Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications (M&C + SNA 2007)
Monterey, California, April 15-19, 2007, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2007)

A POPULATIONAL PARTICLE COLLISION ALGORITHM APPLIED TO A NUCLEAR REACTOR CORE DESIGN OPTIMIZATION

Wagner F. Sacco* and Hermes Alves Filho
Departamento de Modelagem Computacional
Instituto Politécnico – Universidade do Estado do Rio de Janeiro
Caixa Postal 972825 – CEP 28601-Nova Friburgo, RJ, Brazil
wfsacco@iprj.uerj.br; halves@iprj.uerj.br

Cassiano R.E. de Oliveira
Nuclear and Radiological Engineering Program
Georgia Institute of Technology
900 Atlantic Drive, NW, Atlanta, GA, 30332-0405
cassiano.deoliveira@nre.gatech.edu

ABSTRACT

The Particle Collision Algorithm (PCA) is a recently introduced metaheuristic conceptually similar to Simulated Annealing, without, though, the necessity of estimating the free parameters as in the latter algorithm. It is loosely inspired by the physics of nuclear particle collision reactions, particularly scattering and absorption. A "particle" that reaches a promising area of the search space is "absorbed", while one that hits a low-fitness region is "scattered". PCA is also a Metropolis algorithm, as a solution worse than the currently best may be accepted with a certain probability. In this article, we introduce a populational version of the Particle Collision Algorithm, the Populational PCA (PopPCA), which is a hybridization of the PCA and the genetic algorithm (GA). At the end of a generation, the particles reproduce and the fittest individuals survive. We apply the new algorithm to an optimization problem that consists in adjusting several reactor cell parameters, such as dimensions, enrichment and materials, in order to minimize the average peak-factor in a three-enrichment-zone reactor, considering restrictions on the average thermal flux, criticality and sub-moderation. The populational PCA is compared to other metaheuristics previously applied to the problem and shows to perform better than them, thus demonstrating its potential for other applications and further development.

Key Words: Metaheuristics, Stochastic Optimization, Nuclear Reactor Design

1. INTRODUCTION

The Particle Collision Algorithm (PCA) [1, 2] is loosely inspired by the physics of nuclear particle collision reactions [3], particularly scattering (where an incident particle is scattered by a target nucleus) and absorption (where the incident particle is absorbed by the target nucleus). Thus, a particle that hits a high-fitness "nucleus" would be "absorbed" and would explore the boundaries. On the other hand, a particle that hits a low-fitness region would be scattered to another region. This permits us to simulate the exploration of the search space and the exploitation of the most promising areas of the fitness landscape through successive scattering

* Corresponding author.

and absorption collision events. The PCA resembles in its structure that of simulated annealing (SA) [4]: first an initial configuration is chosen; then there is a modification of the old configuration into a new one. The qualities of the two configurations are compared. A decision then is made on whether the new configuration is "acceptable". If it is, it serves as the old configuration for the next step. If it is not acceptable, the algorithm proceeds with a new change of the old configuration. PCA can also be considered a Metropolis algorithm [5], as a trial solution can be accepted with a certain probability. This acceptance may avoid the convergence to local optima.

In this article, we introduce a populational version of the Particle Collision Algorithm, the Populational PCA (PopPCA). The motivation is to promote a greater exploration of the search space and eventually to parallelize the algorithm. This new optimization algorithm is a hybridization of the PCA and the genetic algorithm (GA) [6]. Exactly like in the standard PCA, each particle moves in the search space through successive collision events. By the end of a generation, though, the particles interact for reproduction. The fittest individuals survive and constitute the population for the next generation.

The Populational PCA is applied to a nuclear reactor design optimization problem that was previously attacked using the genetic algorithm [7, 8], and Simulated Annealing and the standard PCA [2], among others. The four algorithms are submitted to the same computational effort and their results are compared.

2. THE POPULATIONAL PARTICLE COLLISION ALGORITHM

The pseudo code description of the Populational PCA is shown in Figure 1. In relation to the pseudo code presented in ref. [2], the difference is the introduction of a loop to iterate over the population and of function "New_Pop", which generates the new population for the next generation in a fashion similar to the genetic algorithm. Please note that the algorithm's default is for maximization problems. For minimization, just multiply the objective function by -1 and invert the ratio in $p_{scattering}$.

First, an initial population is randomly generated. Following that, there is a loop counting the number of iterations until a stopping criterion is met (in this article, we used a maximum number of fitness function evaluations). Inside this loop, there is another loop to perform the same tasks for all individuals in a generation. It starts with the stochastic perturbation of each individual or solution.

After stochastic perturbation, if the fitness of the new configuration is better than the fitness of the old configuration, then the "particle" is "absorbed", there is an exploration of the boundaries searching for an even better solution. Function "Exploration" performs this local search, generating a small stochastic perturbation of the solution inside a loop. In PCA's current version, it is a one-hundred-iteration loop. The "small stochastic perturbation" is similar to the previous stochastic perturbation, but each variable's new value is kept within the boundaries of the original value.

Otherwise, if the fitness of the new configuration is worse than the old configuration's, the "particle" is "scattered". The scattering probability ($p_{scattering}$) is inversely proportional to its fitness quality. The configuration is "scattered" (replaced by a random configuration) or, following the Metropolis paradigm, survives, with its boundaries explored ("else" branch of the function).

At the end of a generation, function "New_Pop" is called, randomly pairing and crossing the individuals using "one-point crossover" [9], where a cutting point is determined at random and the segments before this point are swapped. After that, the parents and siblings are ranked in descending fitness order and the first *Pop_Size* individuals from this ordered array survive to the next generation.

```
For i = 1 to Pop_Size
        Generate an initial solution Old_Config[i]
End For
For n = 0 to # of iterations
        For i = 1 to Pop_Size
                Generate a stochastic perturbation of the solution
                If Fitness(New_Config[i]) > Fitness(Old_Config[i])
                        Old_Config[i]:= New_Config[i]
                        Exploration ( )
                Else
                        Scattering ( )
                End If
        End For
        New_Pop( )
End For

Exploration ( )
        For n = 0 to # of iterations
                Generate a small stochastic perturbation of the solution
                If Fitness(New_Config[i]) > Fitness(Old_Config[i])
                        Old_Config := New_Config
                End If
        End For
return

Scattering ( )
        p_scattering = 1 - Fitness(New_Config[i]) / Best Fitness
        If p_scattering > random (0, 1)
                Old_Config[i] := random solution
        Else
                Exploration ();
        End if
return

New_Pop ( )
        Cross two random individuals from population
        Rank parents and siblings in descending fitness order
        Pick up the first Pop_Size individuals from the ranked array
return
```

Figure 1. PopPCA's pseudo code.

3. THE OPTIMIZATION PROBLEM

The new method was tested for a simple nuclear engineering optimization problem. Briefly stated, this problem consists in designing a reactor core which minimizes the average power peak factor without violating certain operational constraints such as the average fluence of thermal neutrons, reactor criticality and neutron sub-moderation. A cylindrical, 3-zone reactor is considered, with a typical cell composed by the fuel, the fuel cladding and the moderator. The design parameters are the fuel radius, cladding thickness, moderator thickness, each zone's enrichment, the fuel material and the cladding material.

The design parameters that may be changed in the optimization process, as well as their variation ranges are shown in Table I.

Table I. Parameters range

Parameter	Symbol	Range
Fuel Radius (cm)	R_f	0.508 to 1.270
Cladding Thickness (cm)	Δ_c	0.025 to 0.254
Moderator Thickness (cm)	R_e	0.025 to 0.762
Enrichment of Zone 1 (%)	E_1	2.0 to 5.0
Enrichment of Zone 2 (%)	E_2	2.0 to 5.0
Enrichment of Zone 3 (%)	E_3	2.0 to 5.0
Fuel Material	M_f	{U-Metal or UO_2}
Cladding Material	M_c	{Zircaloy-2, Aluminum or Stainless-304}

The objective of the optimization problem is to minimize the average peak-factor, f_p, of the proposed reactor, considering that the reactor must be critical (k_{eff}= 1.0 ± 1%) and sub-moderated, providing a given average flux ϕ_0. Then, the optimization problem can be written as follows:

Minimize $f_p\,(R_f,\ \Delta_c,\ R_e,\ E_1,\ E_2,\ E_3,\ M_f,\ M_c)$

Subject to:

$$\phi(R_f,\ \Delta_c,\ R_e,\ E_1,\ E_2,\ E_3,\ M_f,\ M_c) = \phi_0; \quad (1)$$

$$0.99 \le k_{eff}(R_f,\ \Delta_c,\ R_e,\ E_1,\ E_2,\ E_3,\ M_f,\ M_c) \le 1.01; \quad (2)$$

$$\frac{dk_{eff}}{dV_m} > 0\,; \quad (3)$$

$$R_{f\,min} \le R_f \le R_{f\,max}; \quad (4)$$

$$\Delta_{c\,min} \le \Delta_c \le \Delta_{c\,max}; \quad (5)$$

$$R_{e\,min} \le R_e \le R_{e\,max}; \quad (6)$$

$$E_{1\,min} \le E_1 \le E_{1\,max}; \quad (7)$$

$$E_{2\,min} \le E_2 \le E_{2\,max}; \quad (8)$$

$$E_{3\,min} \le E_3 \le E_{3\,max}; \quad (8)$$

$$M_f = \{UO_2 \text{ or U-metal}\}; \quad (9)$$

$$M_c = \{\text{Zircaloy-2, Aluminum or Stainless-304}\}, \quad (10)$$

where V_m is the moderator volume and the *min* and *max* subscripts refer to the lower and upper limits of the parameters ranges, given in Table I.

For further details, please refer to [7].

4. IMPLEMENTATION AND RESULTS

The optimization algorithm is linked to the Reactor Physics code HAMMER [10] for the cell and diffusion equations calculations. The PopPCA sends to the code by an interface a solution (the reactor's dimensions and materials) and receives the fitness. The fitness function was developed in such a way that, if all constraints are satisfied, it has the value of the average peak factor. Otherwise, it is penalized proportionally to the discrepancy on the constraint, as shown by equation (11).

$$f = \begin{cases} f_p, & \Delta k_{eff} \le 0.01; \Delta\phi \le 0.01\phi_0; \dfrac{\Delta' k_{eff}}{\Delta Vm} > 0 \\ f_p + r_1.\Delta k_{eff}, & \Delta k_{eff} > 0.01; \Delta\phi \le 0.01\phi_0; \dfrac{\Delta' k_{eff}}{\Delta Vm} > 0 \\ f_p + r_2.\Delta\phi, & \Delta k_{eff} \le 0.01; \Delta\phi > 0.01\phi_0; \dfrac{\Delta' k_{eff}}{\Delta Vm} > 0 \\ f_p + r_3.\dfrac{\Delta' k_{eff}}{\Delta Vm}, & \Delta k_{eff} \le 0.01; \Delta\phi \le 0.01\phi_0; \dfrac{\Delta' k_{eff}}{\Delta Vm} < 0 \\ f_p + r_1.\Delta k_{eff} + r_2.\Delta\phi, & \Delta k_{eff} > 0.01; \Delta\phi > 0.01\phi_0; \dfrac{\Delta' k_{eff}}{\Delta Vm} > 0 \\ f_p + r_1.\Delta k_{eff} + r_3.\dfrac{\Delta' k_{eff}}{\Delta Vm}, & \Delta k_{eff} > 0.01; \Delta\phi \le 0.01\phi_0; \dfrac{\Delta' k_{eff}}{\Delta Vm} < 0 \\ f_p + r_2.\Delta\phi + r_3.\dfrac{\Delta' k_{eff}}{\Delta Vm}, & \Delta k_{eff} \le 0.01; \Delta\phi > 0.01\phi_0; \dfrac{\Delta' k_{eff}}{\Delta Vm} < 0 \\ f_p + r_1.\Delta k_{eff} + r_2.\Delta\phi + r_3.\dfrac{\Delta' k_{eff}}{\Delta Vm}, & \Delta k_{eff} > 0.01; \Delta\phi > 0.01\phi_0; \dfrac{\Delta' k_{eff}}{\Delta Vm} < 0 \end{cases} \tag{11}$$

Table II shows the values of radial power peaking factor obtained by the Populational PCA in comparison with those obtained by the GA, by SA, and by the PCA. All algorithms ran for 100,000 fitness function evaluations. In the case of PopPCA, we tested two population sizes, 10 and 20 individuals. Note that the new algorithm obtained the lowest fitness value and showed its robustness finding the lowest averages and standard deviations for both population sizes.

Table II. Comparison with the GA, SA, and the PCA for 100,000 fitness evaluations.

Experiment	GA[2]	SA[2]	PCA[2]	PopPCA	
				10 ind.	20 ind.
#1	1.3195	1.3449	1.2827	1.2828	1.2813
#2	1.3116	1.3390	1.2876	1.2906	1.2867
#3	1.3300	1.3480	1.2964	1.2825	1.2833
#4	1.3294	1.3530	1.2874	1.2883	1.2860
#5	1.3595	1.3553	1.2829	1.2878	1.2790
#6	1.3562	1.3221	1.2791	1.2981	1.2855
#7	1.3372	1.3023	1.2975	1.2842	1.2887
#8	1.3523	1.3387	1.2865	1.2826	1.2874
#9	1.3614	1.3138	1.2908	1.2882	1.2798
#10	1.3467	1.3565	1.2845	1.2831	1.2864
Average	1.3402	1.3374	1.2875	1.2868	1.2844
Std. Dev.	0.0175	0.0186	0.0059	0.0049	0.0034

Table III displays the best configurations obtained by the SGA (in ref. [7]) with 300 individuals until convergence, by the PCA and SA (both in ref. [2]), and by the Populational PCA for 100,000 function evaluations, when applied exactly to the same problem. The first rows show the objectives and constraints that were met and the following ones display the design parameter values obtained by each method.

Table III. Comparison with previously published best results.

		GA[7]	SA[2]	PCA[2]	PopPCA
Objectives and Constraints	**Fitness**	1.310	1.302	1.279	1.279
	Minimum average peak factor	1.310	1.302	1.279	1.279
	Average Flux	8.02×10^{-5}	8.00×10^{-5}	8.06×10^{-5}	8.07×10^{-5}
	k_{eff}	1.000	0.998	0.991	1.007
Parameters	R_f **(cm)**	0.5621	0.5080	0.5497	0.6788
	Δr **(cm)**	0.1770	0.0558	0.1450	0.1284
	Δm **(cm)**	0.6581	0.5475	0.6111	0.6984
	E_1 **(%)**	2.756	2.2072	2.7953	2.4695
	E_2 **(%)**	4.032	2.5069	2.9469	2.5976
	E_3 **(%)**	4.457	3.8932	5.0000	4.4715
	M_f	U-metal	U-metal	U-metal	U-metal
	M_c	Stainless-304	Stainless-304	Stainless-304	Stainless-304

5. CONCLUSIONS

We introduced a populational variety of the Particle Collision Algorithm, and its comparative performance with the GA, SA, and the PCA shows that this algorithm is promising and should be further developed. We intend to implement more refined crossover mechanisms [9] and niching methods [11, 8]. We believe that these modifications will promote a greater exploration of the search space.

ACKNOWLEDGMENT

Wagner F. Sacco is supported by FAPERJ (Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro) under postdoctoral grant E-26/152.661/2005 (Fixação de Pesquisador, Nível 3).

REFERENCES

1. W. F. Sacco, C. R. E. de Oliveira, "A New Stochastic Optimization Algorithm based on Particle Collisions", *Transactions of the American Nuclear Society*, San Diego, CA, June 5-9, Vol. 92, pp.657-659 (2005).
2. W. F. Sacco, C. R. E. de Oliveira, C. M. N. A. Pereira, "Two stochastic optimization algorithms applied to a nuclear reactor core design," *Progress in Nuclear Energy*, **48**, pp.525-539 (2006).
3. J. J. Duderstadt, L. J. Hamilton, *Nuclear Reactor Analysis*, John Wiley and Sons, New York, USA (1976).
4. S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, "Optimization by Simulated Annealing", *Science*, **220**, pp.671-680 (1983).
5. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, "Equations of State Calculations by Fast Computing Machines", *Journal of Chemical Physics*, **21**, pp.1087-1091 (1953).
6. J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, USA (1975).
7. C. M. N. A. Pereira, R. Schirru, A. S. Martinez, "Basic Investigations Related to Genetic Algorithms in Core Designs," *Annals of Nuclear Energy*, **26**, pp.173-193 (1999).
8. W. F. Sacco, M. D. Machado, C. M. N. A. Pereira, R. Schirru, "The fuzzy clearing approach for a niching genetic algorithm applied to a nuclear reactor core design optimization problem," *Annals of Nuclear Energy*, **31**, pp.55-69 (2004).
9. D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, USA (1989).
10. J. E. Suich, H. C. Honeck, *The HAMMER System Heterogeneous Analysis by Multigroup Methods*, Savannah River Laboratory, Aiken, USA (1967).
11. S. W. Mahfoud, *Niching Methods for Genetic Algorithms*, PhD Thesis, Illinois Genetic Algorithm Laboratory, University of Illinois at Urbana-Champaign, Urbana, USA (1995).