2017 International Nuclear Atlantic Conference - INAC 2017
Belo Horizonte, MG, Brazil, October 22-27, 2017
ASSOCIAÇÃO BRASILEIRA DE ENERGIA NUCLEAR – ABEN

Agência Brasileira do ISBN
ISBN 978-85-99141-07-6
9 788599 141076

OPTIMIZATION OF RELOAD OF NUCLEAR POWER PLANTS USING ACO TOGETHER WITH THE GENES REACTOR PHYSICS CODE

Alan M. M. de Lima[1], Fernando S. Freire[2], Andressa S. Nicolau[1] and Roberto Schirru[1]

[1] Programa de Engenharia Nuclear - COPPE
Universidade Federal do Rio de Janeiro
Avenida Horácio Macedo, 2030
Bloco G – Sala 206
Cidade Universitária – Ilha do Fundão
21941-972 Rio de Janeiro, RJ
alan@lmp.ufrj.br, andressa@lmp.ufrj.br, schirru@lmp.ufrj.br

[2] Eletrobrás Termonuclear S.A. - ELETRONUCLEAR
Rua da Candelária, 65 – 7º andar
20091-906 Rio de Janeiro, RJ
ffreire@eletronuclear.gov.br

ABSTRACT

The Nuclear reload of a Pressurized Water Reactor (PWR) occurs whenever the burning of the fuel elements can no longer maintain the criticality of the reactor, that is, it cannot maintain the Nuclear power plant operates within its nominal power. Nuclear reactor reload optimization problem consists of finding a loading pattern of fuel assemblies in the reactor core in order to minimize the cost/benefit ratio, trying to obtain maximum power generation with a minimum of cost, since in all reloads an average of one third of the new fuel elements are purchased. This loading pattern must also satisfy constraints of symmetry and security. In practice, it consists of the placing 121 fuel elements in 121 core positions, in the case of the Angra 1 Brazilian Nuclear Power Plant (NPP), making this new arrangement provide the best cost/benefit ratio. It is an extremely complex problem, since it has around 1% of great places. A core of 121 fuel elements has approximately 10^{13} combinations and 10^{11} great locations. With this number of possible combinations it is impossible to test all, in order to choose the best. In this work a system called ACO-GENES is proposed in order to optimization the Nuclear Reactor Reload Problem. ACO is successfully used in combination problems, and it is expected that ACO-GENES will show a robust optimization system, since in addition to optimizing ACO, it allows important prior knowledge such as K infinite, burn, etc. After optimization by ACO-GENES, the best results will be validated by a licensed reactor physics code and will be compared with the actual results of the cycle.

1. INTRODUCTION

The reactor reload process of the Nuclear Pressurized Water Reactor (PWR) occurs whenever it is no longer possible to maintain the critical reactor producing energy at the nominal power, since the burning of the Fuel Elements (FEs) in the core of the reactor reached a certain value, and the reactor needs to be turned off to exchange the FEs.

FEs are discharged from the core and stored in a used fuel pool. The FEs with very low concentrations of U235 are permanently maintained in this pool because they have normally reached their firing limit, while the FEs with higher concentrations of U235 remaining from the previous cycle together with the new FEs will form part of the arrangement of the next cycle .

The nuclear reactor reload optimization problem (NRROP) consists in determining the positioning of new and used FEs in the core of the reactor in an optimized way, that is, to minimize the cost/benefit ratio of the FEs, making the most of the burning of the FE and satisfying constraints of symmetry and safety.

NRROP is a problem that its difficulty grows exponentially with the number of FEs in the reactor core. For example, the reactor core of the Angra 1 nuclear power plant contains 121 FEs, and the number of possible combinations is approximately 121! (121 factorial), which provides us with 10^{200} possible core arrangements. However, in the core there are symmetries of 1/4 and 1/8 and rules of placement of the FEs that belong to the lines of quartet and octet lines, then considering a symmetry of 1/8 and rules of Quartets and octets, this number drops to approximately 10^{13}, which is still extremely high to try to solve the problem by enumeration. The execution time of a licensed Reactor Physics code to provide us with a solution is approximately 150 seconds, with that time multiplied by 10^{13}, we are approximately 60 million years old, in an attempt to list all possible solutions. In addition to these difficulties, this problem has nonlinear characteristics, discontinuities, local maximum and minimum multiples in its solutions space (GALPERIN, 1995).

From the beginning, nuclear power plants reload is done by expert that aims to doing a combination of FEs that meet the company's expectations. The problem is that a human expert is able to test a few combinations of FEs, since even with experience and knowing that many of the possible combinations become invalid because they violate safety parameters, it is very unlikely that few tests will be able to find the optimal combination of FEs, to provide the best core. The combination of core FEs for future reloads is done well in antecedence (more or less with 2 reloads in antecedence), and has a long time to find a good core.

However, some problems may appear during this time, which can interfere in the reactor reload optimization, forcing the expert to generate a new core combination in a short time. For example, in the case of removing some FE from the core that would return to the next cycle and the same one to be damaged. In this case the expert should generate a new core combination in a short time.

In this work a computational modeling is presented to perform the NRR based on Ant Colony Algorithms (ACO), generating a system called ACO-GENES and its results are compared with the actual cycle 8 reload of the Angra 1 NPP.

The proposed model uses Ant Colony System (SCF) without the use of Heuristics (Heuristics - pre knowledge of some characteristics of the solutions).

The ACO is a system based on artificial ants (RUSSEL and NORVIG, 1995) and makes use of the technique of Reinforcement of Learning (KAELBLING et al, 1996). This algorithm was initially developed to solve the Traveling Salesman Problem (TSP), which is a combinatorial problem conceptually similar to that of NRRP.

The Code of Reactor Physics Genes is a code that has been written by one of the authors of this paper for more than 7 years, this code is able to calculate the value of the power mean in the FE with accuracy and the Boron concentration with an approximate error of plus or minus

20 ppm. With these two values it is possible to calculate how many Effective Full Power Days (EFPP) the cycle will last and thereby calculate the value of the energy to be sold.

With the use of intelligent algorithms the space of solutions is better explored, because at least approximately 30 thousand evaluations per round of the algorithm are executed. So to use Reactor Physics code we would spend a time of approximately 52 days for just one execution of the intelligent algorithm. With GENES, which runs in 5 seconds, we get an execution in 1.7 days, 30 times shorter than 52 days. While the expert runs approximately anything between 10 and 20 evaluations. The licensed Reactor Physics code only runs at one machine, but GENES can be parallelized and run with Compute Unified Device Architecture (CUDA) technology, which can shorten its execution time to something in milliseconds, thus providing us with a core configuration in a few hours. In addition, when the expert finds a good loading patterns the search process is ended, but with the use of intelligent algorithms, many valid core combinations that attend the company are found and stored in the results database and can be consulted at any time.

2. ALGORITHMS BASED ON ANT COLONIES

At the beginning of the nineties, a new algorithm for combinatorial optimization was developed. This algorithm was inspired by the observation of ant colonies and was therefore called Ant System. The Ant System has been applied successfully to complex combinatorial problems such as the Traveling Salesman Problem (DORIGO, GAMBARDELLA, 1997) and the Quadratic Allocation Problem (QAP) (GAMBARDELLA et al, 1999), and has also been applied to several problems of telecommunication chain optimization, vehicle routing, task distribution and especially combinatorial optimization problems (BOECHEL, 2003).

The curious fact in nature was the discovery that even ants being such simple and irrational beings possess natural optimization mechanisms. That is, they are able to find a shorter path between the anthill and a food source, without using visual cues, even if changes occur in the original environment such as the introduction of an obstacle.

The ants deposit during their walks a certain amount of a substance called pheromone, approximately with constant rate, and they prefer to follow a trajectory probabilistically richer in such substance.

The explanation is quite simple: shorter paths receive more pheromone than longer ones and as the amount of pheromone is proportional to the attraction of the ants, sites with more pheromones will attract more ants, so there will be more pheromones. Consequently, shorter paths are quickly found.

The natural behavior of the ants inspired the construction of an algorithm in which a set of artificial ants cooperates to solve a problem through the exchange of information via pheromone deposited on the trails.

There are several algorithms based on Ant Colony (STUTZLE, DORIGO, 2000), but in this work was used the Ant Colony System (ACO) (GAMBARDELLA, DORIGO, 1996).

The ACO was developed for application in combinatorial optimization problems of the NP-Complete type, as is the case of the Traveling Salesman Problem (TSP).TSP is one of the best known in the area of optimization. It is a traveling salesman who must visit a set of cities, starting from an initial city, going through all the others, once, and returning to the city of origin covering as little distance as possible.

When the ACO is applied to the TSP (GAMBARDELLA, DORIGO, 1996), the ant k moves from city to city, until returning to the city of origin, building a solution. To this ant k is associated a list of cities to visit Jk (r), where r is the current city of ant k.

The ant k chooses a city to move through a state transition rule described by Equation (1) (GAMBARDELLA, DORIGO, 1996) as follows:

$$s = \begin{cases} \max \{[FE(r,s)]^{\delta} \, x [HE(r,s)]^{\beta}\} & if \quad q \le q_0 \\ Roulette & if \quad q > q_0 \end{cases} \qquad (1)$$

where:
FE (r, s) is a real and positive value of the pheromone associated with the arc (r, s).
HE (r, s) is the value of the heuristic function relative to the movement of city r to city s.
The parameters δ and β weigh the relative importance between the learning of the ants, with the FE values, and the heuristic values.
q is a randomly chosen value with uniform probability within the interval [0,1] and q0 ($0 \le q0 \le 1$) is an input parameter of the algorithm.
Roulette is a random variable selected according to a given probability distribution.

The rule, represented by Equation (1), defines the move decision policy for the next states, which is a local function and takes into account two basic information. The first one, which is FE values, is the way ants change their search space to benefit the discovery of the best paths. This information is the artificial pheromone that is associated with the technique of reinforcing learning. The second information used in the choice for the next move is the information regarding the specific problem being optimized, that is, the heuristic.

Making use of a representative heuristic of the problem in question is extremely important, because the initial step of the algorithm will be given from this information and not randomly as it happens in several optimization methods, such as Genetic Algorithms.

The FE values are modified through cooperation between the ants to favor the discovery of good solutions. The two rules below are responsible for updating these values.

Local update rule

$$FE(r, s) = (1 - \rho) * FE(r, s) + \rho * FEzero \qquad (2)$$

where:

ρ is the pheromone evaporation parameter
FEzero is the initial value of the amount of pheromone

The local update rule, given by Equation (2), is used after the ants have applied the state transition rule and chosen the next city to be visited. In this way, this update is applied while the solution is being built.

The purpose of the local update rule is to stimulate the search for new regions of the search space and avoid premature convergence.

The amount of pheromone in the arcs is slowed down to allow the artificial ants to enlarge their search spaces. This process is called pheromone evaporation.

The second type of update is the global update rule. It is applied after all the ants have built a complete path and this path has been evaluated by an objective function. This rule is considered to reinforce learning of the algorithm.

The global reinforcement is given by Equation (3):

$$FE(r, s) = (1 - \alpha) * FE(r, s) + \alpha * (W / \text{best result}) \quad (3)$$

where:

α is a pheromone evaporation parameter

W is a parameter, defined by the user, which together with the parameter α express the learning speed of the algorithm.

The Pseudo-Random-Proportional Distribution, Roulette in Equation (1), is randomly selected according to Equation (4) (GAMBARDELLA, DORIGO, 1996) below:

$$\text{Roleta} = \begin{cases} \dfrac{[FE(r,s)]^{\delta} \, x [HE(r,s)]^{\beta}}{\sum_{z \in J_k(r)} [FE(r,z)]^{\delta} \, x [HE(r,z)]^{\beta}} & if \quad s \in J_k(r) \\ 0 & if \quad s \notin J_k(r) \end{cases} \quad (4)$$

where:

FE(r,s) Are the FE values that represent the trace of artificial pheromone.
HE(r,s) Is the value of the heuristic function associated with the arc (r, s).

The parameters δ and β weigh the relative importance of learning, with FE values, and heuristic values.

Jk(r) is a list that contains cities not yet visited by the ant k, when it is located in city r.

This distribution expresses the probability that an ant, being in city r, choose the city s as its next move. It is a random roulette similar to roulette used in Genetic Algorithms (Holland, 1975) to select individuals for the next generation.

3. THE NUCLEAR RELOAD PROBLEM and ACO-GENES

The NRRP is a combinatorial problems related to the nuclear reactor fuel reloading operation, in which part of the nuclear fuel is substituted, since it is necessary that the NPP operates within its nominal power. The nuclear fuel is disposed into fuel assemblies (FAs) that are distributed into the reactor core. The goal of the NRRP is to determine the permutation of FAs which optimize the uranium utilization in terms of the nuclear fuel cycle length, considering planned power demand, subjected to thermo-hydraulics constraints related to safety such as maximum allowed fuel assembly burn-up, hot channel factor and moderator coefficient temperature.

The NRROP consists in positioning the fuel elements in the core of the reactor in order to optimize the objective function.

In this problem, each ant chooses a fuel element to occupy a certain position in the core of the reactor. These choices are made until there are no more combustible elements available.

A position map was constructed to locate the fuel elements in the reactor core as these elements were chosen by the ants. This positioning map was constructed so as to favor the performance of a local heuristic, since for the choice of fuel elements it is necessary to evaluate a local heuristic and this choice is made without knowing the configuration as a whole. Local heuristics were not used in this article.

The positioning map attempts to make two fuel elements, which are chosen consecutively by the ants, occupy adjacent positions in the reactor core. Thus, the question of the proximity between the fuel elements, governed by some local heuristic, is guaranteed.

Figure 1 shows an example of an ordering map. The reactor used in this example was a PWR reactor of 121 combustible elements, such as Angra 1, using a 1/8 core symmetry. The numbers show the position that the fuel elements will occupy when they are chosen by the ants. Thus, the first chosen fuel element will occupy position 1, the second fuel element chosen will occupy position 2 and so on.

C				
6	7			
5	16	8		
4	17	15	9	
3	18	14	11	10
2	19	13	12	
1	20			

Figure 1 - Example of an ordering map for a reactor with 121 EC and 1/8 symmetry

In addition to the ordering map, it is necessary to include a variable in the equations that guide ACO-GENES. This need stems from the fact that two fuel elements chosen subsequently contribute differently to the final configuration result, depending on the positions occupied by these fuel elements in the reactor core.

In optimizing NRR the contribution to the final solution given by the fact that the fuel element s is preceded by the fuel element r depends on the positions of these fuel elements in the reactor core.

Thus, to incorporate this characteristic of the problem in the modeling, it is necessary to include a new variable in the equations that govern the ACO_GENES algorithm. This new variable represents the position of the fuel element r in the reactor core and thus the arcs (r, s) become the arcs (r, s, p). Thus, as we move from the fuel element r to the fuel element s in the p-position, the pheromone matrix will be updated differently as we move from the fuel element r to the fuel element s in the p+1 position.

The equations will have an additional dimension, according to Equation (5) below.

$$s = \begin{cases} \max \{[FE(r,s,p)]^{\delta} x[HE(r,s,p)]^{\beta}\} & if \quad q \le q_0 \\ Roulette & if \quad q > q_0 \end{cases} \tag{5}$$

where Roulette is given by the pseudo-random-proportional probability distribution, shown in Equation (6).

$$Roleta = \begin{cases} \dfrac{[FE(r,s,p)]^{\delta} x[HE(r,s,p)]^{\beta}}{\sum_{z \in J_k(r)} [FE(r,z,p)]^{\delta} x[HE(r,z,p)]^{\beta}} & if \quad s \in J_k(r) \\ 0 & if \quad s \notin J_k(r) \end{cases} \tag{6}$$

The local update rule will be given by Equation (7):

$$FE(r, s, p) = (1 - \rho) * FE(r, s, p) + \rho * FEzero \tag{7}$$

The global update rule will be given by Equation (8):

$$FE(r, s, p) = (1 - \alpha) * FE(r, s, p) + \alpha * (W / best\ result) \tag{8}$$

After completing the construction of all configurations, each one will be submitted to a Reactor Physics code that will calculate the value of the quantities required for optimization.

For a low-leak strategy the objective function used was as follows:

If peak factor > limit value then
    Objective function = peak factor
If no
    Objective function = 1 / Boron concentration
End if

In this function, for peak values of radial power above the limit value of the technical specification, we want to minimize them, because with these values above the limit this core configuration becomes invalid, and for values below this limit, where the configurations of core become valid, we want to maximize the critical concentration of soluble boron. With the use of this function we clearly realize that when we reach the limit value of the radial power peak of the technical specification (1,435), the value of boron tends to be stable or rise, and the value of the radial power peak oscillates below 1.435.

Many intelligent algorithms developed for optimization, such as the Genetic Algorithm (HOLLAND, 1975), are guided in the direction of good solutions by heuristics used to evaluate the generated solutions. These heuristics take into account the knowledge of the problem as a whole and are therefore called global heuristics.

The ACO-GENES algorithm, besides using a global heuristic, represented by the learning reinforcement, will also make use a local heuristic in a future work. The local heuristic takes into account the knowledge of only a part of the problem, which simplifies its application in relation to the global heuristic.

The local heuristic seeks to express the relation between any two elements of the set of elements of the problem, so that this relationship leads to an improvement of the final solution. In other words, through a knowledge that encompasses a small number of elements that are close to the ordering, we intend to obtain an improvement in the overall solution. Thus, to achieve a good local heuristic the question is: chosen a particular element, what will be the best fuel element, among the possible, to be used as the next element in the ordering sequence?

In the case of the optimization of nuclear reload, finding a satisfactory local heuristic is not a simple task, since this heuristic is referring to the process of burning the fuel elements during the production of energy and it is a process of great complexity involving nonlinear quantities .

So our analogy would look like this: if we have 20 fuel elements and these elements can occupy all the positions of the core (not taking into account quartets and octets), we have a problem equivalent to 400 cities, therefore, we have for each core position 20 initial possibilities and as we have 20 core positions, we have a total of 400 possibilities, or a total of 200 possibilities if we take into account the symmetries of quartets and octets.

In case of Brazilian NPP Angra 1, was used 200 ants. In other words, 10 FEs of quartet and 10 of octet, we will have 10 x 10 + 10 x 10 possibilities, equaling 200, so 200 ants.

4. GENE - PHYSICS REACTORS CODE

The GENES code was initially developed to model large PWR reactor cores. GENES makes use of the nodal expansion method (NEM) to solve the neutron continuity equation for two energy groups. In general, in this methodology, the volume of the reactor core is subdivided into volume nodes, $V_n = a_x^n a_y^n a_z^n$; transforming the equation of continuity, in its classical discrete form, namely:

$$\sum_{u=x,y,z}(J^n_{gur}-J^n_{gul})/a^n_u+\Sigma^n_{Rg}\phi^n_g=\frac{1}{k_{eff}}\chi_g\sum_{g'=1}^{2}\nu\Sigma^n_{fg'}\phi^n_{g'}+\sum_{\substack{g'=1\\g'\neq g}}^{2}\Sigma^n_{gg'}\phi^n_{g'} \tag{9}$$

where,

$$J^n_{gus}\equiv J^{+n}_{gus}-J^{-n}_{gus}=-D^n_g\frac{d}{du}\psi^n_{gu}(u)\Big|_{u=u^n_s} \tag{10}$$

and,

$$\phi^n_g\equiv\frac{1}{V_n}\int_0^{a^n_x}\int_0^{a^n_y}\int_0^{a^n_z}\phi_g(x,y,z)dxdydz \tag{11}$$

$$J^{\pm n}_{gus}\equiv\frac{1}{a^n_v a^n_w}\int_0^{a^n_v}\int_0^{a^n_w}J^{\pm}_{gu}(u^n_s,v,w)dvdw \tag{12}$$

$$\psi^n_{gu}(u)\equiv\frac{1}{a^n_v a^n_w}\int_0^{a^n_v}\int_0^{a^n_w}\phi_g(u,v,w)dvdw \tag{13}$$

Nuclear data are considered to be $V_n=a^n_x a^n_y a^n_z$.

After integrating the diffusion equation across the direction u (= x, y, z), we arrive at:

$$-D^n_g\frac{d^2}{du^2}\psi^n_{gu}(u)+\Sigma^n_{gu}\psi^n_{gu}(u)=\frac{1}{k_{eff}}\chi_g\sum_{g'=1}^{2}\nu\Sigma^n_{fg'}\psi^n_{g'u}(u)+\sum_{g'=1}^{2}\Sigma^n_{gg'}\psi^n_{g'u}(u)-L^n_{gu}(u) \tag{14}$$

where,

$$L^n_{gu}(u)\equiv-\frac{D^n_g}{a^n_v a^n_w}\int_0^{a^n_v}\int_0^{a^n_w}\left(\frac{\partial}{\partial v}\phi_g(u,v,w)+\frac{\partial}{\partial w}\phi_g(u,v,w)\right)dvdw \tag{15}$$

It is identified as the transverse leak direction u (= x, y, z).

In the nodal expansion method, the neutron flux at the face of the node is approximated with an expansion as follows:

$$\psi^n_{gu}(u)=\sum_{i=0}^{4}c^n_{igu}h_i(u/a^n_u) \tag{16}$$

Where the expansion coefficients can be obtained using the weighted residuals technique. Details of the methodology can be found in the classical literature and publications on nodal and coarse mesh methods.

The GENES Code has dynamic memory allocation; Thus, it allows a varied number of radial and axial nodes in the modeling.

Its contains classic features of a contemporary commercial depletion code, namely:

- realizes the depletion of the fuel, absorber material and fission products using the predictor-corrector method;

- calculates the isotopic concentrations for the following fission products: Iodine-135, Xenon-135, Promethium (147, 148 and 148m), Samarium-149;

- contains the chain of transmutation of Uranium 235 (U234-U235-U236) and Uranium 238 (U238-NP239-Pu239-Pu240-Pu241-Pu242-Am241-Am242-Am243);

- absorbent materials such as: Pyrex, WABA, IFBA, Erbium and Gadolinium are explicitly or implicitly modeled as desired;

- calculations of the main coefficients of reactivity can be performed, such as: moderator reactivity coefficient, fuel reactivity coefficient, isothermal reactivity coefficient and boron reactivity coefficient;

- criticality surveys can be performed by searching for shutdown margins;
- hot channel factors such as: Fdh, Fxy, Fz and Fq can be obtained through the nodal values. The code does not yet have intranodal flux reconstruction models;
- Thermal-hydraulic calculations are performed during criticality surveys, where temperature distributions are obtained for the moderator and nuclear fuel. The density distribution of the moderator reflects the increase in enthalpy present at the various power levels allowed in the system.
In general, the code contains properties suitable for modeling large commercial reactors.

The code can be used to model PWR reactors from several manufacturers, namely: Reactors Westinghouse, Areva, CE, etc.

5. EXPERIMENTAL RESULTS

The cycle chosen for comparison between the ACO-GENES system and a results by the ETN (ELETRONUCLEAR), was cycle 8, because this cycle is considered as an initial cycle due to internal events in Angra 1.

We performed several tests with standard configurations for the ACO, without using heuristics and only using 20 ants. It is not the complete form of interpretation for ACO-GENES. But they were preliminary tests to prove the viability of the system. The objective function used was to minimize the average power until reaching the safety limit, and after that to maximize the Boron concentration. And the Peak Power Factor power limit for Angra

1 is 1,435, and thus a penalty on the average power value (which is calculated by GENES) is given to the value of 1.25, in an attempt to ensure that The Power Peak factor would be below 1.435.

It was not possible to test all core configurations considered to be good in the licensed reactor physics code, since the ACO-GENES system has a different interface and the format of the input data different from the licensed code.

Below we show a table with the 5 best results found by the ACO-GENES system and also executed in the licensed code to validate the results, and the actual result of cycle 8 of Angra 1.

Table 1: Results

Case	Peak Power Factor	Boron concentration (ppm)
Real cycle 8	1.362	1108
Test 1	1.409	1218
Test 2	1.429	1195
Test 3	1.401	1258
Test 4	1.392	1186
Test 5	1.384	1208

Each test run lasted approximately 24 hours. The GENESs runs in 5 seconds, while the licensed code runs in 30 seconds.

As we can see, the result of Test 3 was the best result, obtaining a Boron concentration of 1258 ppm, against 1108 ppm of the real cycle. The Boron 150 ppm difference gives us approximately 44 EFPD plus operation, which could have provided a larger revenue to the ETN.

6. FUTURE WORK

Because good results have been obtained and in reasonably acceptable times, we believe that the ACO-GENES system can help to find good core configurations and other cycles. So we should update the shock sections of the FEs so we can do similar tests with current and future cycles of the Angra 1 nuclear power plant, and possibly also Angra 2.

With the use of parallelization techniques on several machines and the Compute Unified Device Architecture (CUDA) technology, we can accelerate GENES and thus also optimize the purchase of FEs and also find multi-cycle configurations, which would be a more complete form of the problem fuel reload problem, where we could optimize the characteristics of the new FEs, the quantity, and their uses in multi cycles, optimally exhausting the burning of the FEs.

REFERENCES

1. BOECHEL, T., 2003, Algoritmo de Otimização: Uma Abordagem Híbrida Utilizando o Algoritmo das Formigas e Genético, Tese de M.Sc., UFSC, Florianópolis, SC, Brasil.

2. DORIGO, M., GAMBARDELLA, L.M., 1997, “Ant Colony System: A cooperative Learning Approach to Traveling Salesman Problem”, IEEE Transactions and Evolutionary Computation, v.1, n.1, pp. 53-66.
3. GALPERIN, A., 1995, “Exploration of the Search Space of the In-Core Fuel Management Problem by Knowledge-Based Techniques”, Nuclear Science and Engineering, v. 119, pp. 144-152.
4. GAMBARDELLA, L.M., DORIGO, M., 1996, “Solving Symmetric and Asymmetric TSPs by Ant Colonies”, IEEE Conference on Evolutionary Computation, ICEC’96, Nagoya, Japan.
5. HOLLAND, J.H., 1975, Adaptation In Natural and Artificial Systems, Ann Arbor, University of Michigan Press.
6. KAELBLING, L.P., LITTMAN, L.M., MOORE, A.W., 1996, “Reinforcement Learning: a Survey”, Journal of Artificial Intelligence Research, v.4, pp. 237-285.
7. RUSSEL, S.J., NORVIG, P., 1995, Artificial Intelligence: A Modern Approach, Englewood Cliffs, NJ: Pretince Hall.
8. STUTZLE, T., DORIGO, M., 2000, ACO Algorithms for the Traveling Salesman Problem, IRIDIA, Université Libre de Bruxelles, Belgium.