2009 International Nuclear Atlantic Conference - INAC 2009
Rio de Janeiro,RJ, Brazil, September27 to October 2, 2009
ASSOCIAÇÃO BRASILEIRA DE ENERGIA NUCLEAR - ABEN
ISBN: 978-85-99141-03-8

PARALLEL PARTICLE SWARM OPTIMIZATION ALGORITMS IN NUCLEAR PROBLEMS

Marcel Waintraub[1], Roberto Schirru [2] and Cláudio M.N.A. Pereira [3]

[1] Instituto de Engenharia Nuclear, IEN - CNEN
Rua Hélio de Almeida, 75
21941-972 Rio de Janeiro, RJ
marcel@ien.gov.br

[2] Laboratório de Monitoração de Processos, LMP – PEN/COPPE
Av. Horácio Macedo 2030
21941-914 Rio de Janeiro, RJ
schirru@lmp.ufrj.br

[3] Instituto de Engenharia Nuclear, IEN - CNEN
Rua Hélio de Almeida, 75
21941-972 Rio de Janeiro, RJ
cmnap@ien.gov.br

ABSTRACT

Particle Swarm Optimization (PSO) is a population-based metaheuristic (PBM), in which solution candidates evolve through simulation of a simplified social adaptation model. Putting together robustness, efficiency and simplicity, PSO has gained great popularity. Many successful applications of PSO are reported, in which PSO demonstrated to have advantages over other well-established PBM. However, computational costs are still a great constraint for PSO, as well as for all other PBMs, especially in optimization problems with time consuming objective functions. To overcome such difficulty, parallel computation has been used. The default advantage of parallel PSO (PPSO) is the reduction of computational time. Master-slave approaches, exploring this characteristic are the most investigated. However, much more should be expected. It is known that PSO may be improved by more elaborated neighborhood topologies. Hence, in this work, we develop several different PPSO algorithms exploring the advantages of enhanced neighborhood topologies implemented by communication strategies in multiprocessor architectures. The proposed PPSOs have been applied to two complex and time consuming nuclear engineering problems: i) reactor core design (CD) and ii) fuel reload (FR) optimization. After exhaustive experiments, it has been concluded that: i) PPSO still improves solutions after many thousands of iterations, making prohibitive the efficient use of serial (non-parallel) PSO in such kind of realworld problems and ii) PPSO with more elaborated communication strategies demonstrated to be more efficient and robust than the master-slave model. Advantages and peculiarities of each model are carefully discussed in this work.

1. INTRODUCTION

About two decades ago, due to high complexity of most optimization problems involved in nuclear reactors optimization, metaheuristics have been proposed in substitution of traditional gradient-based techniques. Some years later, due to advances in processor technologies, which provided faster computation, more robust, but time-consuming techniques, such as the population-based metaheuristics (PBMs) could be applied.

Recently, Particle Swarm Optimization (PSO) [1], a PBM inspired in social adaptation models, has been demonstrating to be a good alternative for solving complex engineering problems and overcoming other PBM in several nuclear engineering optimization problems

[3-6]. PSO has gained popularity due to its robustness (less dependent of parameter adjustment), efficiency (better results in less time) and simplicity. Moreover, it generally needs less computational efforts when compared to other PBM such as GA or other evolutionary algorithms. Nevertheless, not differently from other PBM, PSO may still find problems with time-consuming objective functions. In this cases, parallel and distributed models are very useful and, sometimes, mandatory. The default advantage of PPSO is the reduction of computational time. Such characteristic can be easily obtained by master-slave models, in which only the fitness evaluation is calculated in parallel. However, as occur with other PBM, the use of more elaborated communication (neighborhood) strategies propitiates more diversity and consequently, robustness, enhancing the chances of reaching better results.

Chang et al. [7] proposed a coarse-grained approach, outlining the gains, due to the communication strategies, over the traditional (serial) PSO. In that work, however, experiments have been made only with numerical benchmark functions. Few real-world applications of PPSO are also seen in literature. Schutte et al. [8] have applied a master-slave approach to biomechanical Q3 system identification. Good results have been found from engineering point of view. Jin and Samii [9] developed a PPSO for antenna designs. They also used a master-slave approach. Venter and Sobieski [10] focused the speedup due to the use of an asynchronous PPSO applied in the optimization of a typical transport aircraft wing.

A common characteristic observed in above-mentioned real-world applications is the focus on speedup due to parallel processing. Gains due to more elaborated communication strategies are poorly investigated in real-world problems. Hence, motivated by: i) the good performance of PSO in nuclear problems, ii) the time-consuming objective functions involved in such problems and principally iii) the improvement in PSO provided by the use of more elaborated communication strategies [7] [11], this work is aimed to develop PPSO models with different communication strategies for application to real-world nuclear engineering problems. The focus here is not only the speedup itself, but principally improvements in problems' solutions due to the gains in terms of efficiency and robustness, provided by the use of proposed communication strategies.

To accomplish that, four multiprocessors based PPSO approaches, ranging from coarse- to fine-grained, have been developed and applied to two classical nuclear reactor optimization problems: i) reactor core design (CD) and ii) fuel reload (FR) optimization.

The main objective of this work is to analyze the performance of the proposed PPSO in real-world time-consuming nuclear engineering problems, investigating advantages and peculiarities of each PPSO model, outlining the gains due to each parallel approach, not only in terms of speedup, but in the optimization outcome itself.

Next session gives an overview on the standard PSO algorithm and classical models of parallelism. Session 3 describes the proposed PPSO, while Session 4 describes the optimization problems. In Session 5, PPSO applications and results are discussed and, finally, Session 6 presents the concluding remarks.

2. PARTICLE SWARM OPTIMIZATION ALGORITHM OVERVIEW

2.1. Standard PSO

Particle Swarm Optimization (PSO) is an optimization metaheuristic inspired by the behavior of biological swarms and social adaptation. In PSO, a swarm of structures encoding solution candidates (“particles”) “fly” in the n-dimensional search space of the optimization problem looking for optima or near-optima regions. The position of a particle represents a solution candidate itself, while the velocity attribute, provides information about direction and changing rate. Particles are guided by two components: i) cognitive information based on particles’ own experience and ii) social information based on observation of neighbors. Let $\vec{X}_i(t) = \{x_{i,1}(t), \cdots, x_{i,n}(t)\}$ and $\vec{V}_i(t) = \{v_{i,1}(t), ..., v_{i,n}(t)\}$ be, respectively, the position and the velocity of particle i in time t, in an n-dimensional search space. Considering that $\overrightarrow{pBest_i}(t) = \{pBest_{i,1}(t), \cdots, pBest_{i,n}(t)\}$ is the best position already found by particle i until time t and $\overrightarrow{gBest_i}(t) = \{gBest_{i,1}(t), \cdots, gBest_{i,n}(t)\}$ is the best position already found by a neighbor until t, the PSO updating rules for velocity and position are given by

$$v_{i,n}(t+1) = w.v_{i,n}(t) + c_1.r_1.(pBest_{i,n}(t) - x_{i,n}(t)) + c_2.r_2.(gBest_{i,n}(t) - x_{i,n}(t)) \quad (1)$$

$$x_{i,n}(t+1) = x_{i,n}(t) + v_{i,n}(t+1) \quad (2)$$

where r1 and r2 are random numbers between 0 and 1. Coefficients c1 and c2 are given acceleration constants towards $\overrightarrow{pBest}$ and $\overrightarrow{gBest}$ respectively and *w* is the inertia weight.

The inertia weight, w, is responsible for the scope of the exploration of the search space. High values of *w* promote global exploration and exploitation, while low values lead to local search. A common approach to provide balance between global and local search is to linearly decrease w during the search process. The swarm is randomly initialized. Then, while stopping criterion is not reached (here we used a fixed number of iterations), particles are evaluated (in this part a reactor simulator code is called), *pBest* and *gBest* are updated and then, particles move according to velocity and positions’ equations (Equations (1) and (2)).

3. PROPOSED PPSO MODELS

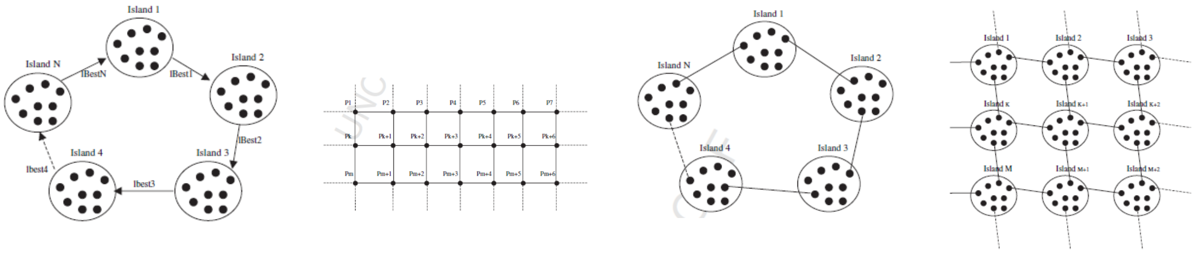
Parallel genetic algorithms (PGA) [12] are the most explored parallel PBM. The concepts used, however, may be extrapolated (with adaptations) to other PBM. Three of the proposed PPSO are inspired in traditional parallel GA (PGA) models. The master-slave, island (ring topology) and cellular PPSO models were adapted from PGA. Another PPSO based in a new different approach has been proposed. The idea is to use concept of neighborhood in PSO, to connect islands, avoiding necessity of defining the “migration interval” parameter. It has been called here Neighborhood-Island PPSO.

The master-slave PPSO is the simplest one, in which evaluations are made in slave processes while the PSO control is centralized in the master process. In the master-slave PPSO, the original PSO algorithm is not modified and no improvement in solutions is obtained. The other models are described in the following subsections.

All models presented in this paper were developed for the cluster of the Nuclear Engineering Institute in language C++ with the Message Passing Interface (MPI) for a Linux distribution operatin system, Fedora.

3.1. Island PPSO: Isl-PPSO

The developed Island PPSO (Isl-PPSO) uses a ring island topology, in which, sub-populations of particles evolve separately, exchanging particles periodically through "migrations", according to the ring topology. After a given number of iterations, the best particle of each island, called local best ($\overrightarrow{lBest}$) "migrates" to another island, according to a ring topology (see Fig, 1a). More precisely, it replaces a randomly chosen particle in the other island. The ring topology has been chosen, due to the easy implementation and the good results obtained with this topology in PGA [2].



a b c d

Figure 1. Proposed Parallel models for PSO: (a) Island PSO – ring topology; (b) Cellular PSO; (c) Neighborhood-Island PPSO with 2 neighbors: N2-Isl-PPSO; (d) Neighborhood-Island PPSO with 4 neighbors: N4-Isl-PPSO.

3.2. Cellular PPSO: Cell-PPSO

In the proposed Cellular PPSO (Cell-PPSO) particles are distributed into processing cells, connected in 2D-grid topology, as illustrated in Fig. 1b, where P_k is the particle k. In the Cell-PPSO, $\overrightarrow{gBest}$ is no more visible to all particles. Instead, particles use a local best, also called $\overrightarrow{lBest}$, which is the best particle among the neighbors.

3.3. Neighborhood-Island PPSO

The Neighborhood-Island PPSO is an alternative island model, in which each island is connected to other ones by N neighbors, according to a given topology. The advantage observed in this model is that it can provide a natural interface between islands, in which the information is naturally exchanged, avoiding definition of "migration strategies" (as occurs in the Island model). In this work, two topologies were investigated: i) a ring topology, in which each island connects to other ones by 2 particles, according to a ring topology and ii) a 2D-grid topology, in which each islands are connected to 4 other ones by 4 neighbors, according to a 2D-grid topology.

The ring topology, called here N2-Isl-PPSO, is illustrated in Fig. 1c while the grid topology, called N4-Isl-PPSO, is illustrated in Fig. 1d.

4. OPTIMIZATION PROBLEMS

4.1. Nuclear core design optimization problem

Here, it is considered a simplified cylindrical three-enrichment-zone pressurized water reactor (PWR), with typical cell composed by moderator (light water), cladding and fuel. Fig. 2 illustrates the proposed core reactor. Design parameters as well as their ranges are shown in Table 1. Reactor height (h) as well as thickness of each enrichment zone (R1, R2 and R3) does not change in the optimization process.

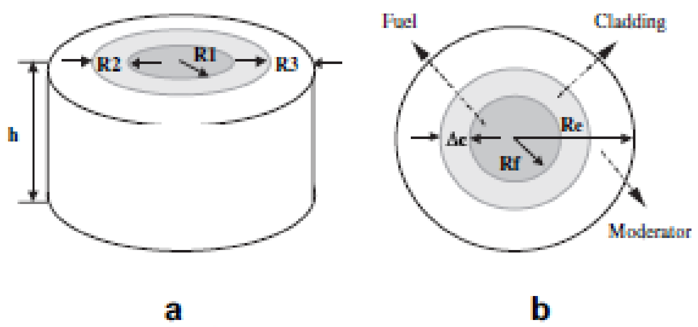


Figure 2. (a) The core reactor and (b) its typical cell.

Table 1. Optimization parameters range.

Parameter	*Symbol*	*Range*
Fuel radius (cm)	Rf	0.508–1.27
Cladding thickness (cm)	Δc	0.0254–0.254
Moderator thickness (cm)	Re	0.0254–0.762
Enrichment of zone 1 (%)	E_1	2.0–6.0
Enrichment of zone 2 (%)	E_2	2.0–6.0
Enrichment of zone 3 (%)	E_3	2.0–6.0
Fuel material	M_f	{U-metal or UO2}
Cladding material	M_c	{Zircaloy-2, Aluminum or stainless-304, Al-Li}

The objective of the optimization problem is to maximize the average thermal flux, ϕ_{AVE}, of the proposed reactor, considering as constraints: criticality, sub-moderation and maximum peak factor, fp_{MAX}. ϕ_{AVE} is the source normalized flux, calculated by Hammer reactor physics code [13] and V_m is the moderator volume. The fitness function to be maximized was developed in such a way that, if all constraints are satisfied, it assumes the value of the average thermal flux, f_{AVE}. Otherwise, it is penalized proportionally to the disagreement on the constraint.

4.1. Fuel reload optimization problem

In this work, Angra-1 nuclear reactor, a Brazilian 626 MW PWR has been considered. The reactor core comprising 121 fuel assemblies (FA), distributed according to the following configuration: i) 80 8-fold type symmetric FA, ii) 40 4-fold type symmetric FA and iii) one central element. Using an eighth-core symmetry, the number of fuel assemblies to be shuffled falls from 121 (the whole core) to 20 (central element is fixed).

The fuel reload objective used in this investigation was to maximize the cycle length by using the ‘‘low-leakage’’ strategy, without making use of burnable poison. The cycle maximization can be obtained by maximizing the end-of-cycle critical boron concentration (C_B). However, to reduce computational costs, the boron concentration at equilibrium of xenon was used. As an operational constraint, imposed by technical specifications, the radial power peaking-factor (F_{XY}) must be limited to 1.435 (upper bound).

In order to test the PPSO approaches, a simplified problem has been proposed, considering: i) no constraint about the allowed positions for the assemblies; ii) no burnable poisons and iii) no rotation in the fuel assemblies.

The reactor physics calculations have been made by the RECNOD [14] code, which runs on the same computational platform used by the PSO. Due to several limitations of the code, the constraint F_{XY} <1.435 (*FXY* is not provided by RECNOD) is substituted by P_{MAX} < 1.395, where P_{MAX} is the maximum normalized assembly power. Value 1.395 in RECNOD implies in F_{XY} <1.435 in the code used in practice (not available for the considered parallel architecture). The objective function, *f*, to be maximized is, then, given by Eq. (15).

$$f = \begin{cases} C_B, & P_{MAX} < 1{,}395 \\ C_B - k.P_{MAX}, & \text{otherwise} \end{cases} \qquad (15)$$

where C_B is the Boron concentration, F_{XY} the radial peaking-factor and *k* a penalization multiplier.

Associating FAs to core positions is a combinatorial problem, which can be mapped into a classical Traveling Salesman Problem (TSP), to which the proposed optimization techniques have been successfully applied, as can be seen in the TSP benchmark (Session 5.1).

5. METHOD APPLICATION AND RESULTS

In this session it is shown that a compilation of the computational experiments made in this work. For all approaches, typical PSO parameters, extracted from literature [15-16] and other investigations, such as inertia weight, social and cognitive coefficients, have been used.

Different number of particles, maximum velocity (velocity constraint is used) and number of iterations, have been tested and the best results obtained by each method are presented.

The parallel platform used was a cluster with 3 Intel Core2-Duo 3.0 GHz processors communicating through the Message Passing Interface (MPI) protocol [17]. For all experiments shown here, a swarm of 96 particles have been used. The distributions of sub-populations are detailed in Table 2.

Table 2. PSO topologies and population distribution.

PSO topology	Population distribution
Standard PSO	Single population (96 particles)
Master-slave PPSO	Single population (96 particles)
Cell-PPSO	Cellular: 2-D grid (96 particles)
N2-Isl-PPSO	6 islands of 16 particles
N4-Isl-PPSO	12 islands of 8 particles
Isl-PPSO-[Migration Interval]	6 islands of 16 particles

5.1. Preliminary validation

Before applying proposed PPSOs to the nuclear reactor problems, some validation tests were done. The first tests were carried out using several numerical functions. In order to compare with results found in literature [18], number of particles and number of iterations have been fixed in 20 and 200,000 respectively. Due to the reduced number of particles, N4-Isl-PPSO and Isl-PPSO were not applied.

Then, a harder benchmark has been proposed. An asymmetric 48-cities Traveling Salesman Problem (TSP) – the Rykel-48 [19]. Since it is a combinatorial problem and (traditional) PSO better deals with continuous variables, a transformation from combinatorial to continuous space is necessary. For more information, refer to Meneses et al. (in press). As a result, a complex nonlinear and multimodal 48-D space is obtained, in which each dimension ranges between 0 and 1. The two motivations for choosing a TSP problem are: i) to promote a great challenge to PSO (which is not exactly skilled to combinatorial problems) and ii) the fuel reload problem uses the same approach for encoding and manipulating solution candidates.

Different parameter combinations have been tested. For each one, 100 experiments with different random seeds have been made, resulting in 11,200 experiments (PSO runs). Both cognitive and social coefficients, c1 and c2, were set to 2.0, as recommended in literature [15-16]. In order to provide good balance between exploration (more global search, at initial iterations) and exploitation (more local search, at later iterations). inertia weight was linearly decreased from 0.8 to 0.2 during 50,000 iterations (maximum iterations).

Through the 11,200 experiments, it could be observed, that limitation of particles' velocity at low values had a positive effect to the PPSO and a negative effect to the standard (or master-slave) PSO. It can be explained by the great improvement in diversity due to the communication strategies used. Hence, many different values of maximum velocities (V_{MAX})

have been tested. Considering that V_N is the greatest distance between 2 points in the search space (in this problem, $V_N = 6.92$), the following values of velocity have been tested: $V_N/1$, $V_N/2$, $V_N/4$, $V_N/8$. The best value for the standard PSO was $V_{MAX} = V_N/2$, which is 4 times greater than the best value for the PPSOs, $V_{MAX} = V_N/8$. The best results for each model are shown in Table 3.

Table 3. Results obtained in Rykel-48 problem.

PSO	f_{ave} (average fit. val.)	STD (std.deviation)	f_{min} (min. fit. val.)	f_{max} (max. fit. val.)	f<15000
Standard PSO	17124	8.37%	14658	21798	3
Master-slave PPSO	17124	8.37%	14658	21798	3
Cell-PPSO	15477	2.21%	14765	16194	5
N2-isl-PPSO	15248	2.40%	14572	16389	29
N4-isl-PPSO	15301	2.33%	14698	16360	22
Isl-PPSO-10	15594	3.18%	14709	17258	7
Isl-PPSO-100	15459	2.79%	14806	17575	7
Isl-PPSO-1000	15398	2.15%	14739	16456	8

Note that all PPSO models were able to find good f_{min} values in 100 runs. However, by observing averages and standard deviation, the coarse- and fine-grained PPSO is quite better demonstrating great robustness. Moreover, the two Neighborhood-Island PPSOs demonstrated to be the best ones, with expressive percentage of values bellow 15,000 (considered very good results). Hence, it may be concluded that communication strategy used in Neighborhood-Island PPSO seems to have advantages over traditional island and cellular models. As the problem's objective function is not time consuming, the speedup has not been computed in this benchmark.

5.2. Application to reactor core design

Results shown in Session 5.1 were a great motivation for application of the proposed PPSO in nuclear reactor core design. This problem, however, is too much more time consuming (a single non-parallel PSO run takes about 10 h) and consequently fewer experiments were done. Based on observations made in the TSP experiments, as well as in preliminary experiments done with simpler (less time consuming) core design problems, the set of PSO parameters were defined. A swarm of 96 particles have been used. Both cognitive and social coefficients, c1 and c2, were set to 2.0. Inertia weight was linearly decreased from 0.8 to 0.2 during 10,000 iterations.

Surprisingly, the problem seemed to be easy to PSO. Even the traditional (non-parallel) PSO was able to find the (probable) best value (f=1.688). However, coarse- and fine-grained PPSO were more consistent in finding near-optimum results (>1.68), while the traditional PSO and the master-slave PPSO were trapped into local optima in three experiments. Table 4 shows the results obtained in 10 runs of each PPSO model.

Table 4. Optimization results for the reactor core design problem.

PSO	f_{ave}	STD (std.deviation)	f_{min}- f_{max}	$f > 1.68$	*T(min)*	*Speedup* *(Tstandard/T)*
Standard PSO	1.6793	0.82%	1.647-1.688	7	443	1
Master-slave PPSO	1.6793	0.82%	1.647-1.688	7	212	2.23
Cell-PPSO	1.6863	0.17%	1.678-1.688	9	225	2.10
N2-Isl-PPSO	1.6859	0.07%	1.683-1.688	10	113	4.18
N4-Isl-PPSO	1.6872	0.03%	1.686-1.688	10	110	4.30
Isl-PPSO-10	1.6871	0.02%	1.687-1.688	10	-	-
Isl-PPSO-100	1.6862	0.05%	1.685-1.688	10	-	-
Isl-PPSO-1000	1.6865	0.04%	1.685-1.688	10	108	4.38

In this problem, differences between results obtained by each approach are not so accentuated as occurred in the TSP benchmark. In fact, although the search space is very non-linear and multimodal, may be due to the small number of variables, the problem seems to be less complex. Nevertheless, such results ratify some conclusions obtained in the TSP experiments. Considering the experiments done, the following observations could be made: i) coarse- and fine-grained PPSO models demonstrated to be better than standard and master-slave PSO; ii) although standard PSO presented a very good performance, it presented premature convergence in 30% of the cases, demonstrating to be less robust; and iii) standard deviation in the PPSO with more elaborated communication strategies (Cell-PPSO, Isl-PPSO, N2-Isl-PPSO and N4-Isl-PPSO) is smaller, demonstrating more robustness.

As expected, great gains are observed in parallel models. It could be observed that master-slave and cellular models are much time consuming due to the communications overhead. Note that although 6 cores have been used, the best speedups were about 4 times (in the island and neighborhood-island models), that occurred due to the relatively fast evaluation of solution candidates (communication overhead becomes more significant) and the use of synchronous approaches (in practice, processes do not take exactly the same amount of time).

Another observation is that, although master-slave presents an overhead in the master process (which centralizes the main PSO loop), the Cell-PPSO is slightly slower. This fact may be attributed to the communication overhead in the Cell-PPSO, which was relatively significant due to the fast evaluation of solution candidates.

5.3. Application to fuel reload optimization

As the proposed reactor design problem was still easy for standard PSO a more difficult and realistic problem has been chosen. The fuel reload optimization problem increases both complexity and evaluation time of objective function.

Considering that, a single non-parallel standard PSO run takes about 30 h, only 5 experiments were done for each PSO (and PPSO). Population size, c1 and c2 constants, as well as inertia weight were the same used in the reactor core design. Inertia weight was linearly decreased from 0.8 to 0.2 during 10,000 iterations. Table 5 shows results obtained in such experiments.

Table 5. Results for the fuel reload optimization problem.

PSO	f_{ave} (average fit val.)	**STD** (std.deviation)	f_{min}- f_{max} (min. fit. val.)	***T (min)***	***Speedup (Tstandard/T)***
Standard PSO	1411	2%	1370-1432	1735	1
Master-slave PPSO	1411	2%	1370-1432	621	2.79
Cell-PPSO	1433	2%	1409-1467	588	2.95
N2-Isl-PPSO	1421	1%	1392-1447	357	4.86
N4-Isl-PPSO	1448	3%	1416-1540	323	5.37
Isl-PPSO-100	1436	1%	1409-1455	-	-
Isl-PPSO-1000	1444	4%	1402-1557	334	5.20

As expected, all coarse- and fine-grained PPSO demonstrated to be better than standard and master-slave PPSO. Here, N4-Isl-PPSO and Isl-PPSO-1000 presented, not only the best average values, but outstanding maximum values (f >1500). In this problem, differently from the core design problem, standard deviation in the PPSO with more elaborated communication strategies (Cell-PPSO, Isl-PPSO, N2-Isl-PPSO and N4-Isl-PPSO) is not smaller in all cases. However, due to the reduced number of experiments, conclusions regarding standard deviation may be prejudiced.

Again, great gains are observed in parallel models. Confirming results shown in last session, it can be observed that master-slave and cellular models are much time consuming due to the communications overhead. As well as occurred in the core design problem, it was observed in this problem, an "inversion" in expected times of N2-Isl-PPSO and N4-Isl-PPSO. Investigating the problem, it could be concluded that 12 island (processes) distributed among 6 cores (2 processes in each core) are well optimized by the operating system, probably due to the great amount of I/O done by the reactor simulator. During the I/O in some islands, processes in other ones may be running (time-sharing).

Note that, although execution times are drastically reduced in parallel models, they may be still high for real-world problems, in which simplifications used here were not applied. For example: if burnable poisons were considered, the simplification of considering boron concentration at equilibrium of xenon would not be applied and the burnup should be simulated till the end-of-cycle, taking about 4–6 times the computational costs measured in this work. The use of more elaborated reactor physics codes may also increase much more simulation times. Therefore, such timeconsuming real-world problems should better be solved in few optimization runs. Hence, by enlarging the optimization tools robustness, the chances of reaching better results in each single run are also increased.

From the economic point of view, the motivation for improving optimization tools may become clearer. Just for illustration, an approximate calculation of the gains for Angra-1 NPP can be made as follows. By extrapolating the critical boron concentration to 0 ppm, it is possible to calculate the cycle length in terms of Effective Full Power Day (EFPD). For the best value in Table 5 (C_B =1557 ppm), the calculated cycle length is approximately 389 EFPD, while for the worst value (C_B = 1370 ppm) it is 343 EFPD. Considering that 1 EFPD generates about US$500,000 to Angra-1 NPP owner, the difference between the best and the worst (which is already very good) optimization result represents an amount of about S$23,000,000 in the NPP operation.

6. CONCLUSIONS

In this work four different PPSO approaches have been developed and applied to classical nuclear engineering problems. As expected, outstanding gains in terms of speedup of the optimization processes, due to parallelism occur in all models. Due to the reduced amount of communication needed, island and Neighborhood-Island models demonstrated to be faster.

In the Rykel-48 benchmark both Neighborhood-Island models (N2-Isl-PPSO and N4-Isl-PPSO) were clearly much better than all other ones. The better performance should be due to the improved ability diversity maintenance during the search process, providing a great consistency in finding near-optimum regions. The third well classified in this problem was the island PPSO with migration interval =1000 (Isl-PPSO-1000).

After benchmark investigation, the PPSO was applied to two classical nuclear engineering problems. The first one was the core design optimization, in which N4-Isl-PPSO demonstrated to be slightly better, and very close to the Isl-PPSO-10. Surprisingly, this problem, seemed to be not so difficult, even for the non-parallel PSO, which performed worse, but differences are not accentuated.

Aiming to test the proposed approaches in a more realistic, complex and time-consuming problem, the fuel reload optimization of Angra-1 NPP was considered. In this problem, N4-Isl-PPSO and Isl-PPSO-1000 obtained the best results. The economic gains obtained in fuel reload optimization problem justify the continuous improvement of optimization tools and outline the relevance of using PPSO models.

Among proposed models, the best performance in all problems considered here were the Neighborhood-Island and the island (with periodic migration) models. Although, island models are less time consuming (due to reduced amount of communication between processors), the performance is affected by the ‘‘migration interval’’ parameter. The relation between such parameter and the performance of the Isl-PPSO seems, however, to be non-trivial (i.e., in Rykel-48 problem, Isl-PPSO with low ‘‘migration intervals’’ performed better, contrasting with the nuclear applications, in which higher ‘‘migration intervals’’ better results). In summary, here, the Neighborhood-Island (specially the N4-Isl-PPSO) and island (with periodic migrations) models demonstrated to be better than the cellular model and much better than the master-slave. The first one has the advantage of eliminating the ‘‘migration interval’’ parameter, while the last one is a little less time consuming.

This work is, may be, one of the first investigations, in which enhanced PPSO models (with different communication/neighborhood strategies) are applied to real-world engineering problems. Nevertheless, investigating other PPSO approaches with different communication strategies, as well as application to other real-world problems, may contribute for refining evidences pointed in this work.

ACKNOWLEDGMENTS

Cláudio M.N.A. Pereira and Roberto Schirru are supported by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) and FAPERJ (Fundação Carlos Chagas Filho de Amparo à Pesquisa no Estado do Rio de Janeiro).

REFERENCES

1. J. Kennedy, R. C. Eberhart, ”Particle Swarm Optimization”, *Proceedings of IEEE International Conference on Neural Networks*, Vol. 4, pp.1942–1948 (1995).
2. C. M. N. A. Pereira, C. M. F. Lapa,”Coarse-grained Parallel Genetic Algorithm Applied To A Nuclear Reactor Core Design Optimization Problem”, *Annals of Nuclear Energy*, 30 (5), pp. 555–565 (2003).
3. R. P. Domingos, R. Schirru, C. M. N. A. Pereira, “Particle Swarm Optimization In Reactor Core Designs”, *Nuclear Science and Engineering*, 152, pp. 1–7 (2006).
4. C. M. N. A. Pereira, R. Schirru, R., C. M. F. Lapa, J. A. C. Canedo, M. Waintraub, A. A. M. Meneses, R. P. Baptista, N. N. Siqueira, ”Particle Swarm Optimization Applied To Nuclear Engineering Problems”, *International Journal of Nuclear Knowledge Management*, 2 (3), pp. 313–332 (2007).
5. A. M. M. Meneses, M. D. Machado, R. Schirru, “Particle Swarm Optimization Applied to Q4 the Nuclear Reload Problem of a Pressurized Water Reactor”, *Progress in Nuclear Energy*, (in press).
6. J. A. C. C. Medeiros, R. Schirru, ”Identification Of Nuclear Power Plant Transients Using The Particle Swarm Optimization Algorithm”, *Annals of Nuclear Energy*, 35 (4), pp. 576–582 (2008).
7. J. F. Chang, S. C. Chu, J. F. Roddick, J. S. Pan, “A Parallel Particle Swarm Optimization Algorithm with Communication Strategies”, *Journal of Information Science and Engineering*, 21, pp. 809–818 (2005).
8. J. F. Schutte, J. A. Reinbolt, B. J. Fregly, R. T. Haftka and A. D. GEORGE, “Parallel Global Optimization with The Particle Swarm Algorithm”, *International Journal For Numerical Methods In Engineerin* 61:2296–2315 (2004)
9. N. Jin, Y. R. Samii, “Parallel Particle Swarm Optimization and Finite-Difference Time-Domain (PSO/FDTD) Algorithm for Multiband and Wide-Band Patch Antenna Designs”, *IEEE Transactions on Antenna and Propagation*, 53, 11, pp.3459-3468 (2005).
10. G. Venter, and J. S. Sobieski, “A Parallel Particle Swarm Optimization Algorithm Accelerated by Asynchronous Evaluations”, *Journal of Aerospace Computing, Information, and Communication*, 3(3):123-137 (2006).
11. W. Jian, Y. C. Xue, J. X. Qian, ”An Improved Particle Swarm Optimization Algorithm With Neighborhood Topologies”, *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics*, pp. 2332–2337 (2004).
12. E. Cantú-Paz, *Efficient and Accurate Parallel Genetic Algorithms*, Kluwer Academic Publishers (2000).
13. J. E. Suich, H. C. Honec, *The HAMMER System Heterogeneous Analysis by Multigroup Methods of Exponentials and Reactors*, Savannah River Laboratory, Aiken, South Carolina (1967).
14. J. L. C. Chapot, “Otimização Automática de Recargas de Reatores de Água Pressurizada utilizando Algoritmos Genéticos”, D.Sc. thesis, COPPE/UFRJ, Brazil (in Portuguese) (2000).
15. Y. Shi, R. C. Eberhart, “Parameter Selection in Particle Swarm Optimization”, *Evolutionary Programming VII: Proceedings of the Seventh Annual Conference on Evolutionary Programming, New York*, pp. 591–600 (1998).
16. C. Trelea, “The Particle Swarm Optimization Algorithm: Convergence Analysis And Parameter Selection”, *Information Processing Letters*, 85 (6), pp. 317–325 (2003).
17. W. Gropp, E. Lusk, A. Skjellum, *Using MPI: Portable Parallel Programming with the Message Passing Interface*, MIT Press (1994).
18. E. S. Peer, F. V. D. Bergh, A. P. Engelbrecht, ”Using Neighbourhoods With The Guaranteed Convergence PSO”, *Proceedings of IEE Swarm Intelligence Symposium*, pp. 235–242 (2003).
19. TSPLIB, http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95 (2008).