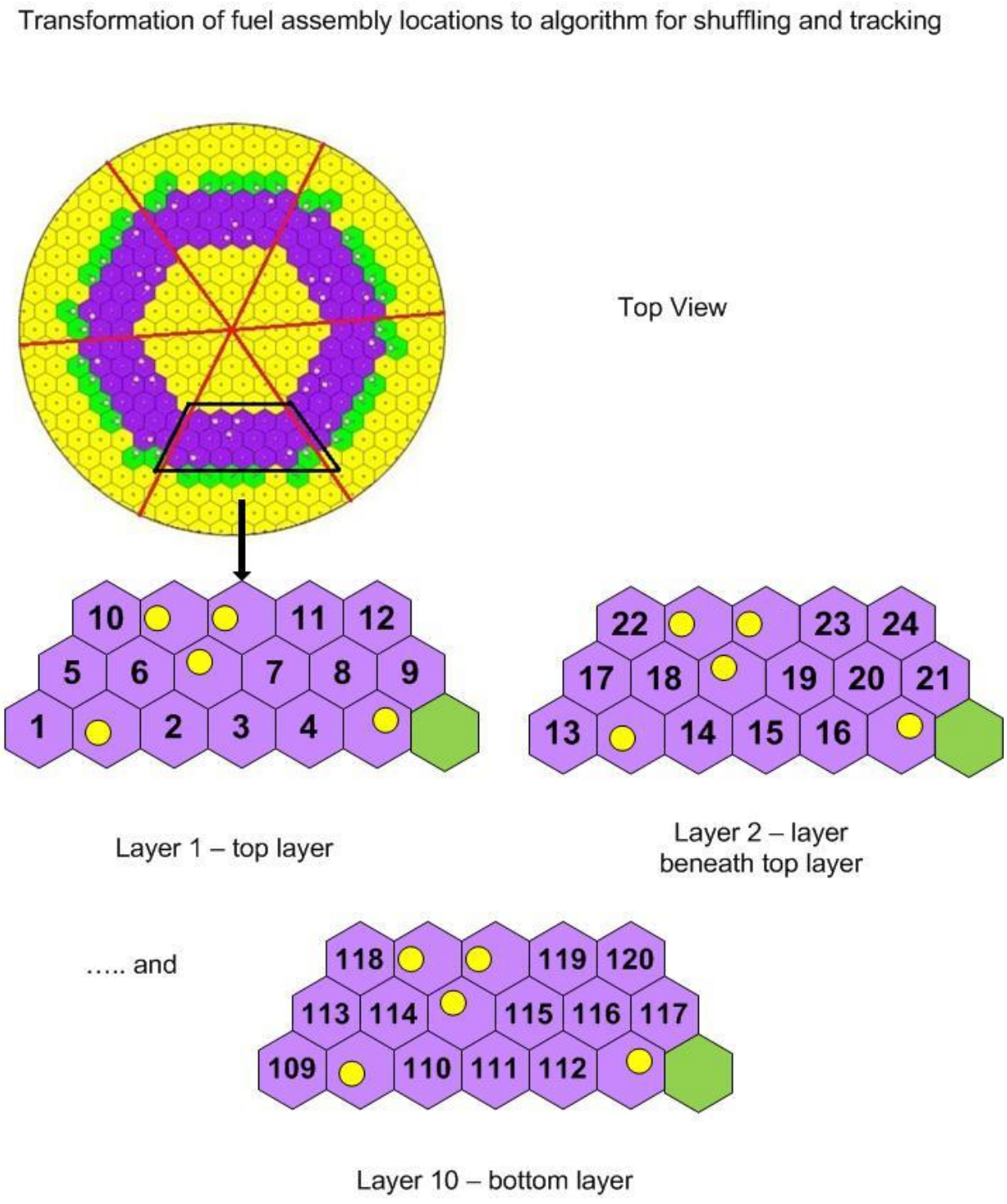


Fig.5 Fuel blocks representation in the matrix in reference to the arrangement in the core

III.C. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization method was initially designed and developed from collective social behavior study of insects, birds and fishes by Kennedy and Eberhart for solving continuous non-linear optimization problems. The authors propose that mind is not an isolated quantity as widely believed because human beings live as social groups and develop their habits base on the social norms and group rules. This algorithm is developed from artificial life and includes evolutionary algorithm characteristics [8, 9]. The authors believe that the evolutionary benefit by sharing the knowledge and experience about food amongst a flock of birds is greater than the competition. It can be a flock of birds or a bee colony may behave the same way [9].

Readjustment, regrouping, changing directions and scattering are inherent properties of the elements of this type of search. The search is synchronous and time inexpensive as it uses only primitive mathematical operators such addition and multiplication only [13]. Since the solutions are given a direction and velocity, the term particle is used. These particles are a disorganized group of moving objects and hence are a swarm rather than a flock. Hence, the name, particle swarms [13].

Any method derived from nature should obey the laws of proximity, quality, diverse response, stability and adaptability. PSO is performed as n-dimensional calculations in iterative time steps and hence obeys the law of proximity. Quality is implemented by the nature of the solution: local and global. The solution adjusts based on the global and local solutions thereby responding to diverse changes. Stability and adaptability both are

established by change in solution only when the global best changes and it does change when a new best is found [13]. This Chapter explains in general the features and steps in the PSO algorithm and its adaptation to the shuffling problem.

III.D. GENERAL ALGORITHM DEFINITION AND APPROACH

The search algorithm PSO draws its method from the social behavior of the birds. The search for the best solution in the entire solution space, which for this type of multi-objective NP hard, combinatorial problems is an entire space with numerous local optimal points and possible multiple near-global optimal point, is analogous to the food domain in which the single source is located. The global optimal point is the single food source towards which the birds fly.

Each of the solutions is called a “particle” which is the bird on the behavioral horizon. Movement of each bird, which is coordinated by a velocity and a direction, is analogous to the movement towards the global optimal point in the algorithm in the solution space. A similar quantity of velocity and direction is calculated for each of the particle. Each of the particles reach a position at the end of each iteration depending on its current position, its own best position and the best position of all the particles in the swarm. The initial population of the particles, the location and the velocity, are initialized randomly. Each particle is then evaluated by a fitness function in each generation and has the best position from its memory and the best position among the swarm that are based on the fitness function [13]. There are two approaches to this method each of which are explained in the next section.

III.D.1.1. Inertia weight

At the end of each generation each of the particles is given a new position that is a step towards the optimal position and a new velocity with which the particles move. The following equations are used for improving the position and velocities of the particles [12]:

$$\begin{aligned} v_{id}^{t+1} &= w^t v_{id}^t + c_1 r_i^t (pbest_{id} - x_{id}^t) + c_2 r_2^t (gbest_d - x_{id}^t) \\ x_{id}^{t+1} &= x_{id}^t + v_{id}^{t+1} \end{aligned} \tag{3.6}$$

where

$i = [1, 2, ..., n], n$ is the number of particles, i.e., the number of fuel blocks, in each swarm

$d = [1, 2, ..., m]$ denotes the number of dimensions of the problem

v_{id}^t = Velocity of the particle i at step t

x_{id}^t = Position of particle i at step t

c_1, c_2 = Acceleration constants (discussed later)

r_1^t, r_2^t = Random numbers between [0, 1]

w^t = Inertia weight at step t

$pbest_{id}$ = Previous best position of particle i at step t called the best neighbor (local best)

$gbest_d$ = Best position among all particles for the current iteration

t = iteration number

All these notations are obtained by modifying the terms in [15] and [13]. Based on the analysis done by the authors in [13], the acceleration constants c_1, c_2 are such that

$$c_1 + c_2 = 4.1 \tag{3.7}$$

Authors who originally came up with PSO also use something similar to this [14]. The choice used is $c_1 = 2$ and $c_2 = 2.1$

The inertia weight factor is not a fixed quantity throughout the optimization process. It is used a factor that reduces linearly over the length of the optimization process based on [15]:

$$w^t = w_{\max} - (\frac{w_{\max} - w_{\min}}{t_{\max}}).t \tag{3.8}$$

where $w_{\max}$ = initial weight

$w_{\min}$ = Final weight

$t_{\max}$ = Maximum generation chosen for the problem

The suggested value for $w_{\max}$ = 0.9 and $w_{\min}$ = 0.4. It is a reducing function because, at the initial stages of the search, the particles should try and reach out of the current neighborhood and look for better solutions. As the number of iterations increases, the particles should converge to a final solution and the hence the weight for change is smaller. The maximum dimension used is 80 and 100 for [15]. This can be obtained by running the code repeatedly for a few times [14].

III.D.1.2. Constricted factor approach

The difference between the inertia weight and constricted factor method is the difference in the velocity formula. The simplified form of the velocity is given by [15]:

$$V_{id}^{t+1} = cfk(V_{id}^{t} + c_1 r_i^t (pbest_{id} - x_{id}^t) + c_2 r_2^t (gbest_{id} - x_{id}^t)) \tag{3.9}$$

where $cfk = \dfrac{2}{\left|2 - \varphi - \sqrt{(\varphi^2 - 4\varphi)}\right|}$

$\varphi = c_1 + c_2$ and $\varphi > 4$

Search in wide areas and the convergence criteria can be controlled by φ.

The procedure as given in [12] is shown below. The position and velocity are shown as vectors denoting the multidimensional nature of the problem, here 3 dimensions for the VHTR. The initial position and the direction of the particles are randomly generated. The limits and the values for the parameters are obtained from the algorithm definition [13] as they are independent of the type for the problem, but are the features of any combinatorial problem to be solved by Particle Swarm Optimization. The fitness function is similar to the one used for reducing the power density in [15].

III.D.2.1. Initialization procedure

For each particle i with P particles:

(i) Initialize $\vec{x}_i^t$ randomly

(ii) Initialize $\vec{v}_i^t$ randomly

(iii) Evaluate the fitness function $f(\vec{x}_i^t)$

(iv) Initialize $\vec{p}_i^t$ with a copy of $\vec{x}_i^t$

Initialize $\vec{g}_i^t$ with a copy of $\vec{x}_i^t$ with the best evaluation

III.D.2.2. Iteration process

For each particle i

(i) Update $\vec{x}_i^t$ and $\vec{v}_i^t$ according to the position and velocity vectors

(ii) Evaluate the objective function $f(\vec{x}_i^{t+1})$

(iii) If $f(\vec{p}_i^{t+1}) < f(\vec{p}_i^t)$ then $\vec{p}_i^t \leftarrow \vec{x}_i^{t+1}$

(iv) If $f(\vec{x}_i^{t+1}) < f(\vec{g}_i^t)$ then $\vec{g}_i^t \leftarrow \vec{x}_i^{t+1}$

$\vec{g}_i^t$ is the best solution of a particular iteration and it saved for comparison with the next obtainable solution. The next one may or may not be better that the current best. The discussion about the fitness function and the penalty function ($f(\vec{x}_i^t)$ and $f(\vec{p}_i^t)$) are in the next section. The iteration is carried out for each particle in the swarm till a pre-defined limit. The process is repeated with different random seeds and the best option may be picked. It is adapted from the work in [12]. The same algorithm is represented as a flowchart in the Fig.6 [15] in a simplified form where the algorithm is stopped with the number of iterations reaches the preset value t_{max}.

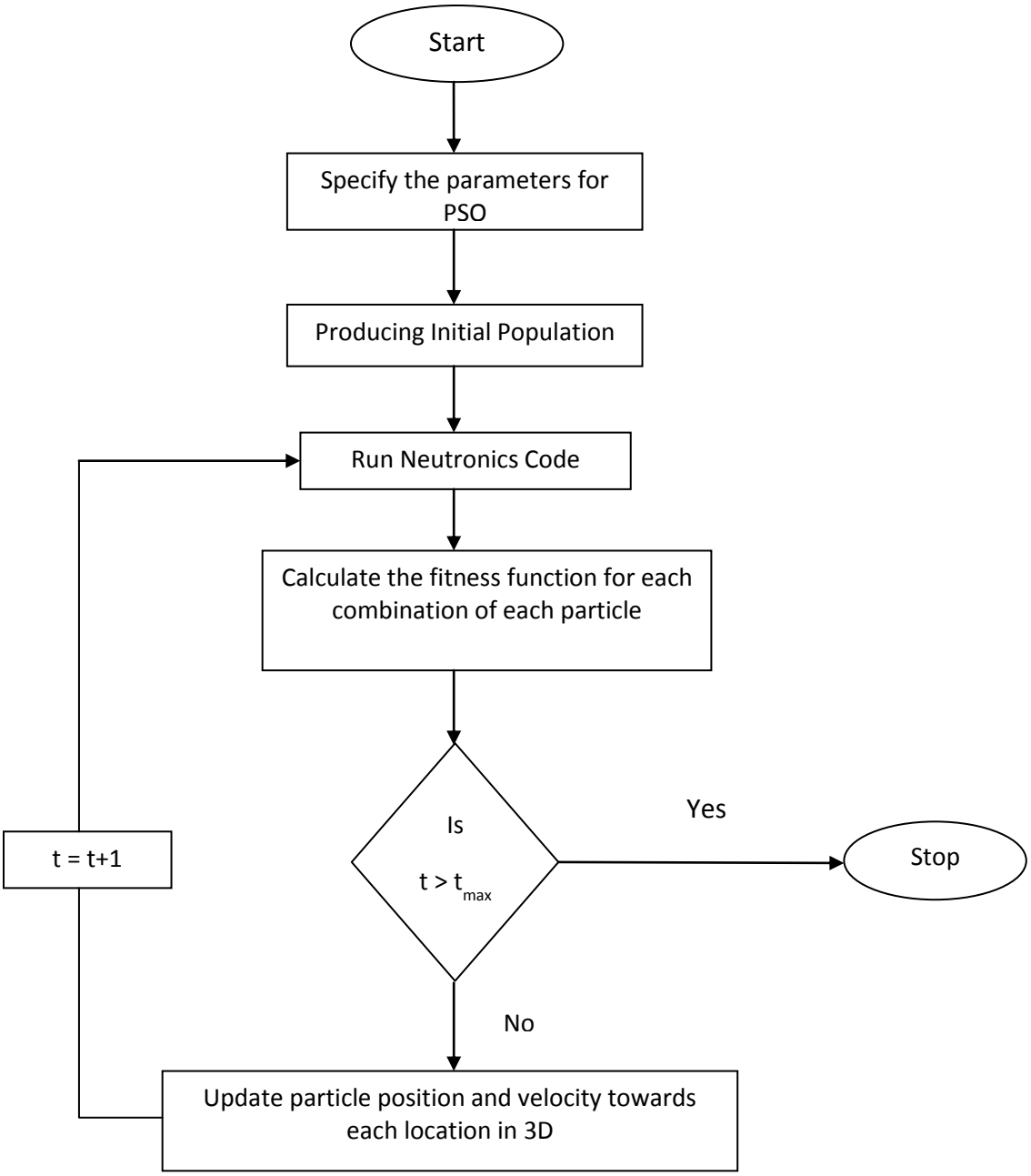


Fig.6 Flowchart of the general PSO steps [15]

III.E. OPTIMIZATION ARCHITECTURE

The top-level view of the problem and solution is depicted in Fig.7. MCNPX burnup is the main tool besides the algorithm. A full core exact geometry with TRISO particle design burnup takes around 2 weeks runtime on a single core processor. A faster code, deterministic or stochastic, could help develop more accurate values of the collision rate densities for each step of the shuffle run.

Initial MCNPX burn deck is generated based on the user given input with similar material at all locations but treating it as separate material cards. After the first run, the required data is extracted using a text parsing script and the data is accessed by the PSO algorithm to calculate the localized reactivity potential. Other required input parameters such as the maximum limit for the localized reactivity potential are given through the user input file PSO.RUN. Intermediate data saves by algorithm are done and the final arrangement per iteration is also saved.

The intermediate data access includes the data saved as predicted values of the localized reactivity potential for a block movement to any location within the $1/6^{th}$ of the core, data saved as matrices. Intermediate data also includes the initial loading patterns generated randomly, intermediate particle fitness function values, etc. The algorithm verification establishes the applicability of the algorithm to the problem. Analyzing the MCNPX run of the final output, the effect of the shuffling process can be understood. Final output is generated based on the user requirements and commands in the

PSO.RUN file and it has values for corresponding parameters with detailed intermediate data and inferences.

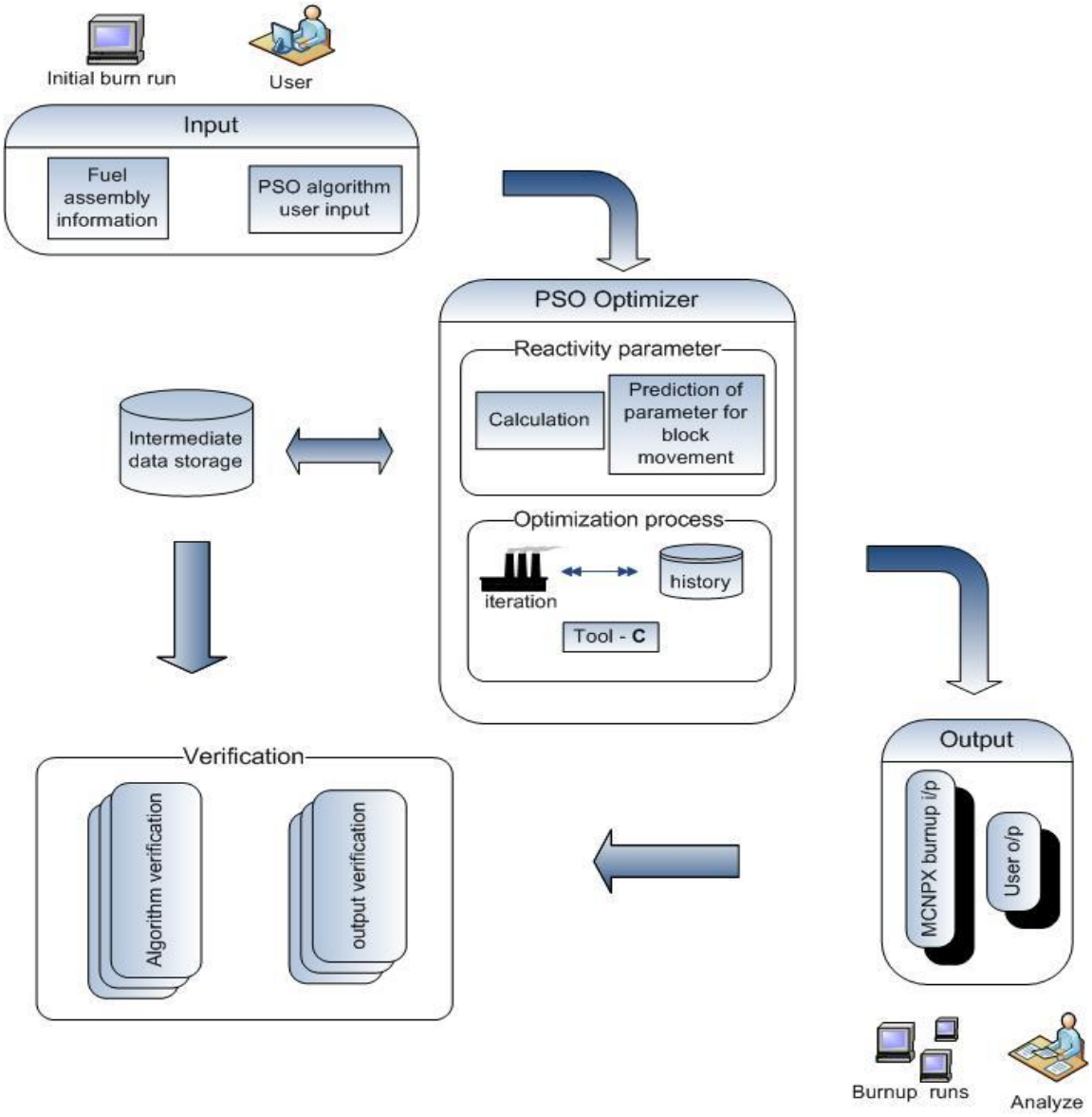


Fig.7 shuffling algorithm architecture

CHAPTER IV

SHUFFLING OPTIMIZATION USING PSO

IV.A. ReloPSO - 3D

The actual adaptation of the PSO algorithm for the optimization of VHTR core shuffling is explained in this section. The developed algorithm is called ReloPSO - 3D and is written in C. The particle set is $x_i^t = \left(x_1^t, x_2^t, ..., x_n^t\right)$ where *n* is the dimensionality of the problem or the number of variables is *n*, for this 3-ring model under consideration it is 120. As it is a n dimensional problem, each of the $\left(x_1^t, x_2^t, ..., x_n^t\right)$ is going to have 120 dimensions each as this problem is attempted at solving in 3D. For an LWR, shuffling is done in 2 dimensions and hence it is solved for the top layer arrangement and so the dimensionality is less. Kennedy and Eberhart [14] who developed this method have tested till a maximum of 30 dimensions and no restrictions are mentioned for the number of particles. Since the number of blocks in $1/6^{th}$ of the core is taken as the dimension, the shuffling problem is more complex than the LWR reloading problem using the same optimization methodology.

The following equations are used for calculating the velocity and new location of the particles and are adopted from [15].

$$v_{id}^{t+1} = w^t v_{id}^t + c_1 r_i^t (pbest_{id} - x_{id}^t) + c_2 r_2^t (gbest_d - x_{id}^t)$$
$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \qquad (4.1)$$

During each iteration *t*, a velocity component is added to each variable x_{id}^{t+1} such that each variable changes based on their local best and the global best among all the particles. In the above equations, v_{id}^{t+1} is the new velocity vector of each particle *i* for iteration *t+1*. v_{id}^{t} is the current velocity vector whose elements are in each of the 120 dimensions in generation *t*. The added velocity values will move the particles towards the optimal solution. The arrangement is not monitored because based on the ranking method explained later the search area is widened and the blocks are tracked at the same time. The quantity by which the reactivity potential parameter must be altered at each block location so that the loading pattern is evaluated with the minimum penalty is decided by the inferred velocity maximum, the previous velocity, the previous best fitness value of the loading pattern and the global best among all the loading patterns. The interchange between the blocks location and the reactivity potential is a feature of the algorithm, which acts as the stimulant to widen the search area.

$pbest_{id}$ is the personal best of each of the particle in among the 50 combinations in the current generation for each FA. x_{id}^{t} is the position of the current particle with components which are the 120 fuel blocks in generation *t*. gbest_{d} is the best in each dimension, that is in each location, among all the 50 particles in the swarm. x_{id}^{t+1} is the new location of each particle in the swarm with components in 120 locations. The position vector is then updated and the process continues for each of the predetermined number of iterations. The whole process can be repeated many times by using different

random seeds for the initial population generation and for the constants in the velocity equation and the best location can be compared between different swarms.

The parameters in the velocity equation c_1 and c_2 are found to be major factors in guiding towards the optimal LP. c_1 and c_2 are the acceleration constants that are used for changing the velocity towards converging on a local best or a global best. When $c_1 >> c_2$, the solutions converge narrowly and into the local solution. $c_1 << c_2$, the particle increases the pace of the search without control and leads to what is called an explosion, which means the diverging to huge values of the fitness function. Controlling the velocity of the particles usually does this. The velocity component v_{id}^{t+1} is given by an interval $[-V_{max}, V_{max}]$[16] [13]. Imposing such a hard constraint on the particle seem have hamper the PSO and so the fine line between exploration into wider solution space and the exploitation as called in [16] and [13]. It means that the solution will not converge to a final loading pattern and it will change continuously when the values for velocity limits are not within velocity range as V_{max} is very problem specific. It is calculated as one standard deviation of the reactivity potential of the initial core arrangement for this problem.

The inertia weight factor in the inertia method is the factor that is used to control the velocity component along with the acceleration components c_1 and c_2. The inertia weight factor, w^t, can be analogous to the fluidity of the particles; greater the value of w^t greater is the fluidity and the range is tested as between 0.9 to 0.4 [13]. Initially, w^t

is high and it is gradually reduced to 0.4, which is analogous to a flow in a low viscous fluid initially, and later it is reduced and so the algorithm is made to concentrate on the optimal point. The value of v_{id}^{t} need not be constrained, but sometimes a high value of V_{max} is initiated with a value equal to the maximum jump that a FA is to make during the shuffling. For this problem, as there are 120 blocks in the 1/6th of the core, it can be set to 120 in each direction of the velocity component. Hence, the ability to manipulate the PSO towards a global solution and homing in on a local solution finally is done by controlling w^t and because of that the numbers of iterations are less in this method. Usually V_{max} is unknown and so the maximum jump along each dimension possible is used as V_{max}. If this value < 2, then a lower w^t as small as 0.4 is used and when ≥ 3, w^t is used as high as possible, like 0.8 based on the experiments by reference [16]. Time varying w^t, is found to accelerate convergence. The remaining factors in the velocity equation can be interpreted as the external forces acting on the particles, which are the personal and the group bests. r_1 and r_2 are random numbers between [0,1] generated for each iteration.

Tracking fuel blocks during the shuffling algorithm is an important part in random shuffle different arrangements with blocks moved to any possible location are tested. In PSO, the algorithm in itself contains the location and the velocity of the particles at each of the iterations.

The total number of rings in the VHTR is 3 and the arrangement of the core may be either annular with 3, 4 or 5 rings of fuel surrounded by reflectors and with reflectors in the middle of the core as shown before (for DB-VHTR). The algorithm can be modified to any arrangement.

As the fuel blocks with control elements are not shuffled with regular fuel blocks, their locations are not considered for the shuffling process. The control element fuel blocks are calculated out from the number of rings given as input. These steps help in determining the number of fuel blocks in case of different number of rings in the fuel core and then the model core map is developed from the number of blocks.

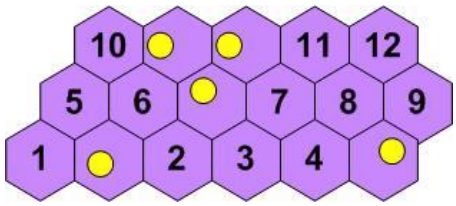


Fig.8 Core representation at layer 1- top layer

A sample numbering of the blocks is shown Fig.8 for a 3 ring core. The current core arrangement is taken as the initial arrangement and 50 possible random arrangements are formed as the initial particle population and the corresponding velocities are also generated randomly. A sample calculation of the tracking mechanism is shown in Fig.9. The values used are random to exhibit the change in velocity, then the reactivity potential value and hence in the arrangement. For every arrangement (particle), the localized reactivity potential set is obtained for each fuel block which are the x_{id}^{t} . The

new velocity set, v_{id}^{t+1}, for that arrangement (particle) is obtained from equation (4.1) based on the previous velocity v_{id}^{t} and other parameters. The x_{id}^{t+1} set for each arrangement is obtained from equation (4.1). Then they are ranked and based on the ranking and by comparing to the unranked x_{id}^{t+1}, the new arrangement is obtained. This process is repeated per iteration for each particle.

The ranking of each block within the loading pattern to get the next loading pattern may appear without any real tracking of blocks but the search is randomized even when there is a local best solution. The current best solution is saved and the next combination is randomly picked and verified if it is better than the current solution. This step helps in branching out of the current solution in search of a better solution. It may lead to a very different solution area and LP obtained may not be optimal, but the search is widened. This method of developing new LPs from the current set of particles is adopted from [12] and [11].

Though this step does not guarantee a better solution after each of the iterations, at the end of each of the iterations, the result obtained is at least as good the one got in the previous step. The particle also converges to the final solution when sufficient numbers of iterations are considered. The convergence is seen in the LP map and the fitness function value. Running the code and looking for the convergence in loading pattern can reach this number.

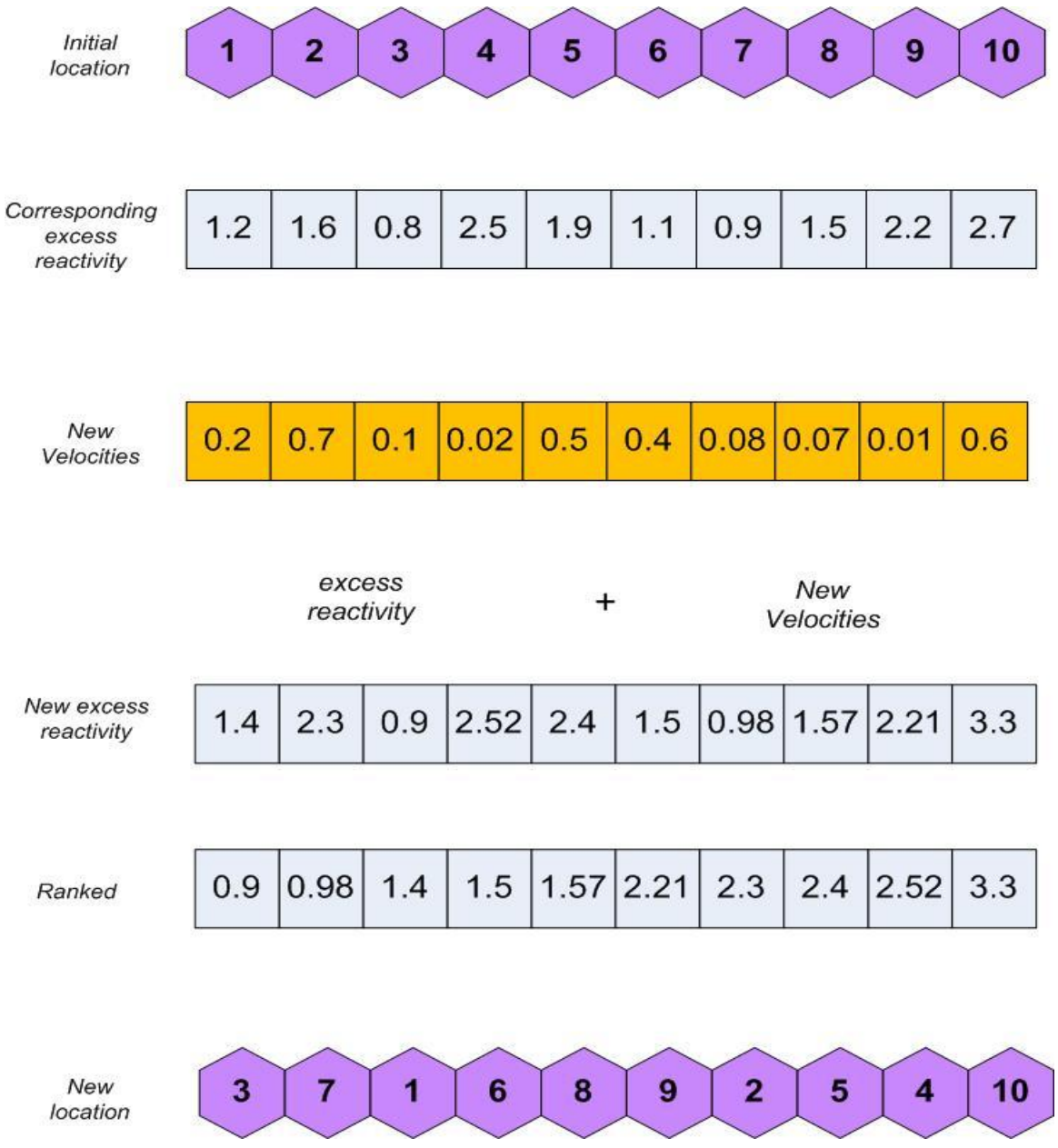


Fig.9 Tracking and new location per arrangement per iteration (random values)

IV.B. FITNESS FUNCTION DEFINITION

Consider a set of *A* fuel blocks and *L* of locations of *n* elements, each which means that 1 FA occupies just one location or the FA, should not be repeated within the reactor. For $i \in A$ and $j \in L$, this condition is given by: $X_{ij} = 1$ if i assigned to *j* and $X_{ij} = 0$ otherwise. The objective function is maximizing the cycle length or minimizing power peaking, which is a function of the position of the fuel blocks (the possible permutations). By minimizing the power peaking, uniform core-wide power distribution can be obtained increasing core safety. The core life may also increase by this process. Since power is proportional to flux density, this can be achieved by varying the number of fissile particles per unit volume of blocks at different locations within the core [10]. The neutron flux increases with increase in reactivity and hence the fission rate increases which in turn increases power. Hence by minimizing the localized reactivity potential, the power peaking can be minimized. The method and the notations were adapted from the power peaking minimization in [15].

P_{FA} = Localized reactivity potential at each Fuel block location calculated from the reaction rates and the atom density for the initial arrangement and estimated for the shuffles.

$$P_t = \sum_{1}^{120} P_{FA} \tag{4.2}$$

P_t is the total localized reactivity potential of 1/6th of the core.

170 is the number of fuel blocks in 1/6th of the reactor core of VHTR (3-ring configuration) including 120 regular fuel blocks and 50 fuel blocks with control rods. In the actual research, it is done on 2D from the top view of the core, consisting of 12 in each of the ten layers.

P_{avg} = Average reactivity potential function of $1/6^{th}$ of the core is given by

$$P_{avg} = P_t / 120 \tag{4.3}$$

The normalized localized reactivity potential at each block is given by

$$np_{FA} = P_{FA} / P_{avg} \tag{4.4}$$

For guiding the search process, a fitness function, which evaluates each loading pattern (LP), under consideration based on its advantages and disadvantages, is developed.

$$\text{Sum squared error function for each LP} = \sum_{i=1}^{120} (np_i - 1)^2 \tag{4.5}$$

The available constraint, the upper limit of the localized reactivity potential factor, could be used to guide the search in a particular direction. According to [15], this is linearly integrated into the fitness function. It makes sure that if the localized reactivity potential of a particular loading pattern is higher than the maximum allowed peaking within the core, that particular arrangement will not have a higher probability to propagate through the search algorithm, i.e., has a lower possibility to be reconsidered. This implemented by means of a linear function based on the safety limit violations. It is added to the

fitness function f_{FIT} if the limit is exceeded and not if it does not. But the LP is considered because a better solution may be obtained from an unfavorable solution during subsequent optimization steps.

The penalty function is given by:

$$f_{pen} = \begin{cases} \dfrac{pf_{\max}}{pf_{\lim}} \cdot \sum_{i=1}^{120} (np_i - 1)^2 & if \quad pf_{\max} \geq pf_{\lim} \\ 0 & if \quad pf_{\max} < pf_{\lim} \end{cases} \tag{4.6}$$

where $pf_{\max}$ is the maximum localized reactivity potential peaking for that particular arrangement during that iteration and $pf_{\lim}$ is the safety limit for the reactivity potential factor for the design. It may be obtained by identifying the limiting localized reactivity potential corresponding to the limiting power peaking value and added to the fitness function.

The penalty function serves two purposes:

(1) It adds up all the peaking that is excess to unity and it is used to evaluate the loading pattern is better or worse. For example $(np - 1)^2$ has a value if the normalized reactivity potential is greater than or less than 1. Less than 1 implies that the fuel block reactivity potential is less than the average reactivity potential (peaking in the negative direction) and greater than 1 implies values analogous to power peaking.

(2) The penalty function penalizes the loading pattern that has a greater maximum reactivity potential more than the one that has a lesser maximum peaking. This solves situations when f_{FIT} values of multiple loading patterns are equivalent. It is penalized if an arrangement has a peaking value greater than the allowed value and hence ranked lower but not removed entirely. It could move to another better arrangement later in the algorithm

The fitness function becomes

$$f_{FIT} = \sum_{i=1}^{120} (np_i - 1)^2 + f_{pen} \tag{4.7}$$

which is the final fitness function [15].

The fitness function is the decision variable of the algorithm. Since it is a minimization problem, the least the fitness function is, the better the solution. The fitness function does not oscillate about a minimum value. The fitness function is a decreasing function for particle swarm optimization. It is due to the fact that at each step the best arrangement is worked on to get a even better solution. The minimum value for the fitness function is 0, which is practically unattainable for such real-time problems. But for the VHTR reloading problem, the initial step in itself is a valid arrangement and the localized reactivity potential is a small number. Hence the fitness function value is low and may get closer to 0.

IV.C. ReloPSO FLOW

Fig.10 shows a detailed flow of the algorithm control and the adaptation of the steps in the Fig.6 (the general algorithm flow). Due to the hexagonal shape of the core, it can be symmetrically divided in to 6 parts and by optimizing one of these sections, the process can be repeated for the rest of the core as burnup is assumed to be symmetric axially. The number of fuel blocks is taken as the dimension of the algorithm. The number of rings is given as input and the number of blocks per ring and the total number of blocks are calculated. The number of iterations the algorithm to be run, the number of particles (agents or LP), the maximum for the power peaking and other details to be printed in the output file are defined in the input file. The discussion about the parameters of the PSO algorithm is an important one as the problem independent nature of those problems is the basis of the usability of this algorithm [16]. Among the fixed parameters, the number of particles in the swarm is fixed. It is the initial number of loading patterns (LP) to start with. From the works of Kennedy and Eberhart [14], it is found to be in the range 20-50 but for our problem it is chosen from 50-120.

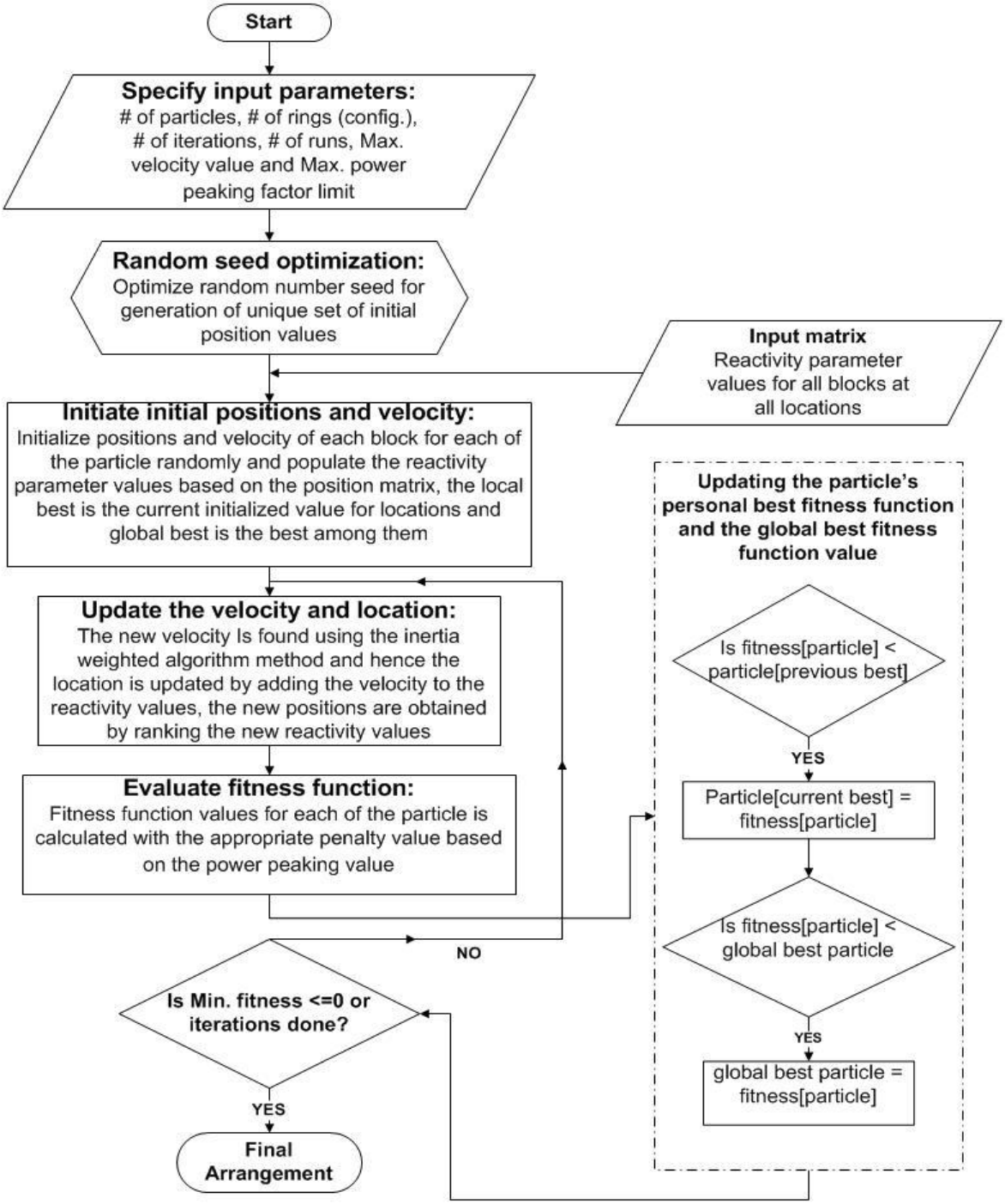


Fig.10 VHTR shuffling algorithm – ReloPSO

Optimization of random number generator is done to get a better sequence of random numbers for generating the LPs and their corresponding velocities. For the algorithm to begin, the initial set of particles of to be generated randomly. It includes the number of particles set of LPs and the initial velocities associated with each fuel block in each pattern (particle). For the input matrix, the atom density of different materials in each fuel block is extracted from the MCNPX output file along with the collision rate densities for different neutron absorption (fission, gamma, alpha, n-p reactions) and neutron production reactions (fission, n-2n, n-3n reactions). The excess reaction rate is then calculated as a ratio of sum of absorption reaction rate to the neutron production reaction rate.

For the initial step, the initial loading pattern set is $pbest_{id}$ and the loading pattern with the best fitness function is $gbest_{id}$. The new loading pattern that is developed from the original set is then obtained from the tracking method discussed before. The process is repeated to find $pbest_{id}$ and $gbest_{id}$ for every loading pattern and every set of loading patterns respectively. Both the values are updated during each of the iterations and the process is repeated till the number of iteration steps defined in the input function is completed or when the fitness function reaches 0.

IV.D. CODE STRUCTURE

The code structure of the ReloPSO - 3D is shown in Fig.11. ReloPSO.cpp is the main file, PSO.RUN is the input file and the other files do the initial random generation of velocity and so on. The initial position is obtained from the core arrangement and hence

the corresponding initial reactivity values. Fig.11 also gives the flow of data in the algorithm that includes files created as intermediate output and final part of the algorithm that are used for verification and output display.

Memory allocation is done dynamically as the matrix sizes differ with the different core configurations. This defines the flexibility of the code. The memory allocation is also done for other differences such as the number of particles in the PSO.

The intermediate storage and the processing of the reaction rates are done outside the main function. The first iteration of the optimization algorithm is separated from the rest of the steps as it is also used to generate intermediate data such as the fitness function and the localized potential parameter. The arrow directions denote the type of flow, one way or both ways, between the functions. The material information such as the nuclide present in each fuel block and the corresponding atomic weights are extracted from the initial MCNPX run and are input into the final MCNPX according to the new locations of the assemblies obtained at the end of the optimization step. Other changes to the input file can be made such as the number of particles for k-code run or the burn steps, etc. Intermediate storage includes multiple run fitness function information, local best values so that a comparison can be made between different runs at the same point of shuffling. They are useful from analysis of the algorithm and verification of the particle properties.

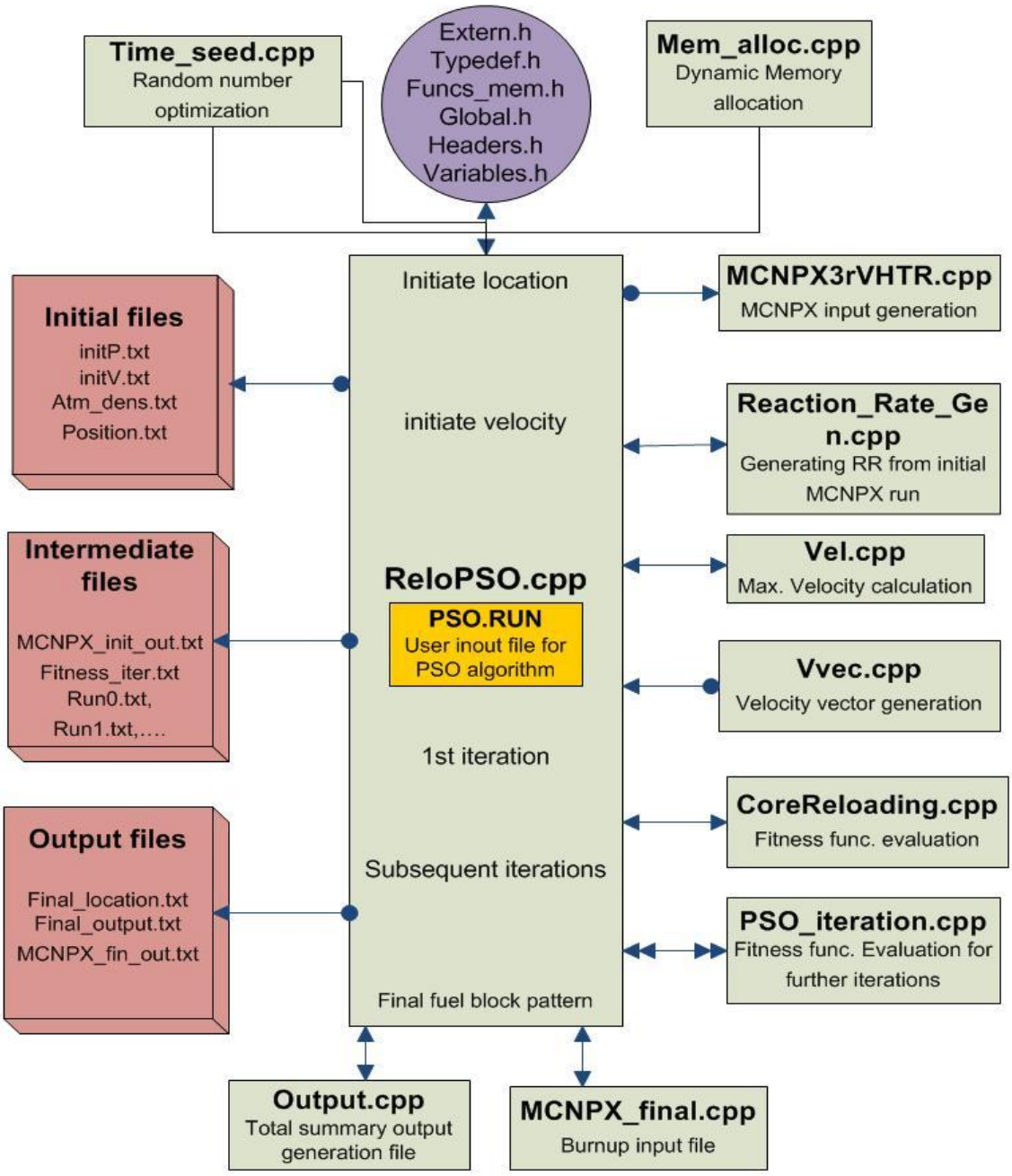


Fig.11 Code structure and control direction in ReloPSO - 3D

CHAPTER V

ALGORITHM EXECUTION, RESULTS AND DISCUSSION

V.A. ALGORITHM VALIDATION

The algorithm is executed in 2 phases: (1) As with any algorithm applied to any real time problem, the calibration of the algorithm and the validation of the results are performed. (2) Then the algorithm is applied to the problem. Calibration of the algorithm involves establishing valid values for the algorithm parameters such as the *V_max,* weight constant limits, etc. Other parameters are the number of iterations that are necessary for the fitness function to converge and the number of particles or the initial loading patterns.

Performing various tests of the fitness function shape and the particle behavior during the course of the search process validates the algorithm. Various parameters are changed and the corresponding effect on the data flow and the fitness function is evaluated. Each experiment is a combination of different input parameters - number of iteration, maximum velocity and the number of particles. Comparing the fitness function behavior to the general PSO fitness function shape validates the ReloPSO algorithm. The fitness function for the optimization of PWR reloading problem using PSO is shown in Fig.12 [15]. As the PSO algorithm works towards finding the optimal solution value and has two solutions at each of the iterations: a local best and a global best. Since there is a global best at all times, for each of the iterations, the fitness function should always decrease. This explains the pattern observed in Fig.12.

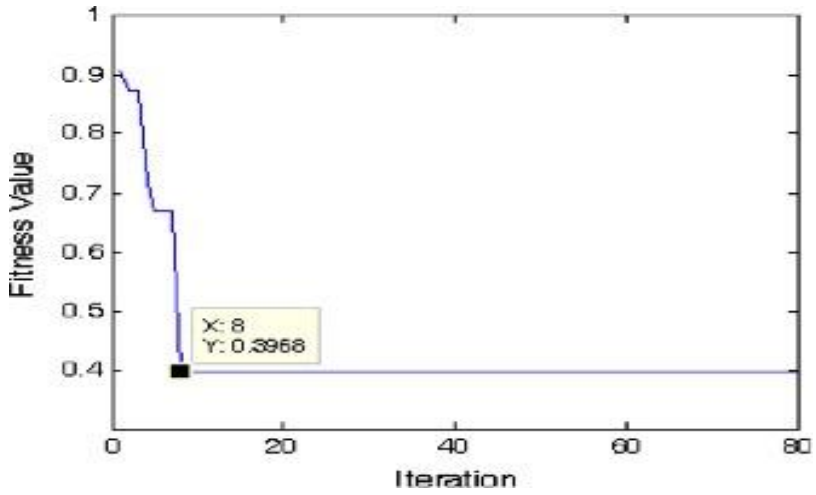


Fig.12 Reference fitness function [15]

The same pattern is observed for the ReloPSO algorithm as seen in Fig.13. The same pattern is observed for various values of *V_max* and different number of iterations.

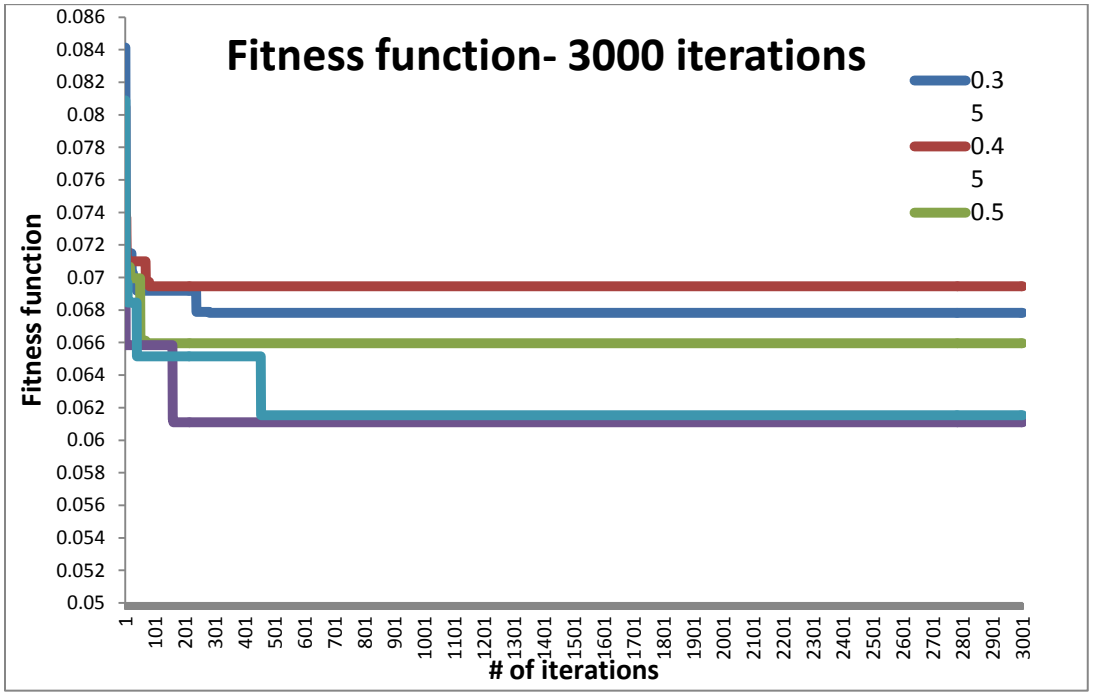


Fig.13 Algorithm parameter estimation – fitness function shape - 3000 iterations

V_max values used for the algorithm are in the range 0.35 - 0.88. The numbers of iterations used are 3000 and 5000. Though the fitness function resembles the Fig.12, no pattern is found with respect to the convergence speed as seen from Fig.13 and Fig.14.

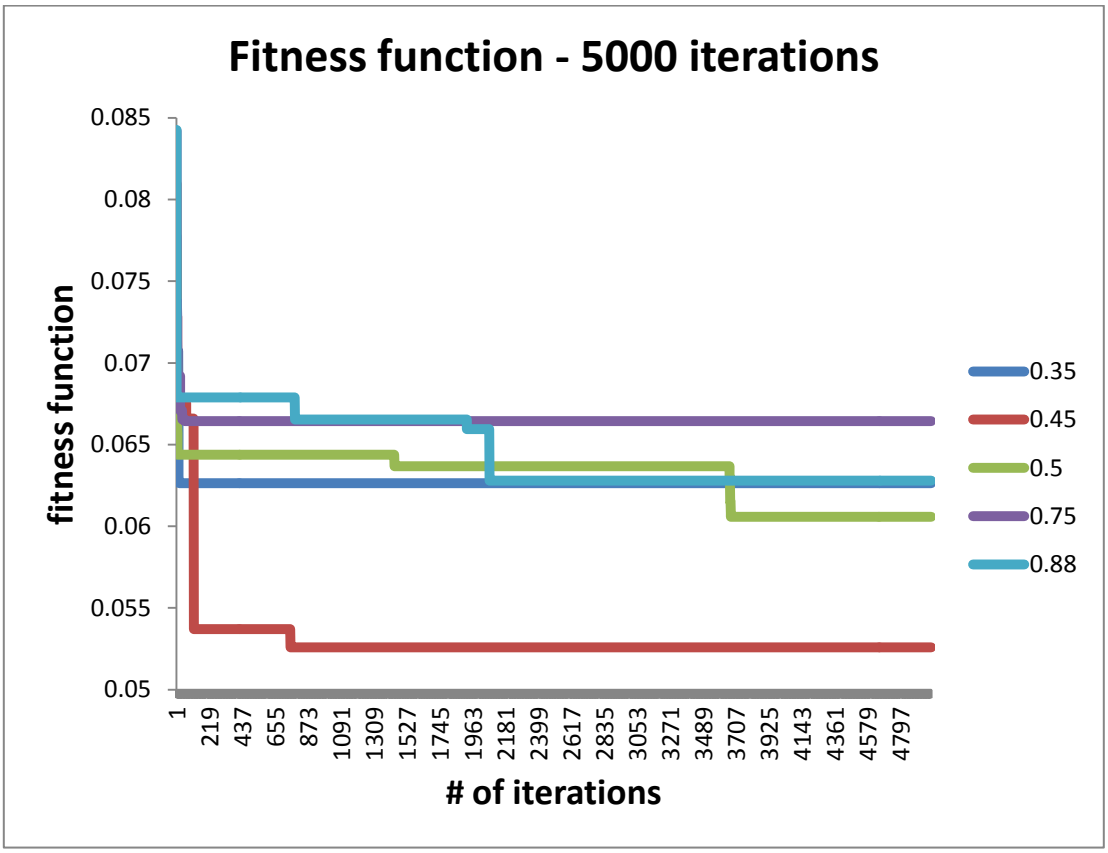


Fig. 14 Algorithm parameter estimation – fitness function shape - 5000 iterations

Maximum velocity is one of the important parameters in the algorithm. It is highly problem specific and it influences the convergence of the fitness function greatly. A higher value might make the particle oscillate within useful range while a lower value may reduce the oscillation of the particle value. One of the tasks of calibration is to find the best combination of the input parameters that could eventually lead to the Pareto set

of outputs. As the maximum velocity value is problem specific, in ReloPSO it is based on the dynamically varying parameter of the problem, the reactivity potential. The other reason is that reactivity potential is the optimization parameter that characterizes the fuel blocks. *V_max* is computed as one standard deviation range of the mean reactivity potential of the initial fuel block arrangement.

The convergence of the fitness function value is tested by plotting the fitness function of each of the iterations of each particle and the global optimal points of the run. For establishing the convergence, a five-particle sample set is used and the fitness function is plotted against the iteration steps.

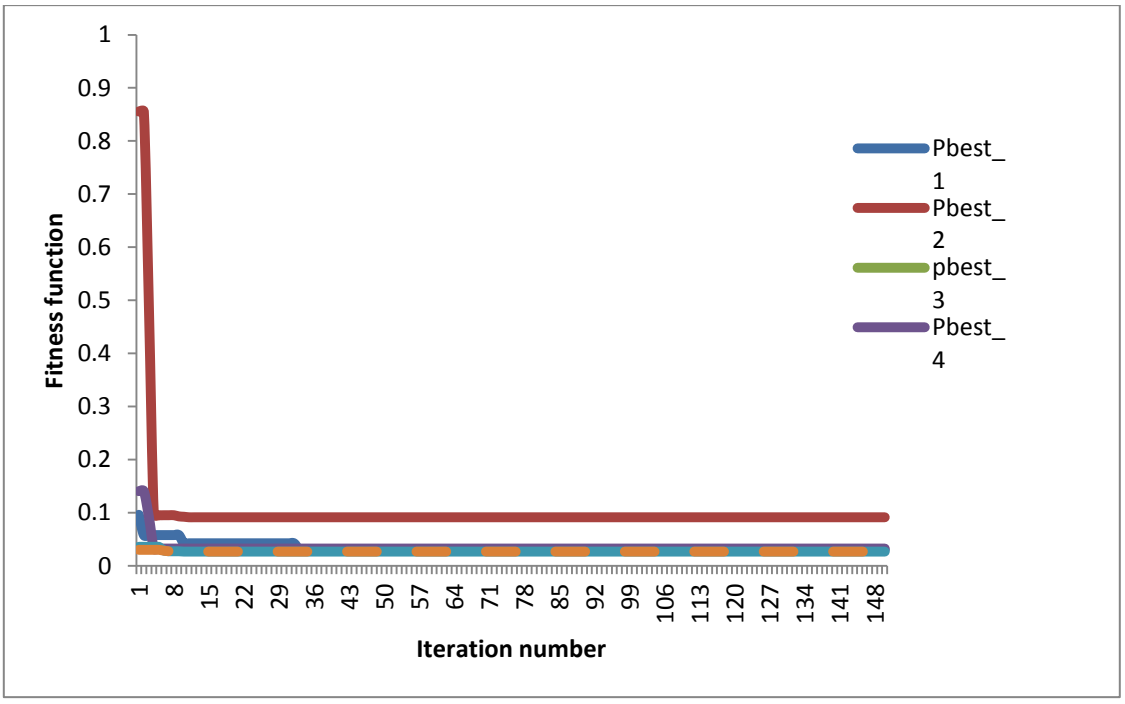


Fig. 15 5-particle sample showing convergence to a global best

Each particle has its local best that is stored in memory and used during the course of algorithm to move to the new better loading pattern. The best solution among all the particles is also saved for each of the iterations and considered for the next better solution. The dotted line in Fig.15 shows the global best (g_best) points of each of the iterations and hence the final solution. It shows that the around 80% of the particles' local best converges to the global best.

V.B. PARTICLE BEHAVIOR

The most important function of combinatorial algorithms is the ability of the solution to get out of a local solution and search for a better solution. The algorithm must search the solution space even after reaching a local solution.

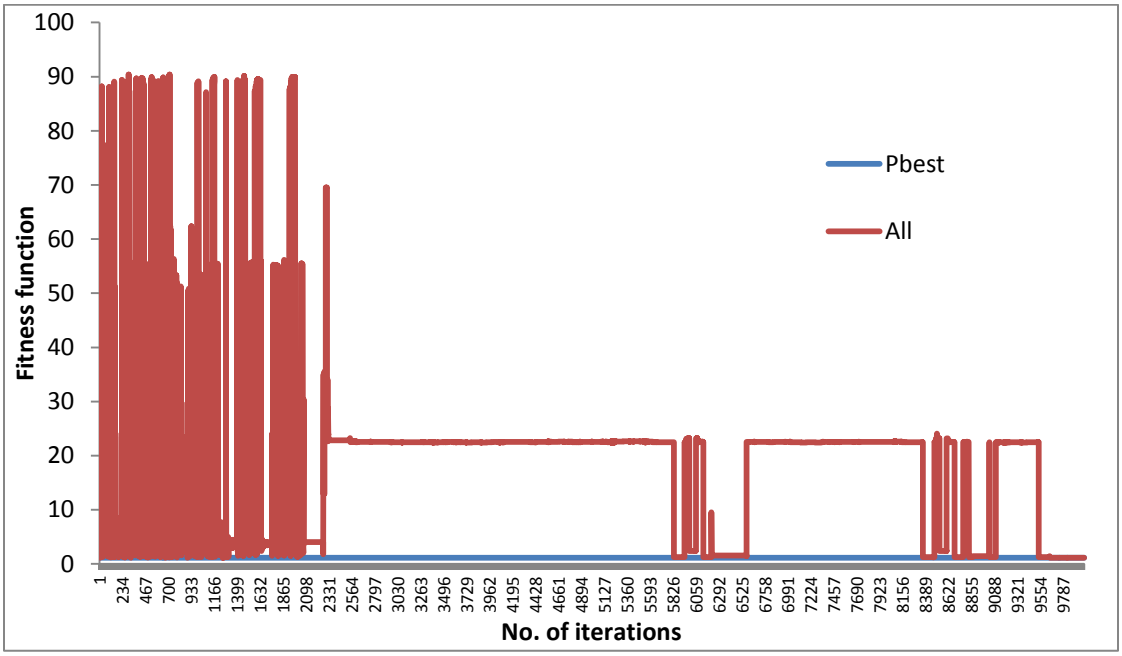


Fig.16 Fitness function plot of a single particle

The Fig.16 establishes the ability of the particle to move towards a global solution. While applying the algorithm to the NGNP model (3-ring VHTR), it is seen that solution space consists of many solutions that are very close to each other because of the relatively even flux distribution throughout the core for this model. The blue line is the memory of the local best of each particle. Each particle behaves differently for each of the iterations towards the final solution [14]. Fig.17 shown below is another sample particle with local best and the fitness function during each of the iterations.

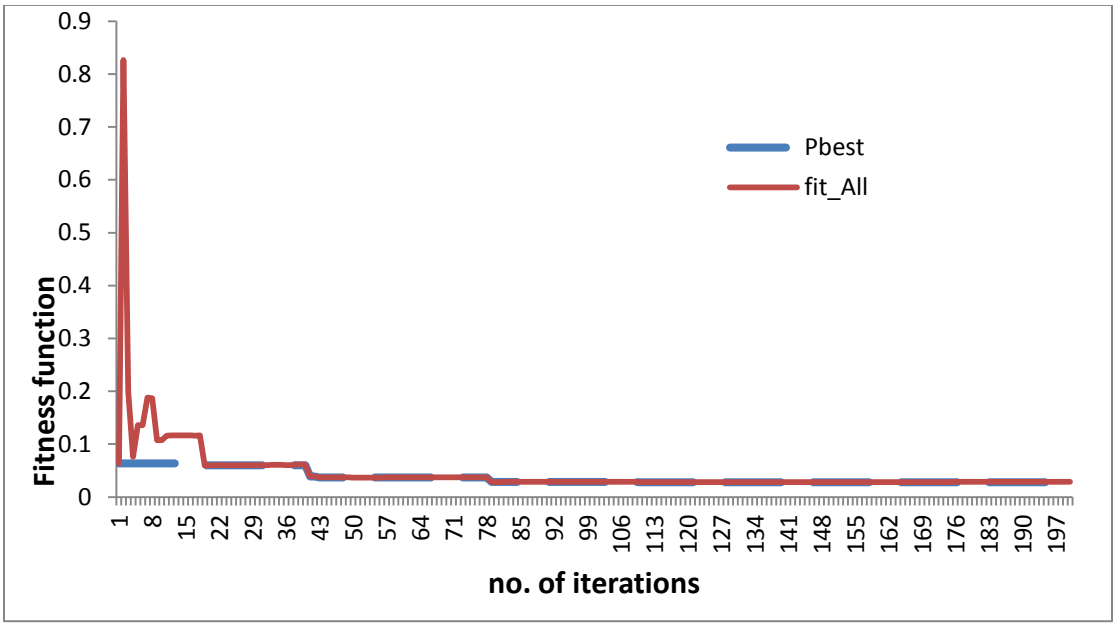


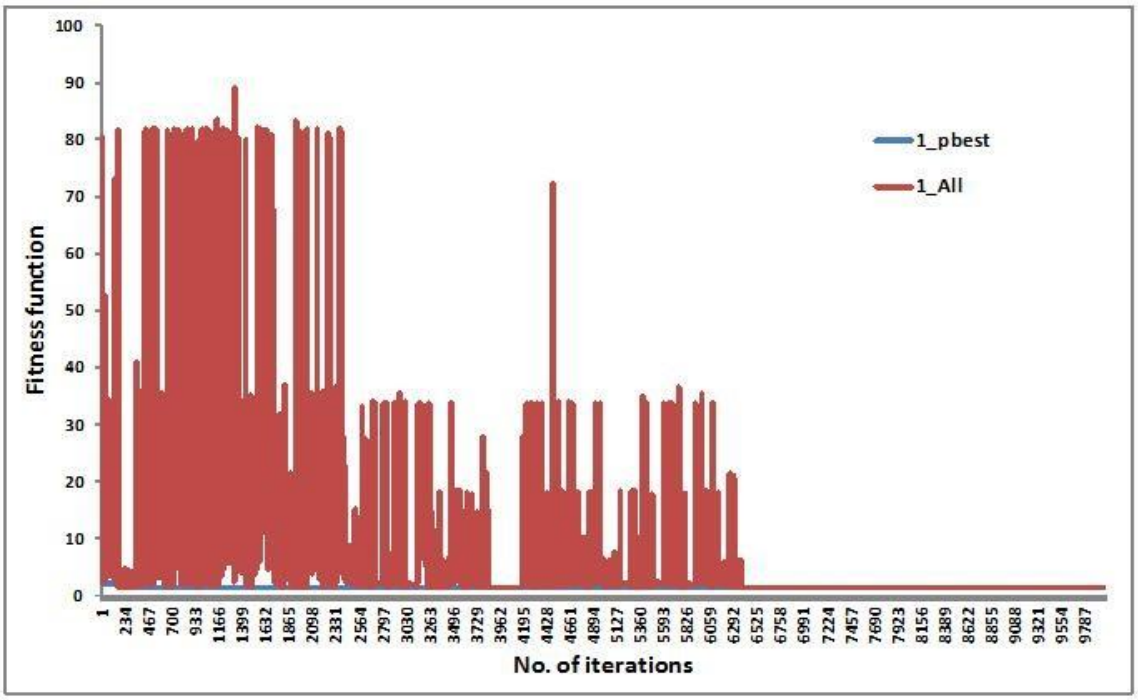
Fig.17 Sample 2 of a single particle behavior

According to [14], particles behave differently based on the direction and velocity it takes. Especially in the inertia weight method of PSO, the particle may converge to give a solution as in Fig.18 (a) or it may oscillate because there may be two solutions present that are equally optimal and equally closer to the particle's current solution as in Fig.18

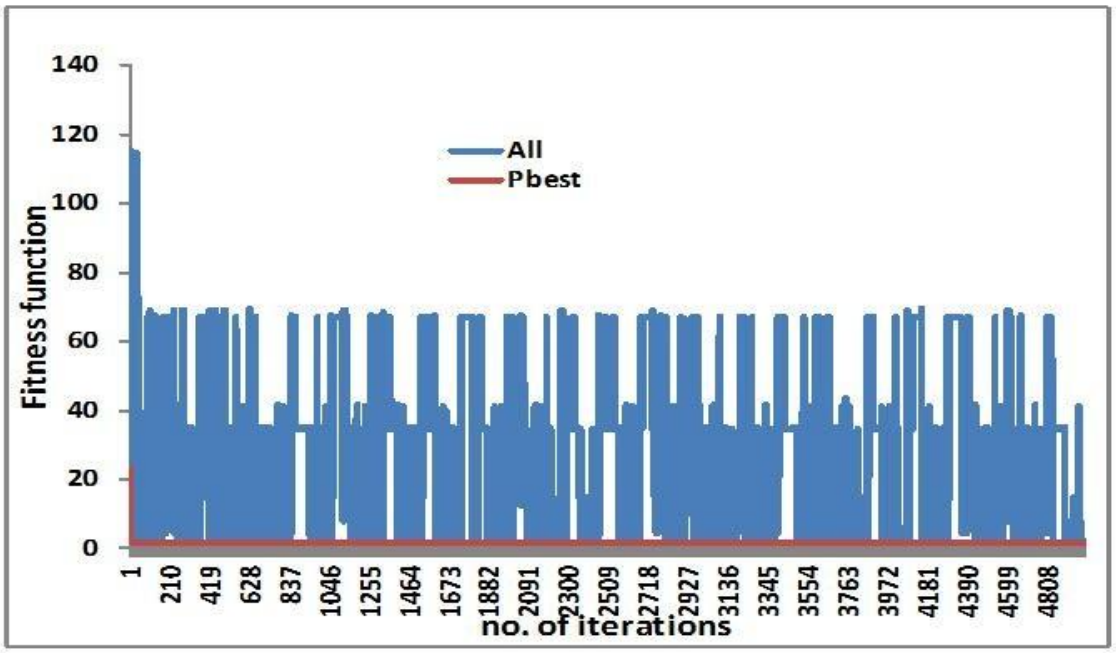
(b). Particle rescaling (Fig.18 (c)) is the phenomenon when the particle after converging on to a value for a number of iterations rescales itself when a new better solution is found in the neighborhood of the current solution and rescales towards the new optimal solution.

In Fig.18 (a and b), the legend "All" denotes that fitness function values of all the iterations are plotted and "Pbest" denotes that the fitness function values only the local best arrangement at each of the iterations is plotted. In (c), rescaling occurs after the particle converges at iteration 63 and rescales at iteration 373 and oscillates. These particle behaviors are common during global search after reaching a local solution [14].

In ReloPSO, the evaluation of each loading pattern, during each of the iterative steps of the algorithm, is not done by updating the reaction rate densities required for reactivity potential calculation as the burnup calculations are done by stochastic codes such as MCNPX. Hence, the number of iterations and particles can be as high as possible. One of the advantages of the PSO method is that it may not be necessary if the fitness function converges faster. By doing the same run many times, there is better chance of getting a better output than running it only once. The loading pattern with the minimum value for fitness function will be our choice of core arrangement.

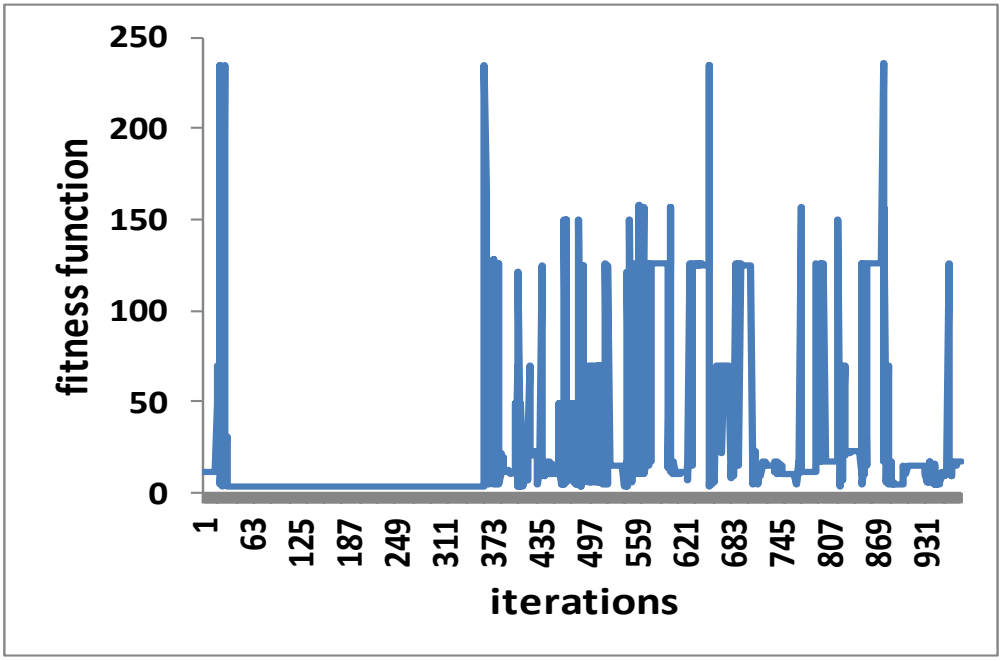


(a) Particle convergence



(b) Particle oscillation

Fig.18 PSO individual particle behavior



(c) Rescaling

Fig.18 Continued.

V.C. PARAMETER CALIBRATION

The next step is to find the algorithm response to different values for the algorithm parameters. Table xx shows the parameters that are tested, where the changes are done, type of values tested with and the algorithm response. The table serves two functions. First, it validates the algorithm and secondly it calibrates parameters. The number of agents (particles) and the dimension size variations are varied and it is found that the algorithm responds without errors as expected. But it is also found that the number of particles can be increased to increase the search space size. These two tests are done to

validate the flexibility of the algorithm to handle problems of different size, in this case different core configurations ranging from 3-ring core to 8-ring core.

By the algorithm definition, the weight parameter *w* can be a linearly decreasing function [13]. But the range it is varied is calibrated by testing the values. The accelerator variable $c_1 + c_2$ can be 4 or 4.1. The evaluation of the fitness function indicates that 4.1 is a better value towards optimization and that the values should be as close as possible ($c_1 = 2,\ c_2 = 2.1$). The most important parameter for calculating the fitness function is the maximum limit of the peaking factor value. The corresponding value of the localized reactivity potential of the maximum power peaking for VHTR is to be calculated for the maximum core power. For this algorithm, the $ppf_{\max} / ppf_{\lim}$ parameter is computed as $ppf / pf_{\max}$ where ppf is the ratio of maximum peaking in each pattern to average peaking for each loading pattern and $pf_{\max}$ is assumed as 1. As explained before, *V_max* is found to be specific and computed based on the reactivity potential. The results of the different tests performed on the algorithm are listed in Table 2.

Table 2 Algorithm test cases

No	Description	Variable	Values	Change in	Result
1	Changing the number of particles used in the algorithm	Agents	50 80 100 160	PSO.RUN	Works as expected
2	Change core configurations, that is change in the number of rings	Rings	3 5 8	PSO.RUN	Works as expected
3	Algorithm parameter - weight	Linearly decreasing function	09 - 0.2 0.9 - 0.4	ReloPSO.cpp	0.9 - 0.2 works better
4	Algorithm parameter accelerator variable	c_1+c_2	c1+c2=4.1 c1+c2 = 4	ReloPSO.cpp	c1+c2 = 4.1 works better
5	Algorithm parameter - particle velocity limit	MAXV	3 4 100	PSO.RUN	Problem specific; no pattern observed
6	Safety limit localized reactivity potential limit	ppf_max	1.01 - 2	PSO.RUN	It converges very soon for ppf_max greater than 1,1 for a 3 ring configuration as the reactivity potential values are very less

V.D. SHUFFLE RESULTS

The goals of a solution to a reloading problem optimization are to identify the best loading pattern and the point of shuffling during the core operation. The optimized shuffled loading pattern is obtained, as the direct output of the algorithm while the point

of shuffling requires further study into the algorithm output. During the algorithm, the fuel blocks are shuffled in 3 dimensions, which mean that any fuel block from any location can end up in any location within the $1/6^{th}$ of the core in which it is located at the end of the shuffling process in 3D. The localized reactivity parameter changes every time a fuel block is moved from one location to another. This value is estimated using equation 2.9.

The distribution of the values is observed by plotting the reactivity potential as a matrix between the original location and the moved location for every fuel block as shown in Fig.19 at 261 days and Fig.20 at 489 days of the core life. The reactivity potential is calculated from the reaction rates and material nuclide information extracted at those two points from the BOL MCNPX burnup run (core with similar fuel material at all locations).

The nature of the plot will provide an idea about the change a block is estimated to undergo when moved to any location within the core. A comparative image of the block reactivity at different time during core operation is also seen from the values and the distribution.

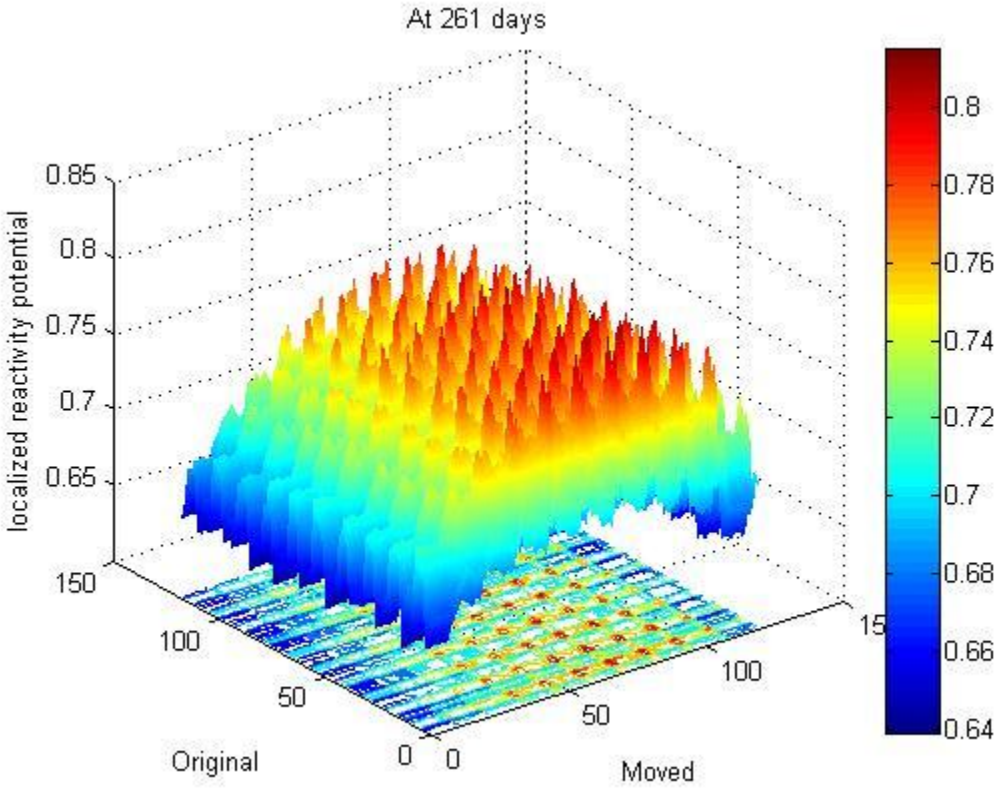


Fig.19 Reactivity potential for fuel blocks shuffled at 261 days

The core goes subcritical after 510 days. The plot exhibits the expected behavior; fuel blocks that are moved from the periphery to the center show peaks and the blocks moved from the center to periphery show relatively lower peaking. It is also seen that shuffle after 261 days, the reactivity potential is more evenly spread when compared to the shuffle after the 489 days. It can be seen from the maximum values and distribution in the color bar. This is due to the fact that the core is more burnt at 489 days and the reactivity potential shows more peaking as a result of uneven burnup of the core compared to the 261 days with relatively partial burnup.

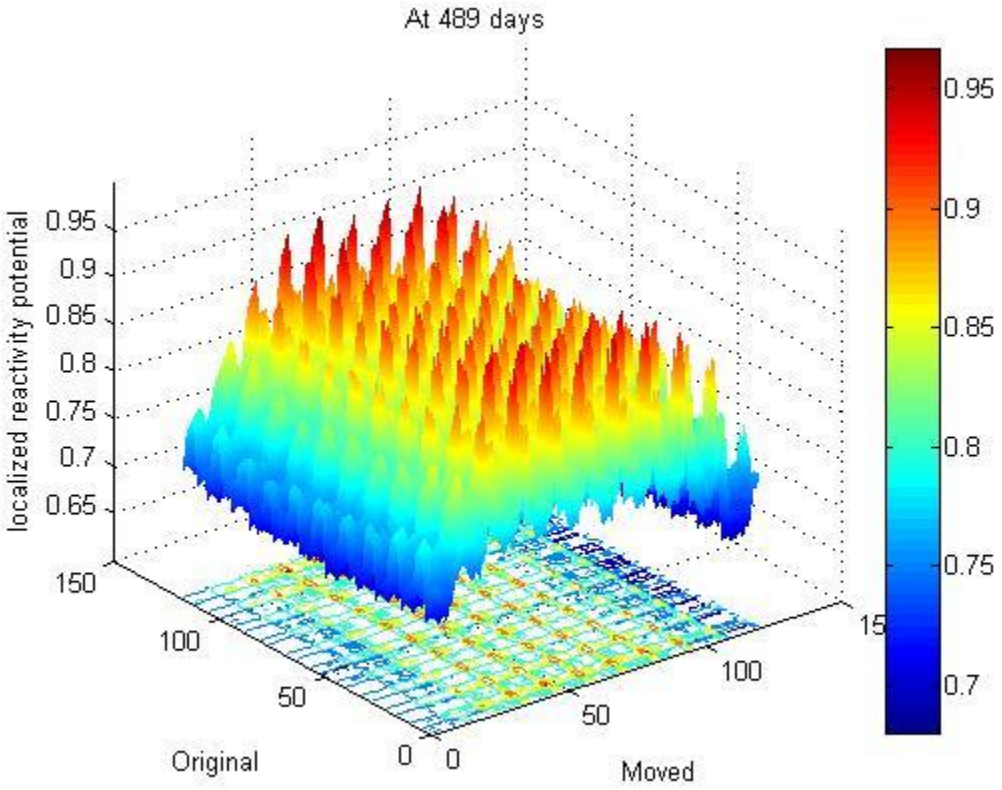
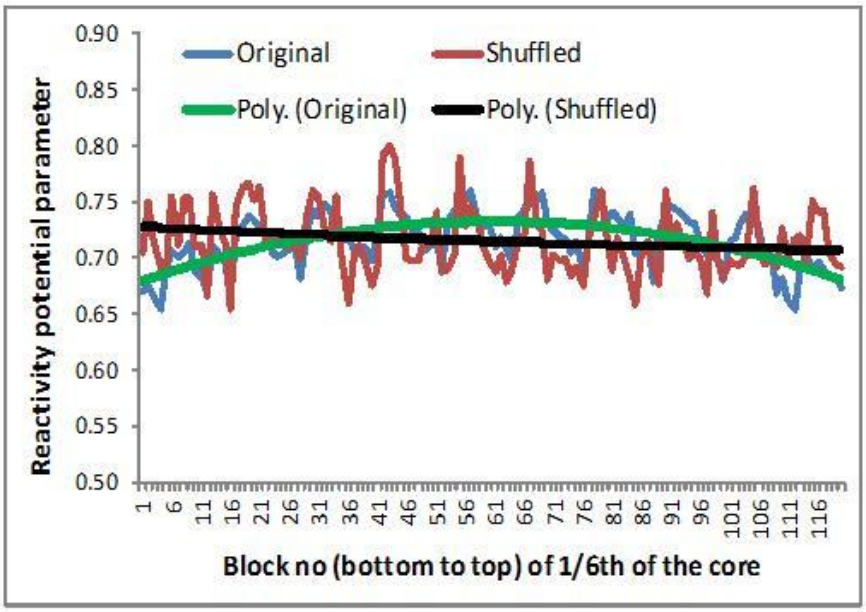


Fig.20 Reactivity potential for fuel blocks shuffled at 489 days

The reactivity potential parameter is plotted against each block to analyze the difference between its value before and after shuffle to verify if the new loading pattern is an improvement on the current loading pattern. From Fig.21 it can be seen that the initial distribution of the reactivity potential (shown in green trend line) shows peaking and the shuffled values give a flattened reactivity potential distribution. As the reactivity potential function parameter is directly proportional to the block power peaking, flattening of the reactivity parameter is analogous to the flattening of the peaking within the core. The plot appears distorted because the blocks are numbered from bottom to top sequentially and the blocks at the outer rings have lower potential than the inner ring

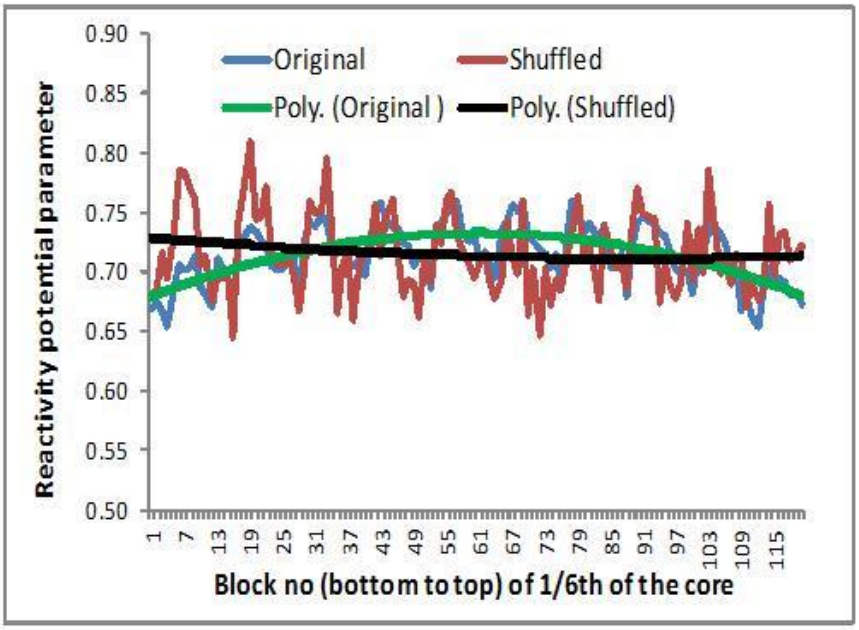
blocks. As the outer blocks receive more thermal neutrons from the reflector, they have a greater production. The reactivity parameter shows very little peaking because the core is at the middle of its lifetime and blocks at each ring are burnt relatively uniform. The relatively uniform flux distribution over the 3-ring annular NGNP model is also a reason for this behavior. For DB-VHTR where the core is a minimum of 5 rings upto 8 rings with no central reflectors, the peaking will be higher. For this 3-ring core, when the ReloPSO algorithm is applied with the current core as one of the inputs, the current core appears to be the arrangement with least peaking.

When the algorithm is run without the current arrangement as loading pattern, it can be seen from the fitness function values that the final loading pattern is in the neighborhood of the current core. In addition, the reactivity parameter peaking is reduced after applying the algorithm, given that the estimated values of the reactivity potential is obtained under the assumption that the reaction rates at each location remains constant. When the final configuration is burnt in Vesta coupled with MCNP5 1.5, the k_{eff} remains almost the same for the remainder of the core life for the shuffle after 261 days (Fig.22).



(a)

Particles	50
iteration	10000
fitness function	
original	0.303639
shuffled	0.33646



(b)

Particles	100
iteration	10000
fitness function	
original	0.303639
shuffled	0.339837

Fig.21 Reactivity potential before (from MCNP) and after (estimated) shuffling at 261 days

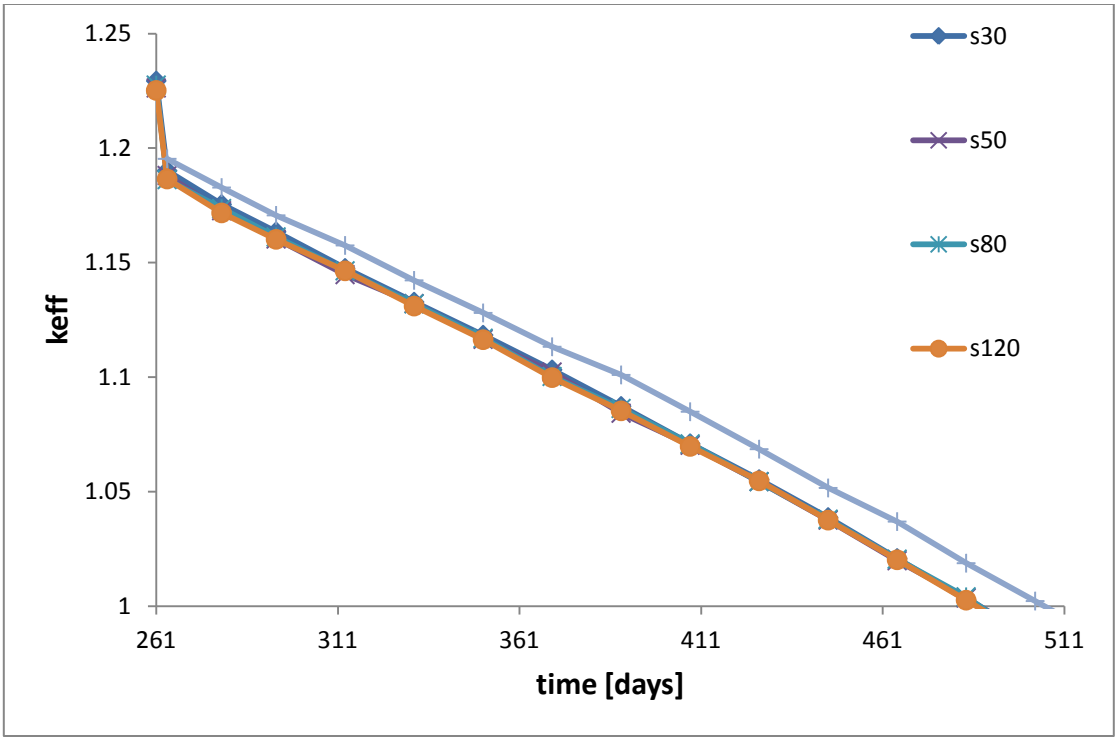


Fig.22 k_{eff} vs time for shuffle at 261 days

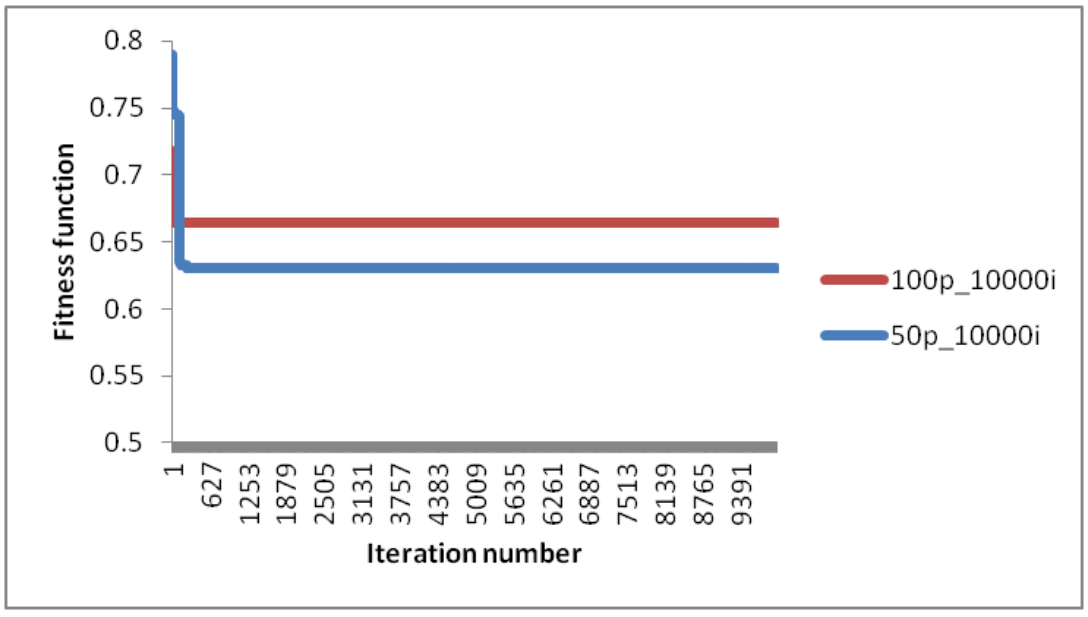


Fig.23 Fitness function plot for shuffle after 489 days

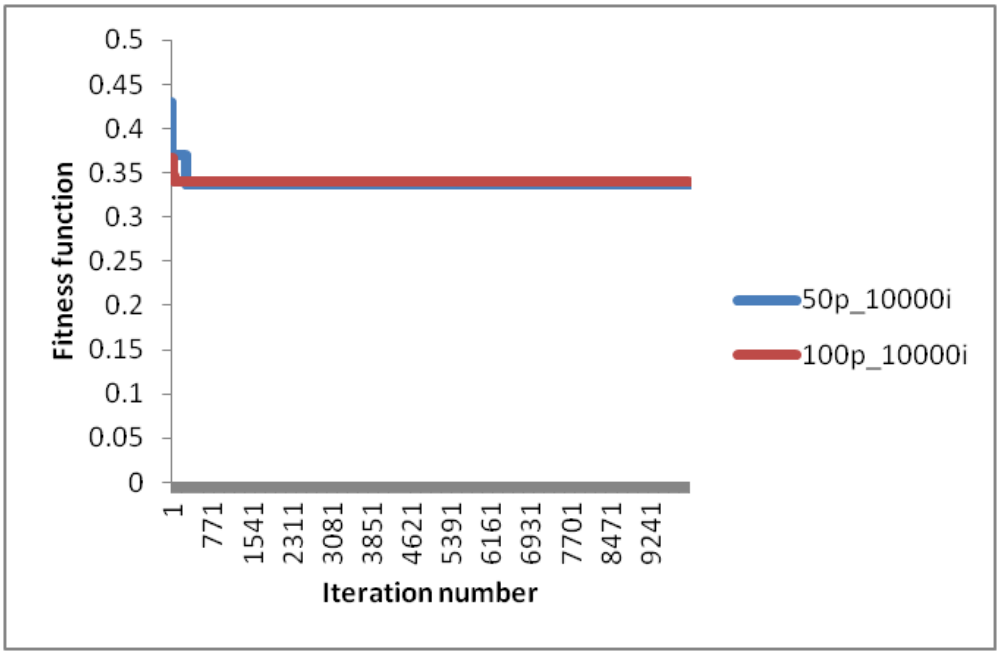
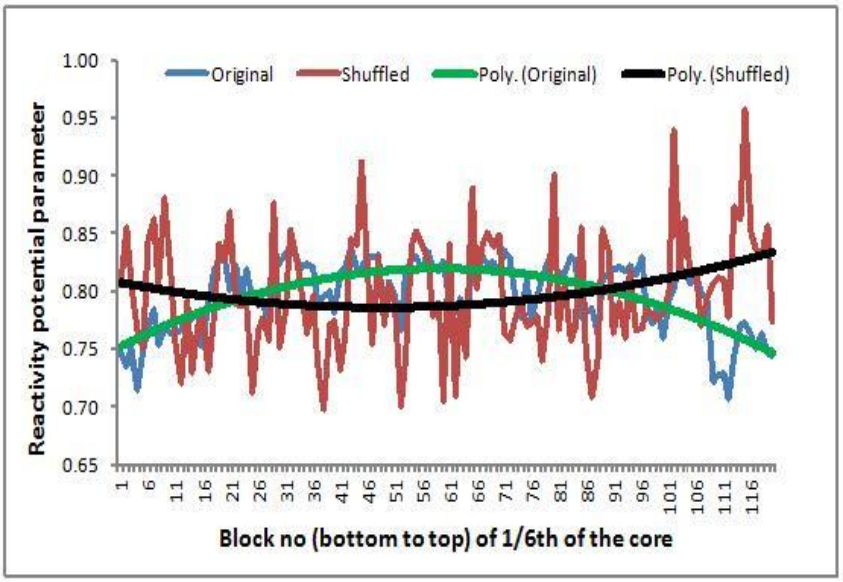


Fig.24 Fitness function plot for shuffle after 261 days

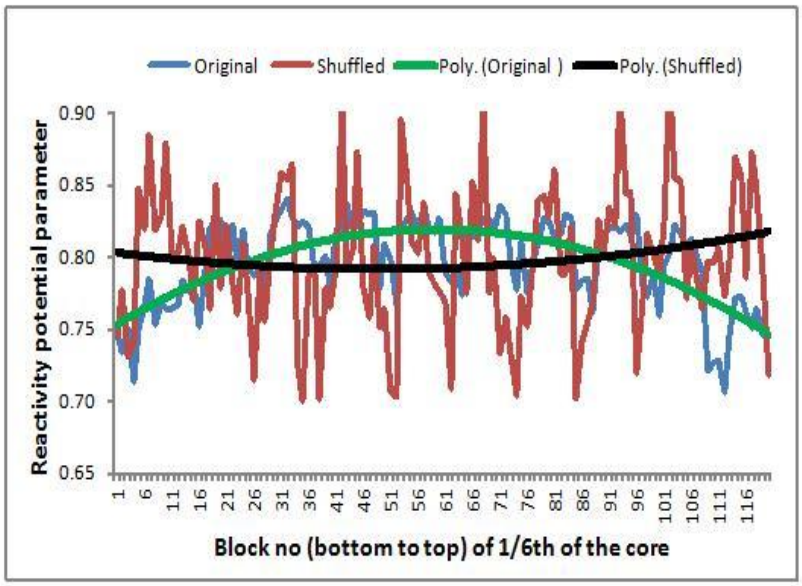
The simulations are done with 50 particles and 100 particles for 10000 iterations each. It can be seen that the fitness function for the 261 days run is very close to the original pattern value.

The plot in Fig.24 is from day 261 when the core is decayed for 22 days after shuffle and then burnt till after it reaches subcritical for different number of particles for the ReloPSO algorithm (30, 50, 80, 120). The drop in the keff shows the decay step of the burnup.



(a)

Particles	50
iteration	10000
fitness function	
original	0.366998
shuffled	0.630302



(b)

Particles	100
iteration	10000
fitness function	
original	0.366998
shuffled	0.664097

Fig.25 Reactivity potential before (from MCNP) and after (estimated) shuffling at 489 days

When the core is shuffled at 489 days, the reactivity potential plot before and after optimization looks to be peaking in the opposite direction (Fig.25). When the initial core is included in the input, it came out as the final configuration further confirming that for the NGNP model, the reactivity potential is relatively uniform. Interestingly, when the shuffled core is burnt using MCNP, there is an increase in keff and it is sustained till the core reaches subcriticality (Fig.26). This burn run does not include any decay step.

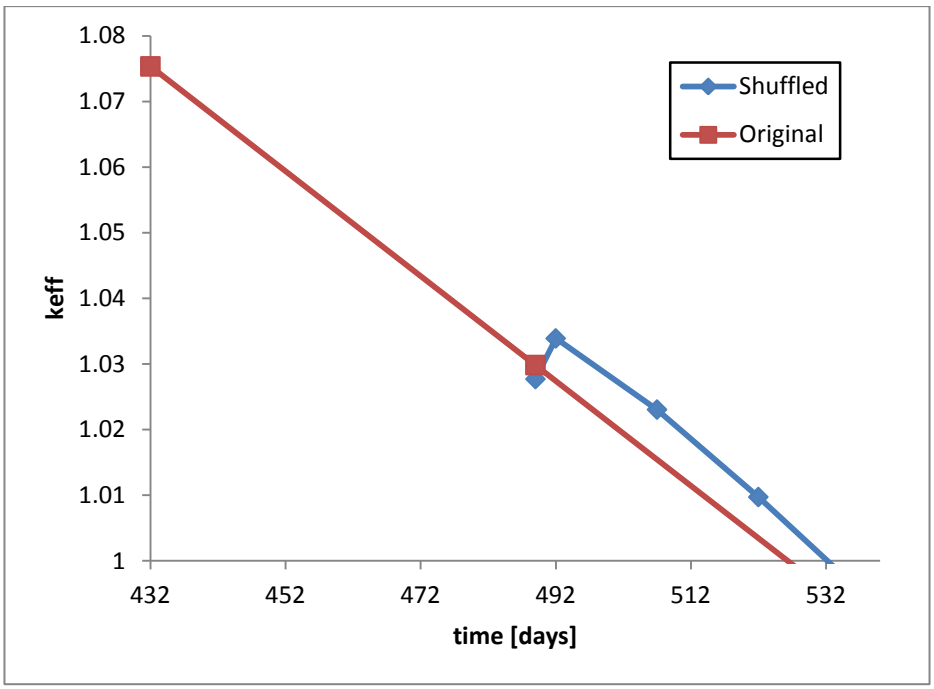


Fig.26 k_{eff} vs time for shuffle at 261 days

From the above phenomenon, it is understood that the core life can be increased and the reactivity potential peaking can be reduced by using the algorithm at the correct time step. If the core life can be increased more than the time required for reloading the core,

then additional power can be generated from the same core and more TRUs can be burnt (DB-VHTR).

It can also been from the reactivity parameter block plots Fig.19 and Fig.20 that the difference in range is from 0.65 to 0.75 (for 261 days) and from 0.7 to 0.9 (for 489 days). This may be due to absorption reactions dominating the neutron production reactions late in the core life. Since the core peaking is very less in the above two scenarios, if the algorithm is applied to a core with localized reactivity potential peaking, the effect of the algorithm can be tested. A core with peaks at 261 days is taken as sample and the algorithm is tested (Fig.27).

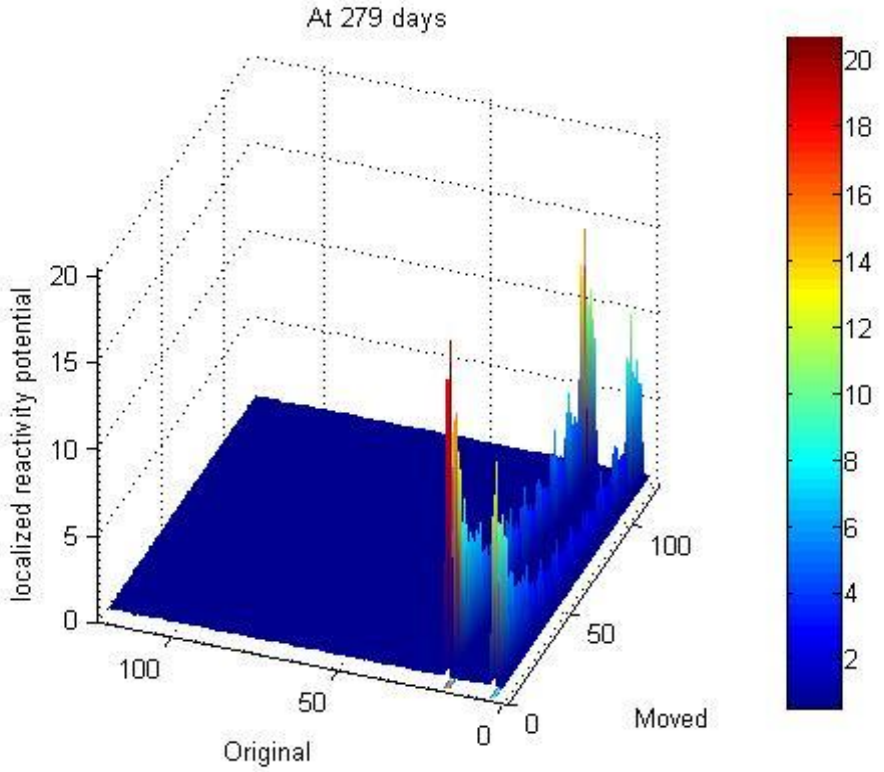


Fig.27 Reactivity potential for fuel blocks with uneven peaking

The above plot is the reactivity potential plot when a couple of blocks are changed to get uneven peaks. From the plot, it can be seen that the algorithm searches towards the loading pattern that minimizes the peaking. The peaks still exist in the core because the peaking blocks are not replaced. The curve fit for the algorithm shows that the shuffled loading pattern has better reactivity potential distribution within the core. The value of the peaks are reduced and hence reduction the net peaking. The trend lines used are quadratic polynomial fits to show the trend in the curvature (Fig.28).

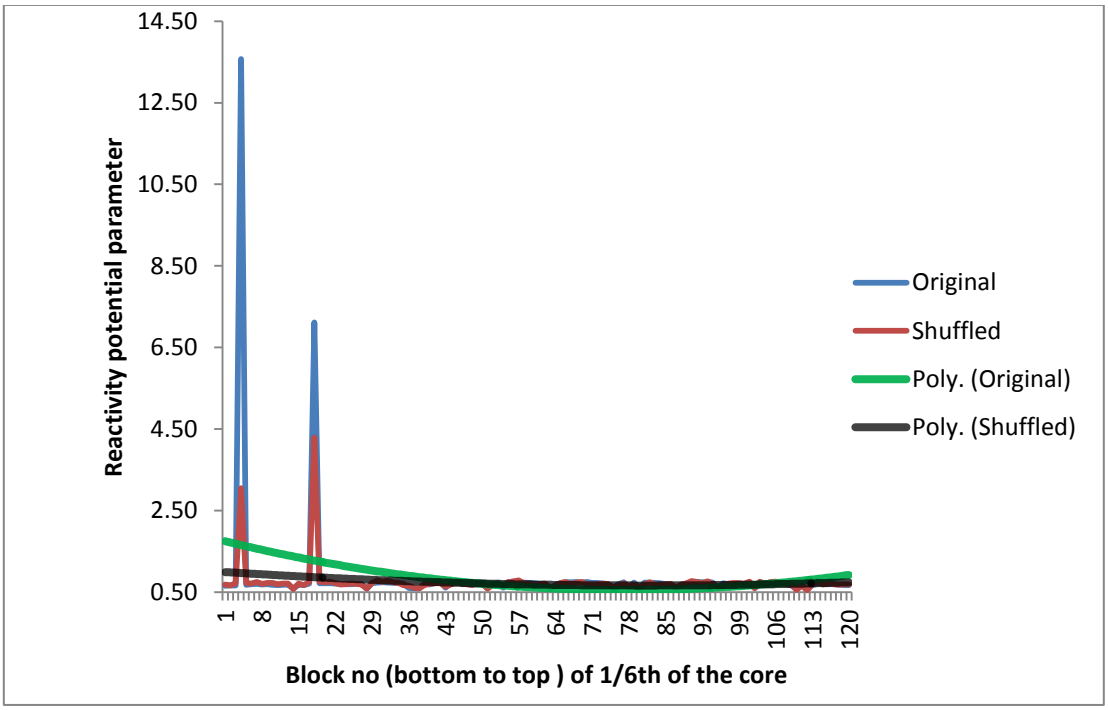


Fig.28 With peaks in the reactivity parameter

V.E. DISCUSSION

The purpose of the algorithm is to work on the combinations of the loading and obtain a solution that has the minimal constraining parameter value, the reactivity potential peaking. The reactivity potential estimation method is used as a means to circumvent the need to simulate extremely time consuming burnup runs during every shuffle using Monte Carlo methods. But even with a search algorithm, the reactor characteristics such as the atom density, flux change and the reaction rate are required for optimizing the multi-dimensional loading process. Given the assumptions made, such as the elimination of leakage and the constant reaction rates, the actual core performance is expected to be different from the optimization process. But without this step, the algorithm could not have been developed for the problem as the parameter reflects the change in the block characteristics before and after shuffle. If this parameter is not developed, the algorithm would have random numbers to work on with which the reactor response cannot be simulated during the shuffling process. The importance of the parameter cannot be stressed more.

The shuffled core at 261 days is burnt on MCNP just for one time step because the time constraint due to the exact geometry core model to compare the reaction rate densities from the MNCP runs to that of the estimated values.

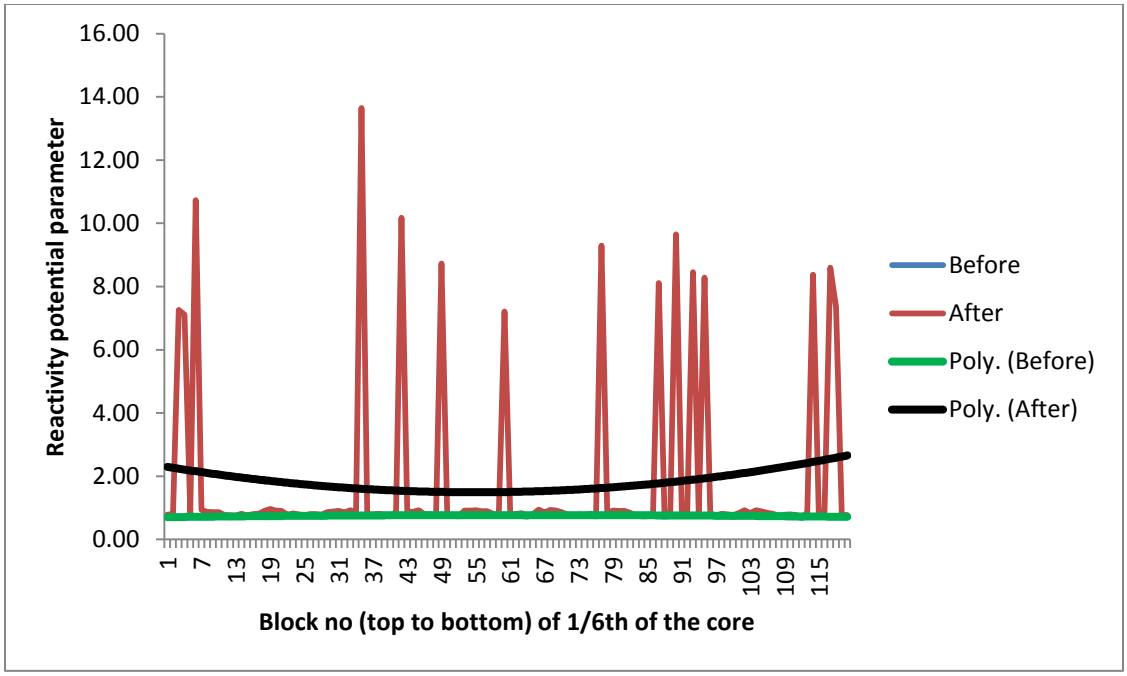


Fig.29 Reactivity parameter before and after shuffle from MCNP burns

The one time step run is a 3 days burn with the shuffled core without decay. As seen from Fig.29, there appear huge peaks of the reactivity parameter that are not a set of blocks but 12 blocks among the 120 fuel blocks. The reason for the peaks may be they are reaction rate calculation noise from MCNP or the estimation process does not reflect the actual values of reactivity parameter. When the reactivity parameters are plotted without the outliers, the profile is flat as expected (Fig.30).

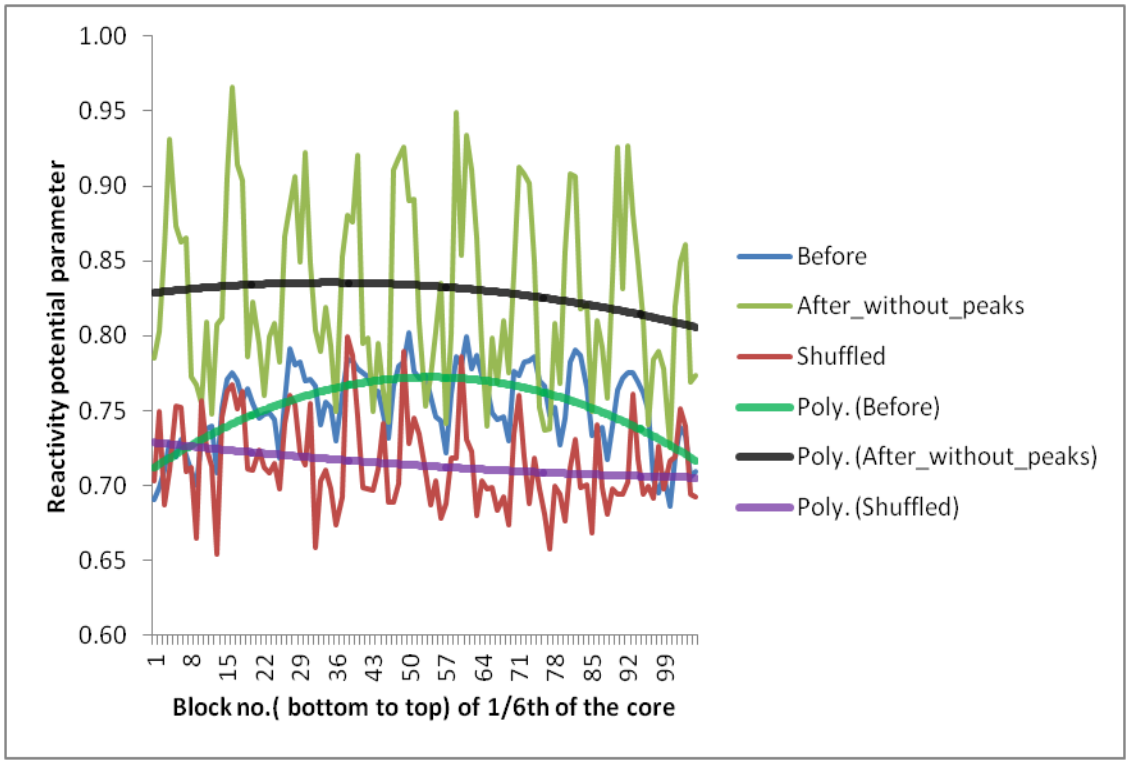


Fig.30 Reactivity parameter before and after without the peaking blocks

It is also seen that the reactivity after shuffle is an order greater than the estimated value when the outliers are removed (Fig.30), but without analysis the reason for the difference or the relationship between the curves cannot be determined. Further analysis is required to establish any connection between the estimated and calculated reactivity potential values. The purpose of the algorithm is to optimize the shuffling process based upon any core parameter that can either be estimated for a change in location or can be calculated deterministically. When the optimization is performed based on such a parameter, insight into the relationship between the calculated and estimated parameter can be drawn with more history and analysis.

CHAPTER VI

CONCLUSION AND FUTURE WORK

The project is aimed towards developing a code for shuffling the fuel blocks in three dimensions to get a flattened power profile. Instead of optimizing the power peaking factor, a parameter called the localized reactivity potential is developed based on the reaction rates. For simplification and calculation of the parameter without depending on the time consuming MCNP simulations, it is assumed that the reaction rates at each block location within the core is assumed not to change during each shuffle. Hence by changing the inventory of nuclides at each location, the reactivity parameter is calculated.

The VHTR has a hexagonal core with prismatic fuel blocks and hence the symmetry of the reactor is utilized for optimization. By optimizing $1/6^{th}$ of the core, the same pattern is mirrored for the remaining $5/6^{th}$ of the core. The NGNP model of the VHTR uses a 3-ring core configuration with 170 blocks in each of the $1/6^{th}$ of the reactor arranged in layers of 17. In each layer, there are 5 blocks with control rod insertion facility and they cannot be moved to a regular fuel block location. This results in a combinatorial problem with a solution space size of 120! combinations of the fuel assemblies. The purpose of the algorithm is to search for optimal arrangement in this solution space.

A stochastic population based algorithm called the Particle Swarm Optimization that is developed based on the social behavior of birds is used. The algorithm is developed using C and the simulations for the core are performed using MCNP burn runs coupled

to Vesta or MCNPX burn runs. From the fitness function discussion, the algorithm performance and ability is established. The code is tested for different scenarios and the parameters are optimized. The algorithm is also flexible to optimize any number of rings (3 to 8) and hence a bigger problem set. The algorithm is metaheuristic and hence it is executed multiple times at to improve the current solution. The Pareto set of output loading patterns are then analyzed and the best loading pattern is chosen based on the criteria.

Analysis of the effect of algorithm on VHTR operation is a continuing work on this project. A detailed analysis on the shuffle output is required for adapting the ReloPSO code into an effective reactor design and shuffling optimization. Due to time constraints and scope of the thesis, the analysis using neutronics codes is not performed in detail. But the outputs obtained so far give insight and direction towards the next phase of the project.

(i) The first important view is to rationalize the reactivity potential parameter towards a representative quantity with additional information. If not for this step, the shuffling algorithm should be coupled to a deterministic neutronics code that simulates each arrangement of the shuffling process. The additional information includes the addition of leakage quantity to the neutron loss function in the reactivity potential parameter:

By adding the leakage to the numerator by either using surface current meshes to find the net current at each location based on the neighbor blocks

or by adopting a deterministic method such as using the buckling and leakage parameters, the reactivity potential parameter can be improved.

(ii) A method to find the safe limit of the reactivity potential for the core configuration to be used should be adopted. This value will be based on the limiting power peaking of the core.

(iii) The fitness function can be changed to improve better representation of the parameters that affect the core performance. For the LWRs, the two dimensional shuffling is suffice because of the intuitive knowledge about the type of blocks to be located at suitable locations. For example, radial power peaking is considered so that the fuel assemblies are moved to reduce the peaking. In block type fuel cores such as the VHTR model, there exists the lack of intuitive knowledge of how the core performs under different fuel types at different locations. The axial peaking should also be considered to refine the search algorithm.

(iv) Shuffling at various time steps of the core will give an understanding of the time to shuffle so that a most favorable time can be identified that would optimize the core operation. This is a time consuming study, as the core has to be burnt at different time steps. Each burn run of a whole core exact geometry model takes more than two weeks to completely burn the core. This can be improved by migrating to Serpent or Vesta coupled with MCNP. But even then for a high fidelity burn run, the time required is estimated to be more than a month. By studying the core shuffling outputs at different time

steps, impact of the other parameters such as the fission product composition, change in flux, etc. can be identified and the algorithm can be improved.

(v) By adding fresh fuel blocks to the core, the performance can be improved. Identification of the blocks to be replaced based on their burnup level is required.

(vi) The study of the effects of shuffling using TRU fuel for a sustained core operation can be developed from the algorithm (DB-VHTR).

(vii) Further analysis on the burn runs of shuffled core is required for improving the algorithm to adapt for changing reactor configurations.

The ReloPSO algorithm performs the function of minimizing algorithm acting on a NP hard combinatorial problem with results closer to the optimum. As a continued project this algorithm can be effectively converted into a design tool and analysis for block type reactors and when it is improved further by adding features to operate with other deterministic codes, it will be a fully operational shuffling tool. If the assumptions made to simplify the estimation of the reactivity potential can be overcome by mathematically modeling the flux behavior or statistic data of history of runs, the code can be a very fast and useful tool in the shuffling process.

REFERENCES

[1] NUCLEAR ENERGY INSTITUTE (NEI), "Resources and Stats" www.nei.org/resourcesandstats/nuclear_statistics. Accessed on February 10, 2012.

[2] U.S. DOE NUCLEAR ENERGY RESEARCH ADVISORY COMMITTEE AND GEN IV INTERNATIONAL FORUM, "Technology Roadmap for Generation IV Nuclear Energy Systems" www.gen-4.org/Technology/roadmap.htm. Accessed on February 10, 2012.

[3] UNITED STATES GOVERNMENT ACCOUNTABILITY OFFICE, "DOE Nuclear Waste" www.gao.gov/products/GAO-11-230. Accessed on February 11, 2012.

[4] C.RODRIGUEZ, A.BAXTER, D.MCEACHERN, M.FIKANI and F.VENNERI, "Deep-Burn: Making Nuclear Waste Transmutation Practical," *Nuclear Engineering and Design*, **2805**, 1-19 (2003).

[5] G.BACCAGLINI, S.BALL, T.BURCHELL, T.FEWELL, M.LABAR, et al., "Very High Temperature Reactor, Survey of Material Research and Development Need to Support Early Deployment," INEEL/EXT-03-0014 (2003).

[6] Y.KIM, C.KEUN JO and J.MAN NOH, "Optimization of Axial Fuel Shuffling Strategy in a block-type VHTR," American Nuclear Society International Meeting and Nuclear Technology Expo, Washington D.C., 406-407, (2007).

[7] E.H.L. AARTS and J.K. LENSTRA, *Local Search in Combinatorial Optimization*, Princeton University Press (2003).

[8] P.J.M. VAN LAARHOVEN and E.H.L. AARTS, *Simulated Annealing: Theory and Applications*, D.Reidel Publishing Company (1992).

[9] I.M.S.OLIVEIRA and R.SCHIRRU, "Swarm Intelligence of Artificial Bees Applied to In-Core Fuel Management Optimization," *Annals of Nuclear Energy*, **Vol. 38**, 1039-1045, (2011).

[10] S.GLASSTONE and A.SESONSKE, *Nuclear Reactor Engineering*, Van Norstland Reinhold Company (1967).

[11] A.A.M.MENESES, L.M.GAMBARDELLA and R.SCHIRRU, "Particle Swarm Optimization Applied to the Nuclear Reload Problem of Pressurized Water Reactor," *Progress in Nuclear Energy*, **51**, 319-326, (2009).

[12] A.A.M.MENESES, L.M.GAMBARDELLA and R.SCHIRRU, "A New Approach for Heuristics-guided Search in the In-Core Fuel Management Optimization," *Progress in Nuclear Energy*, **52**, 339-351, (2010).

[13] J.KENNEDY and R.C.EBERHART, "Particle Swarm Optimization," *Proc. IEEE International Conference on Neural Networks* (Perth, Australia), IEEE service Center, Piscataway, NJ, **IV**, 1942-1948, (1997).

[14] J.KENNEDY and R.C.EBERHART, *Swarm Intelligence*, Morgan Kauffman Publishers (2001).

[15] F.KHOSHAHVAL, A.ZOLFAGHARI, H.MINUCHEHR, M.SADIGHI and A.NOROUZI, "PWR Fuel Management Optimization Using Continuous Particle Swarm Intelligence," *Annals of Nuclear Energy*, **37**, 1263-1271, (2010).

[16] I.BHATTACHARYA and S.SHUKLA, "Parameter Selection and Performance Study in Particle Swarm Optimization," *Proc. International Conference on Modeling, Optimization, and Computing (ICMOC 20110)*, Durgapur, India. October 20-30, 2010. American Institute of Physics Conference Proceedings, **Vol. 1298**, (2010).

VITA

Name:	Sathish Kumar Lakshmipathy
Address:	Department of Nuclear Engineering, Texas A&M University, 3133 TAMU, College Station, TX 77843-3133
Email Address:	la.sathish@gmail.com
Education:	B.E. Mechatronics Engineering, Anna University, 2007 M.S., Nuclear Engineering, Texas A&M University, 2012