Id.: EN43

PBIL algorithm and PCA Method applied to the Accident Identification Methodology with Don't Know response for PWR Nuclear Reactors

Diego J. S. N. de Souza[1], Andressa S. Nicolau[2], Roberto Schirru[3]
[123]Federal University of Rio de Janeiro (Av. Horácio Macedo 2030, Block G, Room 206, Rio de Janeiro)
diego.nuzza@poli.ufrj.br

Keywords: Nuclear Power Plant; Nuclear Accident Identification; Don't Know Response; PBIL; PCA Applications;

ABSTRACT

The Nuclear Accident Identification Problem (NAIP) is a critical issue faced by Nuclear Power Plants (NPPs), focusing on accurately and rapidly identifying an unknown occurrence in an NPP facility. The NAIP is approached by analyzing data from simulations of Design-Based Accidents (DBA) presented in the Final Safety Analysis Report (FSAR) of a Brazilian Nuclear Power Plant. This work proposes utilizing the Population-Based Incremental Learning (PBIL) Algorithm to model the NAIP in NPPs. The concept underlying the proposed methodology involves classifying anomalous events using data from normal operational conditions and three design basis accidents within the dataset of the plant state list simulated for the Brazilian Nuclear Power Plant Angra 2. In this approach, the NAIP is seen as a class separation problem, where PBIL is used to find the representative vector of each plant state class, aiming to maximize the number of correct classifications. The proposed method enhances classification based on Voronoi Diagrams to define the regions of influence for each plant state, enabling the generation of a "Don't-Know" response. The Don't-Know Response Generation (DKRG) helps differentiate normal and abnormal plant states, thereby aiding in timely decision-making and mitigating the impacts of potential nuclear accidents. In this study, the plant state variables will be selected using the Principal Component Analysis (PCA) method, an intelligent approach for choosing the most important plant variables for the problem. This research demonstrated that the PBIL algorithm is capable of obtaining a centroid vector with 100% accuracy for the four operating conditions outlined in the methodology. The algorithm was able to generate the "unknown" response for the tested operating conditions and remained robust against data noise up to 5%. The results of this proposed methodology will be compared with those from the literature.

1. INTRODUCTION

Nuclear Power Plants (NPPs) are important facilities in clean energy production, there are 440 nuclear power reactors in the world [1], representing 25% of the world's low-carbon electricity production. In a NPP, there are several protocols that guarantee safety during an abnormal event. On this occasion, the operators have to identify the situation in progress to be able to follow the protocols and maintain control. In order to help the operator identify an abnormal event that may occur, the Nuclear Regulatory Commission has made the presence of Safety Parameter Display Systems (SPDS) [2] necessary in the control room. These systems should help the operators by synthesizing the information available in the control room, by processing indicators from alarms and sensors.

In order to provide an accident diagnosis methodology to assist the operators in classifying an event, a methodology is needed that can classify an abnormal event with accuracy and quickness [3]. The literature shows that the academic community proposes prototype systems based on artificial intelligence algorithms for accident identification, such as the Genetic Algorithm [4], Swarm Algorithms like Particle Swarm Optimization [5], or Neural Networks [6].

This article proposes a methodology based on the Population-Based Incremental Learning (PBIL) Algorithm [7][8] combined with the Don't Know methodology [9] and the use of the Principal Component Analysis (PCA) [10][11] variables, to address the Nuclear Accident Identification Problem. The PBIL Algorithm is used to find the optimal representative vector, formed by prototype vectors representing each accident employed within the model. The Don't Know methodology has the objective to delineate the "influence zones" of the representative vectors found by the PBIL Algorithm, and to assign the Don't Know Response for events out of the influence zones of the representative vectors. The Principal Component Analysis is used to reduce the dimensionality of the problem, finding the best set of variables for the problem.

The proposed approach is evaluated using a case study that involves data from four operational conditions of a Pressurized Water (PWR) Reactor, using a simulated dataset of the Normal Operation, Loss of Coolant Accident (LOCA), Main Feed Water Break (MFWBR), and Steam Generator Tube Rupture (SGTR) at the Brazilian nuclear Power Plant PWR Angra 2.

2. METHODOLOGY

2.1. Population-Based Incremental Learning Algorithm

The PBIL algorithm was proposed by Bajula (1994) [12], and relies on the concept of competitive learning idea the basic structure of the Genetic Algorithm [13]. The PBIL is simpler than the GA, but can outperform it in the optimization of certain problems [8]. In this algorithm, the individuals are created at each generation based in a probability distribution vector called vector *P*, where each component represents the probability of a bit being set to 1. In PBIL, the individuals are encoded in binary, and the population is generated based on vector *P*.

In GA, crossover and mutation operators are used to work on the population to find the best solution. In contrast, PBIL does not use such operators but instead focuses on optimizing the probability vector *P*. The goal is to adjust the *P* vector to have high probabilities for generating individuals that represent the best solutions for the problem. The *P* vector is initialized with all components set to 0.5, as shown in Fig. 1.

P =	0.5	0.5	0.5	…	0.5	0.5	0.5

Fig 1. Representation of the P vector, at start.

This ensures that every region of the search space has the same probability of being chosen; in this case, the probability of generating the value '0' or '1' is the same for each bit. From vector *P*, the binary encoded individuals are formed. After the individuals are formed, they are evaluated, receiving a fitness value, and the best and the worst individuals are used in the optimization process.

Knowing the best and the worst individuals, the vector *P* is updated. The approach involves adjusting the *P* vector closer to the best individual and further from the worst individual. The update uses the parameters *Lr_P* and *Lr_N*, which symbolize the positive learning rate and the negative learning rate, respectively. The process is iterative: in each generation, the vector *P* is adjusted to be closer to the best solution and further from the worst solution. After updating the vector *P*, the individuals produced are more similar to the best solution and less similar to the worst solution in each generation. The aim of the PBIL algorithm is to evolve the *P* vector to resemble the best solution, as depicted in Fig 2, considering the best solution as the vector *[1,0,0,...,1,1,0]*.

P =	0.98	0.03	0.01	…	0.96	0.99	0.02

Fig 2. Representation of the P vector, at the end.

The *P* vector shown in the Fig. 2 is likely to produce solutions similar to the vector *[1, 0, 0,..., 1, 1, 0],* with high probability. Being *P* a vector with *K* bits, and being I the index of the

bits present in the list *[0, 1, 2,…, k-1],* the incremental in the learning is performed according to Eq. 1 and Eq. 2, for positive incremental learning and negative incremental learning, respectively. In this work, negative incremental learning was not used: assuming $Lr\_N = 0$, and this algorithm will be referred to as PBIL_S.

$$P[i] = (1 - Lr\_P)P[i] + Lr\_P * (Best[i]) \quad (1)$$

$$P[i] = (1 - Lr\_N)P[i] + Lr\_N * (Worst[i]) \quad (2)$$

2.1.1. PBIL_N

The PBIL_N [9] is a variation of the original PBIL algorithm, and uses the information of the *N* best individuals in each generation to update the vector *P*. The learning rate is dedicated to generating new individuals similar to the *N* individuals of the entire population. Being P the probability vector with size *K*, *i* the index of the bits *[0, 1, 2,…, K-1]* and *j* being the index of bests solution, being the list *[0, 1, 2,…, N]* the Eq. 3 shows the updating of the probability vector components in PBIL_N. The Eq. 4 shows how the learning rate *Lr* is defined. The *fitness_N* represents the fitness level of the current individual and the *fitness_O* is the fitness level of the best-evaluated individual.

$$P[i] = (1 - Lr)P[i] + Lr * (Best[j][i]) \quad (3)$$

$$Lr = Lr \times \frac{Fitness\_N}{Fitness\_O} \quad (4)$$

2.2. Don’t Know Methodology

To accurately generate a 'Don't Know' response, the model needs to rule out the hypothesis that the abnormal event being analyzed is one of the accidents present in the model. This classification is challenging because it requires precisely defining when an abnormal event will be classified as an accident. If an event cannot be classified as any known accident, a 'Don't Know' response will be generated. Nicolau (2014) [3] proposed a methodology based on Voronoi diagrams to define the influence areas of accidents.

Voronoi diagrams divide a plane into *N* regions, each represented by a centroid vector such that any point in a region is closer to its centroid than to any other. Implementing the 'Don't Know' response methodology involves determining the representative vectors for accidents using an optimization algorithm, in this case, was used the PBIL algorithm. Once the representative vectors are determined, the influence area of a centroid is defined as half the smallest distance between that vector and the others. This gives the influence radius of the accident. For an event to be classified as an accident, it must fall within this influence radius — the distance between the event and the centroid must be smaller than the influence radius. This methodology allows the model to classify abnormal events that fall outside the influence radius of any accidents as 'Don't know.'

2.3. Objective Function (Fitness)

The use of the PBIL algorithm aims to find the centroid vector of the considered accidents. In this work, four operational conditions were considered: NORMAL (Normal operation condition), LOCA (Loss of Coolant Accident), MFWBR (Main Feed Water Break) and SGTR (Steam Generator Tube Rupture). The algorithm optimizes the number of correct identifications made by the representative vector, as illustrated in Fig. 3, where each space represents a centroid vector.

vector =	…	…	…	…

Fig. 3. Illustration of the centroid vector.

The representative vector is the optimized vector, with most right classifications, of events. The evaluation function, fitness, has two steps. First, the classification is made by the Euclidean distance of the centroid of accidents and the unknown event in process, classifying by the minimum distance. For step number two, to being considered a correct classification the abnormal event must be within the influence area of the accident.

This study utilized 59 seconds from the simulation made by Alvarenga (1997) [14] and four operation condition, the maximum number of correct classifications is 472, being 236 from step 1 and 236 from step 2, these 236 correct classifications are 59 from each accident. With the 2-step classification, for an event to be classified as an occurrence of an accident, it should be within the influence area of the centroid vector of that accident. The events that are out of the influence area of all the centroids, will receive the Don't Know Response, and will not count as a correct classification for the solution.

2.4. Dataset Preparation

For this study, it was used the dataset from the simulation made by Alvarenga (1997) [14] for the PWR power plant of Angra 2, in Rio de Janeiro, Brazil. The dataset simulates the evolution of 18 plant variables over 61 seconds of some operational conditions for a PWR reactor at 100% power level. From the simulation of Alveranga, eight operational conditions were used in this work: NORMAL (normal operation), LOCA (Loss of Coolant Accident), MFWBR (Main Feed Water Break), SGTR (Steam Generator Tube Rupture), BLACKOUT (Loss of electrical power), MFWISO (Main Feed Water Isolation), STMLIBR (Steam Line Break) and MSTMISO (Main Steam Isolation).

To maintain consistency, all datasets were normalized using the MAX-MIN technique. This method involves using the maximum and minimum values of each plant variable to linearly scale the datasets within the [0,1] range, as shown in the Eq. 5.

$$N(x) = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{5}$$

2.5. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a valuable technique in statistical signal processing, primarily used for reducing the dimensionality of datasets, which is beneficial for tasks such as pattern recognition, understanding, and interpreting data [11]. PCA was applied to reduce the dimensionality of the dataset by calculating the principal components, which are new variables formed by linear combinations of the original variables [10]. The purpose of applying PCA in this study was to determine which plant variables are more important for problem resolution.

This methodology was previously introduced by de Souza (2024) [10], using PCA to extract the best plant variables by analyzing their contributions to forming the principal components. This method reduces the dimensionality from 17 plant variables to 4 plant variables. The plant variables selected by de Souza (2024) are presented in Tab. 1.

Tab. 1. Plant variables selected using the PCA method proposed by de Souza, 2024.

Number of Variable	Name of Variable	Unit
8	Feedwater flow	Kg/s
10	Flow in the rupture	Kg/s
15	Nuclear Power	%

17	Pressurizer level	%

2.6. Method validation

The methodology for PBIL_S and PBIL_N algorithms were developed in Python. The algorithms were used to find the representative vector for 4 accidents within the set of variables presents by the Tab. 1. The parameters for the algorithms are present in the Tab. 2. and Tab. 3. The set of parameters were used following experiments present in the literature [8].

Tab. 2. Parameters for PBIL_S experiment.

Number of Population	**Learning rate**
50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150	0.001, 0.003, 0.01, 0.03, 0.05, 0.08

Tab. 3. Parameters for PBIL_N experiment.

Number of Population	**Learning rate**	**Number of Best Individuals**
50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150	0.001, 0.003, 0.01, 0.005,0.007,0.01	2,3,5

After the parameter sensitivity test results of the algorithms, the best vector - the vector that achieves the maximum number of correct classifications in the fewest generations - from the best parameter group - which has the lowest average number of generations, will undergo testing for generating 'Don't Know' responses for the BLACKOUT, STMLIBR, MFWISO, and MSTMISO incidents. This testing will involve introducing 1%, 2%, and 5% noise in the data. The data noise will be applied in the vector, as if the centroid vector were moved a little.

3. RESULTS AND DISCUSSION

Tab. 4. shows the classification results achieved by the PBIL_S algorithm. From the set of parameters listed in Tab. 2., a subset was selected for presentation in Tab. 4. The routine was executed 10 times, and the results shown represent the average generation at which the algorithm achieved the maximum number of correct classifications.

Tab. 4. Results for PBIL_S experiment.

Population	**Learning rate**	**Average Number of generations**
50	0.03	66.9
	0.05	43.1
	0.08	31.6
100	0.03	48.4
	0.05	38.5
	0.08	24.4
150	0.03	43.6
	0.05	30.7
	0.08	22

Tab. 4. shows that as the population size and learning rate increase, the average number of generations decreases. This means that the algorithm can achieve the maximum number of classifications in fewer generations, resulting in lower computational cost. Tab. 5. Shows the classification results achieved by the PBIL_N algorithm. From the set of parameters listed in Tab. 3, a subset was selected for presentation in Tab. 5. The routine was executed 5 times, and the results shown represent the average generation at which the algorithm achieved the maximum number of correct classifications.

Tab. 5. Results for PBIL_N experiment.

Population	**Number of bests individuals**	**Learning rate**	**Average Number of Generations**
50	2	0.003	255.6

		0.005	102.4
		0.01	100.6
	3	0.003	190.4
		0.005	130.4
		0.01	69.8
	5	0.003	151.2
		0.005	97.0
		0.01	53.4
100	2	0.003	179.6
		0.005	135.6
		0.01	81.2
	3	0.003	171.0
		0.005	94.4
		0.01	54.2
	5	0.003	115.4
		0.005	70.4
		0.01	40.4
150	2	0.003	167.0
		0.005	136.2
		0.01	65.2
	3	0.003	118.4
		0.005	83.8
		0.01	56.6
	5	0.003	106.4
		0.005	61.6
		0.01	38.2

Tab. 5. shows that, like PBIL_S, an increase in population size and learning rate for PBIL_N allows the algorithm to achieve lower computational costs in finding a vector that maximizes the number of classifications. For PBIL_N, it is also evident that the reduction in computational cost is further influenced by increasing the number of N best solutions, reaching the lowest average number of generations at 38.2. Tab. 6. presents the experiment conducted with the best vector from the group of vectors that achieved the maximum number of classifications with the lowest average number of generations. For PBIL_S, this vector had a population size of 150 and a learning rate of 0.08. For PBIL_N, the vector had a population size of 150, a learning rate of 0.01, and an N value of 5. Tab. 7. provides a summary of the results presented in Tab. 6. In Tab. 7., it is provided the average accuracy for 1%, 2%, 5% and the mean accuracy.

Tab. 6. Comparison between algorithms for Don't Know Generation with data noise

Data Noise	**Unknown Operation Condition**	**Response operation condition**	**Pbil_S Accuracy**	**Pbil_N Accuracy**	**COA Accuracy**
1 %	LOCA	LOCA	100 %	100 %	100 %
	MFWBR	MFWBR	100 %	100 %	100 %
	SGTR	SGTR	100 %	100 %	100 %
	NORMAL	NORMAL	100 %	100 %	100 %
	BLACKOUT	DON'T KNOW	100 %	100 %	100 %
	MSTMISO	DON'T KNOW	100 %	100 %	100 %
	MFWISO	DON'T KNOW	100 %	100 %	100 %
	STMLIBR	DON'T KNOW	100 %	100 %	100 %

2%	LOCA	LOCA	100 %	100 %	100 %
	MFWBR	MFWBR	100 %	100 %	100 %
	SGTR	SGTR	100 %	100 %	100 %
	NORMAL	NORMAL	100 %	100 %	100 %
	BLACKOUT	DON'T KNOW	100 %	100 %	100 %
	MSTMISO	DON'T KNOW	100 %	100 %	100 %
	MFWISO	DON'T KNOW	100 %	100 %	100 %
	STMLIBR	DON'T KNOW	100 %	100 %	100 %
5 %	LOCA	LOCA	100 %	100 %	100 %
	MFWBR	MFWBR	100 %	100 %	100 %
	SGTR	SGTR	100 %	100 %	100 %
	NORMAL	NORMAL	100 %	100 %	100 %
	BLACKOUT	DON'T KNOW	100 %	100 %	100 %
	MSTMISO	DON'T KNOW	100 %	100 %	100 %
	MFWISO	DON'T KNOW	86 % (*)	100 %	98 % (*)
	STMLIBR	DON'T KNOW	100 %	100 %	100 %

(*) Was misclassified as MFWBR.

Tab. 7. Resume of results from Tab. 6.

Algorithm	**Data Noise**	**Accuracy**
PBIL_S	1 %	100 %
	2 %	100 %
	5 %	98,25 %
	Mean	99,42 %
PBIL_N	1 %	100 %
	2 %	100 %
	5 %	100 %
	Mean	100 %
COA	1 %	100 %
	2 %	100 %
	5 %	99,75 %
	Mean	99,92%

Tab. 6. and Tab. 7. demonstrate that all algorithms find vectors that are robust to instrumentation error, defined here as up to 2% noise in the data. Particularly, only the vector found by PBIL_N shows resilience to 5% noise in the data achieving 100% accuracy, followed by the vector found by COA, achieving 99.75% accuracy under 5% data noise, and PBIL_S, achieving 98.25% accuracy.

4. CONCLUSIONS

This article explores the Population-Based Incremental Learning Algorithm's ability to address the Nuclear Accident Identification Problem (NAIP), building on the studies by de Souza et al. (2024). It implements a methodology for accident identification with the generation of 'Don't Know' responses based on nearest neighbor theory, enabling real-time diagnostic support systems. The variable set used was identified using Souza et al.'s 2024 method, which employs Principal Component Analysis for variable reduction.

This study includes a sensitivity analysis of PBIL algorithm parameters in two variations of the algorithm, PBIL_S and PBIL_N, demonstrating both variants can tackle the NAIP. According to Tab. 4., Tab. 5. and findings by de Souza et al. (2024), PBIL_S can identify the vector with the lowest computational cost. Moreover, Tab. 6. and Tab. 7. show that PBIL_N can identify a vector less sensitive to data noise, achieving 100% accuracy even submitted to

5% data noise. The three algorithms shown in this work, PBIL_S, PBIL_N and COA, are capable of finding optimal results. The work shows the details in comparison when such algorithms are subjected to instrumentation error. With the results of this study, it is possible to say that the PBIL algorithm can find a better centroid vector than the COA algorithm in fewer generations, resulting in lower computational cost.

ACKNOWLEDGMENTS

The authors would like to acknowledge CAPES (Coordination for the Improvement of Higher Education Personnel) for financial support.

REFERENCES

[1] World Nuclear Association. (2024, May 7). Nuclear power in the world today. Retrieved from [https://world-nuclear.org/information-library/current-and-future-generation/nuclear-power-in-the-world-today#1]
[2] C. M. N. A. Pereira et al. (1998) Learning an Optimized Classification System From a Data Base of Time Series Patterns Using Genetic Algorithm. In: Ebecken, N. F. F (ed), I&& Mining, 1 ed., Computational Mechanics Publications, WIT Press, England. [https://www.witpress.com/elibrary/wit-transactions-on-information-and-communication-technologies/22/6924]
[3] A. S Nicolau, (2014). Algoritmo Evolucionário de inspiração quântica aplicado na otimização de problemas da engenharia nuclear. (Doctoral dissertation, Universidade Federal do Rio de Janeiro). [http://antigo.nuclear.ufrj.br/DScTeses/teses2014/Tese_Andressa_Nicolau.pdf]
[4] V. H. C. Pinheiro & R. Schirru. (2019). Genetic programming applied to the identification of accidents of a PWR nuclear power plant. Annals of Nuclear Energy, 124, 335-341. [https://doi.org/10.1016/j.anucene.2018.09.039]
[5] D. J. S. N. de Souza. 2023. Algoritmos de Inteligência Artificial Aplicados no Problema de Identificação de Acidentes Nucleares para Usinas Nucleares do Tipo PWR. [http://www.repositorio.poli.ufrj.br/monografias/projpoli10042305.pdf]
[6] V. H. C. Pinheiro et al. (2019). Deep rectifier neural network applied to the accident identification problem in a PWR nuclear power plant. Annals of Nuclear Energy, 133, 400-408. [https://doi.org/10.1016/j.anucene.2019.05.039\
[7] G. H. Caldas, and R. Schirru, (2008). Parameterless evolutionary algorithm applied to the nuclear reload problem. Annals of Nuclear Energy, 35(4), 583-590. [https://doi.org/10.1016/j.anucene.2007.08.014]
[8] M. D. Machado. (2005). Algoritmo evolucionário PBIL multi-objetivo aplicado ao problema da recarga de reatores nucleares. DSc thesis, COPPE/UFRJ, Brazil. [https://www.nuclear.ufrj.br/images/Tese_DSc_Marcelo_D_Machado.pdf]
[9] A. S. Nicolau and R. Schirru. (2017). A new methodology for diagnosis system with 'Don't Know' response for Nuclear Power Plant. Annals of Nuclear Energy, 100, 91-97. [https://doi.org/10.1016/j.anucene.2016.10.018]
[10] D. J. S. N. de Souza et al. (2024). Accident classification methodology with don't know response for PWR nuclear reactors using the cuckoo optimization algorithm and principal component analysis method. Nuclear Engineering and Design, 423, 113200. [https://doi.org/10.1016/j.nucengdes.2024.113200]
[11] K. Y. Li et al., (2021). An automated machine learning framework in unmanned aircraft systems: new insights into agricultural management practices recognition approaches. Remote Sensing, 13(16), 3190. [https://doi.org/10.3390/rs13163190]
[12] S. Baluja. (1994). Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Carnegie Mellon University.
[13] D. E. GOLDBERG, "Genetic Algorithms in Search, Optimization & Machine Learning", Addison-Wesley, Reading, MA, USA (1989).
[14] M. A. B. Alvarenga, (1997). Diagnóstico do desligamento de um reator nuclear através de técnicas avançadas de inteligência artificial (Doctoral dissertation, Thesis of D. Sc., COPPE/UFRJ, Rio de Janeiro Brazil).