KAIRÓS: Intelligent System for Scenarios Recommendation at the Beginning of Software Process Improvement

Ana Marys Garcia Rodríguez, Yadian Guillermo Pérez Betancourt , Juan Pedro Febles Rodríguez,
Yaimí Trujillo Casañola and Alejandro Perdomo Vergara
Universidad de las Ciencias Informáticas, La Habana, Cuba
E-mail1: agarcia@uci.cu, ygbetancourt@uci.cu, febles@uci.cu, yaimi@uci.cu, apvergara@uci.cu

Software Process Improvement provides benefits to organizations. However, improvement efforts are not guided by the combined use of Good Practices and Critical Factors that influence success. Resources are dedicated without a prior analysis that guides the actions intentionally. The objective of this research is to support decision-making in Software Process Improvement. To achieve this, an intelligent system is conceived, which based on association rules, identifies dependencies between Good Practices and Critical Success Factors. In addition, this system implements a Genetic Algorithm to optimize improvement scenarios and an evolutionary Artificial Neural Network to predict success in Software Process Improvement. The methods used to validate the results corroborated the contribution and usefulness of the proposal.

Povzetek: Predstavljen je inteligentni sistem KAIROS, ki na osnovi metod umetne inteligence zasnuje scenarij izdelave sistema pred začetkom softverskega procesa.

1 Introduction

The analysis around the Critical Success Factors that influence the Software Process Improvement (SPI), allows to infer that its use in function of the organizational contexts, contributes to the success of the improvement project [1; 2]. In spite of the advances in the treatment of Critical Success Factors [3; 4; 5], insufficiencies associated with the reuse of knowledge persist. This hinders to obtaining evaluations that are close to reality and makes it difficult to provide scenarios that guide organizations in the improvement. Also, the influence performed by the combination between Critical Success Factors and Good Practices in the SPI is not analyzed [6]. The weights assigned to the Critical Success Factors are not adjustable and their relevance changes according to the context.

An analysis to guides organizations at the beginning of the SPI is appropriate. It is cumbersome to process information when a large number of elements affects the decision-making of an organization. An effective alternative is the application of artificial intelligence techniques that transform SPI experiences into useful knowledge to guide insertion in an improvement project.

The research problem is: how to recommend improvement scenarios from the use of Critical Success Factors and Good Practices, to support decision-making at the beginning of the SPI? The objective of this paper is to develop an intelligent system for scenarios recommendation, which combines the use of Critical Success Factors and Good Practices to decision-making

support at the beginning of the SPI. To the development of this research were used some scientific methods:

- Historical-logical and dialectical to the critical analysis of researches associated with the use of Critical Success Factors and Good Practices in the SPI.
- Induction-deduction to the identification of the problem, as well as its solution variants.
- Hypothetical-deductive to the proposal of this research line.
- Analytical-synthetic to the decomposition of the problem in elements that allow its analysis.
- Bibliographic analysis for literature review.
- Survey to know the degree of customer satisfaction with the system developed.
- Experimental to evaluate the utility of the obtained results.
- Consult experts to the research validation.
- Focal group to the conceptualization of Good Practices and recommendations.
- Iadov technique to evaluate the solution satisfaction.
- Statistical methods to the analysis of applied surveys.

 Scientific contributions of the research:
- An informatics system (KAIRÓS) which combines artificial intelligence techniques, to support decision-making in SPI. In order to achieve this, the system optimizes improvement scenarios and predicts their success in the SPI.
- A genetic algorithm (GA) to optimize improvement scenarios from the redefinition of selection and

crossover operators and from the definition of a new mutation operator.

- An evolutionary ANN that uses genetic algorithms to topology design, and integrates the backpropagation algorithm and genetic algorithms for the net learning. Also, it uses the Principals Components Analysis technique to handle the changeful number of neurons in the input layer.

2   Theoretical bases

To solve the introduced problem, it was realized a research study to clarify the approach of the Good Practices use in SPI. In addition, it was analyzed the artificial intelligence techniques that facilitate the reutilization of Good Practices and Critical Success Factors experiences.

2.1   A survey of Good Practices and Critical Success Factors association in SPI

The literature analysis reveals the need to apply Good Practices for a successful SPI projects execution. However, only four papers consider the influence of Good Practices on the behavior of Critical Success Factors [5; 7; 8; 9]. These papers contribute important elements, associated with the Good Practices incorporation to have a positive influence in Critical Success Factors behavior. However, there are some insufficiencies that affect the use of this relationship:

- The dependencies between Good Practices and Critical Success Factors are considered without detailing which are the relationships specifically.
- The experiences reuse is assumed, but based on the Critical Success Factors and without considering the influence of Good Practices or their combined use.
- Trujillo [5] defines the Critical Success Factors and their measurements, establishes the weighting coefficient of Critical Success Factors, but does not assess its dynamic treatment.
- Improvement scenarios are not offered to support decision-making of organizations in the SPI.

In this sense, it is necessary to extend the treatment of Good Practices and Critical Success Factors, and to consider the influence performed by the combination of Good Practices on Critical Success Factors. Also, the dynamism of Critical Success Factors relevance, must be taken into account.

2.2   Artificial intelligence applied to SPI

For the forecast and recommendation of scenarios before the investment in SPI, it was considered the application of artificial intelligence techniques. With the aim of supporting decision-making in SPI from two perspectives: guide the efforts of organizations towards better scenarios in the SPI and forecast the result prior to investing in the SPI, the experiences reuse, associated with Critical Success Factors and Good Practices is adopted. In this sense, three needs were identified:

- Recommendation of scenarios to improve an organization initial state, prior to invest in SPI. It is considered as an optimization problem and is solved with the implementation of a GA [10; 11].
- Identification of relationships between Good Practices and Critical Success Factors measurements. It is considered as an association problem, taking into consideration the dependencies identification between dependent variables (Critical Success Factors measurements) and independent variables (Good Practices), whether metric or non-metric. It is solved with the use of association rules [12; 13].
- Forecast of success or failure of scenarios in SPI. It is considered as a classification problem, where it is necessary to identify the tendency to success or failure of an organization in the SPI. It is solved by the implementation of an evolutionary Artificial Neural Network (ANN) [14; 15; 16].

2.2.1   Considerations for KAIRÓS optimization

The GA of KAIRÓS for scenarios optimization takes the functioning principles from the operators described by the literature and provides new operators, capable to solve the particularities of the problematic. It was considered to not provide solutions unattainable since the organization capabilities to the implementation of selection operator. To the crossover and mutation operators, it was considered to not change the values of the Critical Success Factors measurements, which are not affected by the good practices applicable by the organization.

In this research, the chromosomes represent the initial state and the improvement scenarios. Genes are measurements of the Critical Success Factors, where $M = \{m \in R,\ 0 \leq m \leq 1$: m is a measurement of Critical Success Factors$\}$.

Selection operator redesign:

Several operators were analyzed where some was discarded and others partially satisfy the solution:
- *Selection by roulette* is ruled out, due to the randomness factor that it uses.
- *Selection by tournament* is ruled out, due to its high computational cost.
- *Hierarchical selection* and *selection by rank* partially satisfy the needs of the problem, because they do not necessarily obtain individuals close to the initial state.
- *Selection by rank* partially satisfies, because it does not consider fitness.

For those reasons, selection operator was redesigned from the hierarchical ordering of the population chromosomes, taking as criterion of order the fitness of each individual. Then a range of individuals is selected, which will be closest to the initial state.

Crossover operator redesign:

A set of operators were analyzed where were discarded:

- *Crossover by a point* and *Crossover by N points,* because the measurements influenced by Good Practices do not have a specific order or position within the chromosome.
- *Arithmetic crossover,* because it does not allow to establish which measurements will be modified, these are determined by the randomness.
- *Uniform crossover,* because it affects all the genes in the chromosome.

Finally, the authors determined that *uniform crossover* with binary mask is the operator that satisfies the most of the problem needs. However, the randomness when generating the binary mask does not make it feasible to the solution. Therefore, it was redesigned in such a way that the binary mask is intentionally generated from the dependencies between Good Practices to be applied and measurements of Critical Success Factors. The dependencies are obtained by applying association rules between these variables.

Mutation operator design:

A set of operators were analyzed where were discarded:

- *Binary mutation,* because it does not correspond to the coding of the chromosomes in this research.
- *Mutation to the edge* and *Uniform mutation,* exchange values of the attributes, which can alter genes not influenced by the Good Practices.

Therefore, a new operator was designed that uses the binary mask of the redesigned crossover operator and randomly identifies a position of the mask. If the position value is 1, proceed to mutate by 1% in this position.

2.2.2 Considerations for KAIRÓS dependences identification

The dynamic identification from the accumulated experience of the association relationships between Good Practices and Critical Success Factors measurements, is relevant. The objective is to know what measurements to enhance in the optimization process, based on the Good Practices that the organization can apply. To determine the dependencies association rules are applied, due to their potential to identify relationships between variables in combination, as well as the treatment of both metric and non-metric variables.

To generate the rules combinations the algorithm *Apriori* is used [17; 18; 19], with the aim of reducing the number of candidates through the technique of reduction by pruning. In this sense, all variants of rules whose elements are not frequent are discarded, because their combinations will not be.

2.2.3 Consideration for KAIRÓS classification

To predict the success of initial state and improvement scenarios, it must be taken into account that:

- The weights relevance associated to the Critical Success Factors, must have a dynamic treatment.
- The Critical Success Factors and their measurements can change over time.

Based on the above, an ANN is implemented because it favors learning by readjusting the weights associated with network connections. Considering that a classification problem is addressed, is appropriate to use supervised learning, specifically the multilayer perceptron. This architecture is usually trained using the backpropagation algorithm. However, an architecture that provides a solution to one problem can't be used to solve another [16]. Under the conditions of the problem, the construction of a self-adapting intelligent system based on ANN is required. This research considers the use of an evolutionary ANN, which allows adapting to the input patterns. Genetic algorithms are applied to the design and learning of the evolutionary network.

3 Intelligent system KAIRÓS

In this article, the Critical Success Factors and its measurements defined by Trujillo [5] are assumed for the processing of KAIRÓS. In addition, with the aim of defining Good Practices to improve the behavior of the Critical Success Factors, the bibliographic review, Delphi and focus group methods were applied.

For the identification of Good Practices, a bibliographic review of 77 articles and documented experiences was made, of which 15 allude to the use of Good Practices to diminish the influence of the Critical Success Factors in the SPI [4; 8; 20; 21; 22; 23; 24; 25; 26; 27; 28; 29; 30; 31; 32]. Then it was refined with the help of experts, managers and members of software development organizations, using the Delphi method in a first round. The results were submitted to the exploratory focal group, where the proposal was enriched with the recommendations for the execution of the Good Practices. Finally, a second round of Delphi method was applied with the refined information. As a result, 49 Good Practices and 127 recommendations that guide its application were defined [33].

KAIRÓS [34; 35] has the purpose of processing and automating the information associated with Critical Success Factors and Good Practices to support decision-making in SPI. The following describes its components.

3.1 GA for scenarios optimization in SPI

For the proposal of improvement scenarios, a GA is conceived [34]. The restriction of the optimization problem is associated to achieve a distance between the initial state and the improvement scenario in an affordable range, attending to the Good Practices that the organization can apply.

GA description:

Step 1. Generate initial population: the size of the initial population sample is calculated with the probabilistic method of calculating the population sample size, knowing the population size [36]. The individuals of this population are taken randomly from the knowledge base.

Step 2. Select scenarios: the scenarios of the initial population are assessed with the evaluation function (equation 1). The scenarios are ordered hierarchically, according to their fitness (value of evaluation function).

Then, a population sample corresponding to a new calculation of sample size is selected.

$$f_{eval} = \sum_{i=1}^{n} f(Sn(i), Sm(i)) \, Dg(Ov(i), Sm(i))/n \quad (1)$$

Where:

f_{eval} is the evaluation function.

n is the amount of measurements (genes) of the scenario (chromosome).

i is the position of the measurement (gene) subject to analysis.

$Sn(i)$ is the value of the measurement (gene) in the position i of initial state (chromosome).

$Sm(i)$ is the value of the measurement (gene) in the position i of the scenario (chromosome) to be analyzed.

$f(Sn(i), Sm(i))$ is the aptitude function for the measurement (gene) i and is determined by the equations 2 and 3.

$Ov(i)$ is the optimum value achievable for $Sm(i)$ (obtained from the analysis of association between Good Practices and Critical Success Factors measurements).

$Dg(Ov(i), Sm(i))$ is the degree of improvement for the measurement (gene) i.

The aptitude function $f(Sn(i), Sm(i))$ for each measurement, it is calculated depending on whether the measurement under analysis i, belongs or not to the set of measurements that are affected by the association between Good Practices and Critical Success Factors measurements.

Being:

$M = \{m \in R, \ 0 \leq m \leq 1: m$ is a measurement of Critical Success Factors$\}$

$MGP = \{mgp \in R, \ 0 \leq mgp \leq 1: mgb$ is a measurement affected by Good Practices$\}$

$MGP \subset M$

i is the position of the measurement (gene) subject to analysis.

$m(i)$ is the measurement in position i, represented to evaluate whether the measurement in a given position is affected by the association with Good Practices, $m(i) \in M$.

If $m(i) \in MGP$

$$f(Sn(i), Sm(i)) = \begin{cases} 1 & si \quad Sn(i) < Sm(i) \\ 0,5 & si \quad Sn(i) = Sm(i) \\ 0 & si \quad Sn(i) > Sm(i) \end{cases} \quad (2)$$

Where:

$Sn(i)$ is the value of the measurement (gene) in the position i of initial state (chromosome).

$Sm(i)$ is the value of the measurement (gene) in the position i of the scenario (chromosome) to be analyzed.

If $m(i) \notin MGP$

$$f(Sn(i), Sm(i)) = \begin{cases} 1 & si \quad \delta(Sn(i), Sm(i)) = 0 \\ 0 & si \quad \delta(Sn(i), Sm(i)) = 1 \end{cases} \quad (3)$$

Where:

$Sn(i)$ is the value of the measurement (gene) in the position i of initial state (chromosome).

$Sm(i)$ is the value of the measurement (gene) in the position i of the scenario (chromosome) to be analyzed.

$\delta(Sn(i), Sm(i))$ is the distance between the value of $Sn(i)$ y $Sm(i)$ and it is calculated by:

$$\delta(Sn(i), Sm(i)) = \begin{cases} 1 & si \quad Sn(i) - Sm(i) \neq 0 \\ 0 & si \quad Sn(i) - Sm(i) = 0 \end{cases} \quad (4)$$

The improvement degree is calculated by subtracting one from the normalization of the difference between the optimum value achievable by the measurement and the value of said average.

$$Dg(Ov(i), Sm(i)) = 1 - |Ov(i) - Sm(i)| \quad (5)$$

Where:

$Sm(i)$ is the value of the measurement (gene) in the position i of the scenario (chromosome) to be analyzed.

$Ov(i)$ is the optimum value achievable for $Sm(i)$ (obtained from the analysis of association between Good Practices and measurements of the Critical Success Factors).

Step 3. Check if the solution is among the selected scenarios: if the last scenario of the population sample fitness exceeds the 0,75 threshold, returns the first and last chromosome from the population, else step 4 is executed.

Step 4. Cross scenarios: the binary mask is generated by assigning 1 to the positions of measurements favored by the Good Practices and 0 to the rest of the positions. For each gene of the scenario if its position corresponds with value 1 in the binary mask, the gene of the scenario being analyzed is added to the new scenario. If its position corresponds with value 0 in the mask, the gene of the initial state is added to the new scenario. Finally, the new scenario is added to the set of crossed scenarios.

Step 5. Mutate scenarios: the same binary mask used in the crossover is applied for mutation. A random number greater than 0 and less than the number of Critical Success Factors measurements that compose the initial state, is generated. If the random number coincides with the position of some measurement affected by the association, the value of this measurement will increase by 1%, otherwise the scenario in the mutation process will be ignored. Finally, the new scenario is added to the set of mutated scenarios.

Step 6. Increase population: the crossed and mutated scenarios are added to the population.

Step 7. Execute step 2.

3.1.1 Association rules to identify dependencies between Good Practices and Critical Success Factors measurements

Several practices can influence more than one measurement of the Critical Success Factors. It is considered relevant to identify dynamically the association relationships between Good Practices and Critical Success Factors measurements from the accumulated experience. For this, association rules are applied in the present research.

$GP = \{gp: gp$ is an SPI action that decreases the negative influence of Critical Success Factors$\}$

$M = \{m \in R, \ 0 \leq m \leq 1: m$ is a measurement of Critical Success Factors$\}$

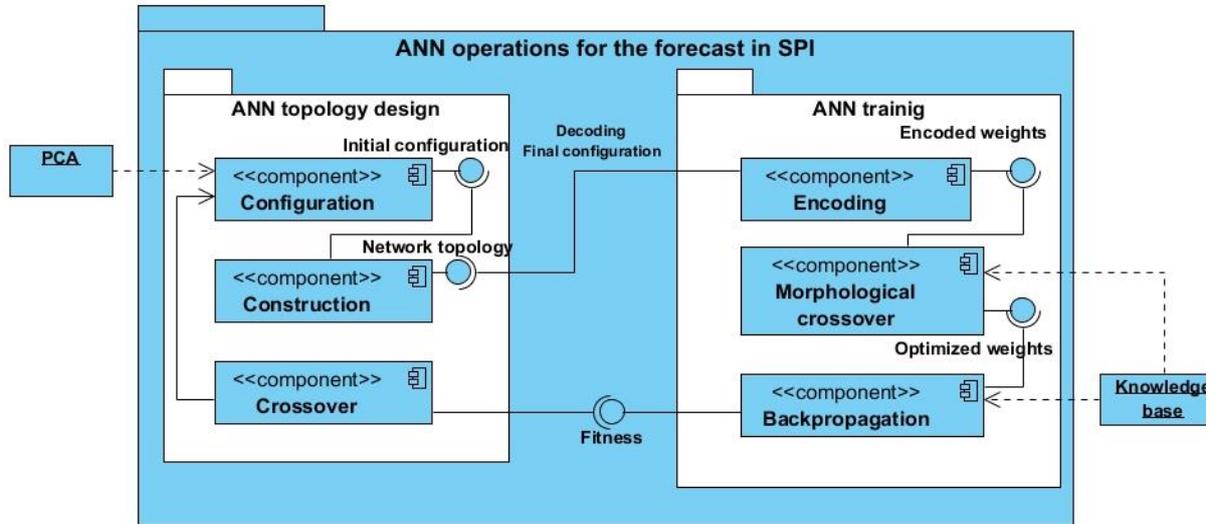The association rules are represented as: X → Y,   Where:

Figure 1: ANN operations for the forecast in SPI

being X and Y sets of elements, where: $X \subset GP$, $Y \subset M$ y $X \cap Y = \emptyset$.

Step 1. Transformation of knowledge in transactions: a search is made in the knowledge base about the measures of Critical Success Factors, which evolved positively from the initial state to the improvement scenario reached, as well as the Good Practices applied by the organization for the change. The recovered information is stored as transactions in a temporary list for further processing. T is a set of transactions where $T = GP \cup M$.

$T = \{GP; M: gp_1, gp_2, ..., gp_n, m_1, m_2, ..., m_m\}$, example: $\{GP_1, GP_2, M_1, M_2, M_3\}$.

Step 2. Calculation of support indexes: being the rule $X \to Y$, where $X \subset GP$ and $Y \subset M$, the support of the rule is calculated as:

$$Sup \ (X \to Y) = \frac{Nt(XY)}{T_t} \qquad (6)$$

Where:
$Sup \ (X \to Y)$ is the support of the rule $X \to Y$.
$N_t(XY)$ represents the number of transactions that contain the elements of X and Y.
T_t represents the total of transactions of T.

Step 3. Identification of frequent elements sets: elements sets with equal or greater support than the established threshold (0,75) are identified.

Step 4. Generation of candidate rules: combinations of candidate rules are generated. The *Apriori* algorithm is applied [17; 18; 19; 37; 38] to reduce the number of candidates, through reducing by pruning. All the rules whose elements are not frequent are discarded, because their combinations will not be.

Step 5. Calculation of confidence indexes: being the rule $X \to Y$, where $X \subset GP$ and $Y \subset M$, the confidence index is calculated from the equation 7.

$$Conf \ (X \to Y) = \frac{N_t(XY)}{N_t(X)} \qquad (7)$$

$Conf \ (X \to Y)$ is the confidence of the rule $X \to Y$.
$N_t(XY)$ represents the number of transactions that contain the elements of X and Y.
$N_t(X)$ represents the number of transactions that contain the elements of X.

Step 6. Obtaining association rules: rules with a confidence index lower than the defined threshold (0,75) are discarded and then, the association rules are generated.

Step 7. Application of association rules: the information of the association rules generated, is provided to the GA. This information is about which Critical Success Factors measurements are favored by which Good Practices.

3.2   ANN for the forecast in SPI

Considering the characteristics of the classification problem, the implementation of an evolutionary ANN based on the execution of GA for its design and learning is required. The ANN operations are represented in figure 1 [35]. The Critical Success Factors measurements are input patterns and they can be dynamic, the output layer responds to the success or failure in SPI.

The design of the network topology is done in the *Configuration* component, where the initial configuration is created to build the network topology in the *Construction* component. Later the decoding of the network and the final configuration are performed. Subsequently, the network training begins.

The *Codification* component encodes the weights of the ANN, which are used in the *Morphological Crossover* component to perform the evolution of the weights for the network topology obtained. These values are used as initial weights in the *Backpropagation* component. Once the data of the knowledge base is obtained, the training of the ANN is realized. The fitness (mean square error of the network) is calculated as a value that allows to determine how effective is the network. Then, the *Crossover* component is executed to obtain a new ANN topology.

The entire process is repeated until an ANN architecture of lower fitness is obtained.

The design and training of the ANN are described in the following steps:

Step 1. Principal Component Analysis (PCA) is applied to reduce the number of input measurements to the ANN. PCA is the problem of fitting a low-dimensional affine subspace to a set of data points in a high-dimensional space. It has become one of the most useful tools for data modeling, compression, and visualization [39]. A binary matrix M of dimension Dim_x x Dim_y is created, which is initialized by 0. The size of the matrix depends on the number of input and output neurons. Dim_x (rows) is the number of input neurons n plus the number of output neurons m (in this case 1), and Dim_y (columns) corresponds to the maximum number of hidden neurons to consider.

In the matrix M, which represents the topology of the ANN with one hidden layer, the meaning for the position (i, j) is defined as follows. Being n the number of input neurons, if $i \leq n$ then (i, j) represents a connection between the input neuron i and the j-th hidden neuron; if $i > n$, (i, j) represents a connection between the j-th hidden neuron and the $(i - n)$-th output neuron. The individuals or chromosomes (seeds of growth and pruning) of the population are generated randomly and in random positions.

Step 2. The growth seeds are located in the matrix according to the values of their genes. The initial configuration of the network is performed, replicating the growth seeds sequentially over its quadratic neighborhood. During replication if a new seed has to be placed in a position previously occupied by another seed, the first one will be replaced.

Step 3. Each chromosome is decoded and converted into network locations, where each seed is represented by two genes (X, Y), corresponding to the coordinates in the matrix.

Step 4. The algorithms of growth and pruning of seeds are applied.

Being a_i, the value in the position (i, j) of the matrix and S the set of growth seeds, $S = \{s_k: s_1, s_2, ..., s_n\}$, a seed s_k is copied, which grows when a position is inactive $(a_{i,j}=0)$ and there are at least three seeds that grow identical in their quadratic neighborhood.

The pruning configuration is performed. The pruning seeds are placed in the positions where a_i, =0. The pruning rule is designed to eliminate the seeds that grow in the network.

Being $a_{i,j}$ the value in the position (i, j) of the matrix and D the set of pruning seeds, $D = \{d_r: d_1, d_2, ..., d_n\}$, a s_k growth seed is extracted, when two contiguous neighboring positions contain identical growth seeds and another neighboring position contains a pruning seed d_r. If two pruning seeds are present in the vicinity, the rule is not activated.

Step 5. The matrix is decoded and the compliance of the necessary restrictions is verified, to obtain the model of the ANN architecture.

- Every seed of growth takes value 1 and pruning seeds acquire value 0. Every 1 in the matrix is interpreted as a connection and 0 as absence of connection.

- Columns with values 0 in the matrix are eliminated. If the elements of the column of order k are 0, there are no connections from the inputs to the hidden neurons k-th and there are no connections from the k-th hidden neuron to the outputs.

- The columns where the value a_i, =0, if $i>n$ (where n is the number of input neurons), are eliminated. If a neuron in the hidden layer has no connection with the output layer, it is eliminated, as it will have no influence on the outputs.

- The rows with values 0 in the matrix are eliminated. When there is a neuron of the input layer without any connection with the hidden layer, it is eliminated, because it will not influence the outputs.

Step 6. The ANN weights are initialized for the defined architecture, in order to obtain a fast convergence in a multilayer perceptron. The weights are encoded by real coding, which allows to explore the domain of the evaluation function (medium square error) with great precision.

Step 7. The ANN is trained through the evolution of connection weights (morphological crossover) to find the best weights configuration.

The selection is made by tournament. They are chosen at random, as many individuals (weights) of the population as has been prefixed in the size of the tournament (given by the number of input neurons). The best individual of the tournament group is selected and the process is repeated until the desired number of individuals to be selected is obtained. Individuals with the best initial weights are considered, to be used by the backpropagation. Subsequently, the morphological crossover is performed, which reinterprets the morphological gradient operation, to obtain a measure of the genetic diversity.

The morphological crossover operates with populations of λ individuals constituted by chains of real numbers with length l. Starting from an odd n number of progenitor chains ($n \leq \lambda$), obtained without repetition of the current population, a set of intervals called crossing intervals (C_i), is obtained. The descendant chains of the operator are generated from the crossing intervals. The following actions are carried out for the morphological crossover:

- Calculation of the measure of genetic diversity, gene to gene from the n individuals taken as parents. Being G the progenitor matrix with dimension $(n \ x \ l)$, for the l columns of G, the one-dimensional vector f_i is defined. f_i contains the n values of the n progenitors for the gene i.

$$G = \begin{bmatrix} a_{10} & a_{11} & \cdots & a_{1l-1} \\ a_{20} & a_{21} & \cdots & a_{2l-1} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n0} & a_{n1} & \cdots & a_{nl-1} \end{bmatrix}$$

$$f_i = (a_{1,i}, a_{2,i}, \ldots, a_{n,1}) \quad with \quad i = 0, \ldots, l-1$$

It is defined as a measure of genetic diversity of gene *i* in the population, the value $g_i \in [0,1]$ calculated as:

$$g_i = g(E(n/2) + 1) = (f_i \oplus b)(E(n/2) + 1) - (f_i \ominus b)(E(n/2) + 1) \tag{8}$$

Where:

g_i is the measure of genetic diversity.

$E(n/2) +1$ is the component located in the middle position of the vector f_i.

f_i is the one-dimensional vector.

n is the number of progenitors for the gene *i*.

$f_i \oplus b$ is the dilation of vector f_i on the point $E(n/2) +1$, with the structuring element *b*. The result is the maximum value of the components of the vector, because the structuring element *b* iterates through f_i from the component E (n/2) + 1−E (n/2) = 1, to E (n/2) + 1 + E (n/2) = 2 E (n/2) + 1 = n (where *n* is odd).

$f_i \ominus b$ is the erosion of vector f_i on the point $E(n/2) +1$, with the structuring element *b*. It is obtained in the same way as dilation, but calculating the minimum value of the components of vector f_i.

- The crossing intervals are calculated, determining the lower and upper bounds of the crossing interval C_i denoted by $C = \{C_0, \ldots, C_{l-1}\}$. The maximum gene is calculated from the equation 9 and the minimum gene by the equation 10. The crossing intervals $C_i = [g_i min, g_i max]$.

$$g_i max = max(f_i) - \phi(g_i) \tag{9}$$

Where:

$g_i max$ is the maximum gene of the crossing interval C_i.

$max(f_i)$ is the dilation of the vector f_i at the midpoint of C_i.

$\phi(g_i)$ is the value of the exploration / exploitation function at the point g_i.

$$g_i min = min(f_i) + \phi(g_i) \tag{10}$$

Where:

$g_i min$ is the minimum gene of the crossing interval C_i.

$min(f_i)$ is the erosion of the vector f_i at the midpoint of C_i.

$\phi(g_i)$ is the value of the exploration / exploitation function at the point g_i.

- Obtaining descendants. It is the final result of the morphological crossover operator. The descendants are determined by:

$o = (o_0, \ldots, o_{l-1})$ and $o' = (o'_0, \ldots, o'_{l-1})$.

o_i is a random value of the crossing interval C_i

o'_i is obtained by the equation 11:

$$o'_i = \left(min(f_i) + max(f_i) \right) - o_i \tag{11}$$

Where:

$i = 0, 1, \ldots, l-1$.

$min(f_i)$ is the erosion of the vector f_i at the midpoint of C_i.

$max(f_i)$ is the dilation of the vector f_i at the midpoint of C_i.

- Then, the worst individuals of the starting population are replaced with the new descendants, taking as an evaluation function the mean square error.

- Subsequently, the selection by tournament is made again, obtaining a new progenitor matrix. This procedure is carried out in several iterations until obtaining the values of the connection weights, which minimize the mean square error for the configuration of the network in question.

The application of GA for the evolution of weights is not very efficient in local searches, but it is effective in global search. Therefore, training can be improved with the incorporation of the local search method, backpropagation. It is very appropriate to perform a combination where the GA searches for a suitable region in the search space and then the backpropagation refines the solution found, obtaining a result closer to the optimum in said region.

Step 8. The training of the RNA is refined using the weights optimized for the architecture obtained, through backpropagation. The fitness defined by the mean square error is obtained to determine the network efficiency. This step is carried out in several iterations until obtaining the values of refined connection weights, which minimize the mean square error.

Step 9. The ANN resulting from the previous step is encoded.

Step 10. From the set of chromosomes used in the *Configuration* component, the genes that compose the chromosomes are crossed and new populations of topologies are obtained, which will be used in next iterations in *Configuration* component.

Step 11. The steps from 3 to 10, are executed through different iterations, until obtaining in the intermediate step between *Backpropagation* and *Crossover*, an ANN with fitness lower than the established threshold (0,05). This ANN will be used for the forecast of result in the SPI of an initial state or improvement scenario.

In this way, KAIRÓS automates the processing in combination of the Critical Success Factors and Good Practices, to support the decision-making in the SPI. It implements artificial intelligence techniques for the scenarios optimization, the proposal of recommendations, the forecast of the state of organizations to face an SPI project and the generation of association rules between Good Practices and Critical Success Factors measurements.

4 Solution validation

To assess the effect of implementation on decision-making support, a quasi-experiment of multiple chronological series was developed with two pre-tests, two post-tests and a control group in 12 software development centers of the University of Informatics Sciences, with degrees of manipulation (with and without stimulus).

- Pre-test 1: the initial diagnosis was applied and it was identified with KAIRÓS, that the forecast of the initial state was failure for four centers in both groups. The forecast of the minimum improvement scenario was successful in four centers for both groups. No significant differences were identified.

- Pre-test 2: five days after the application of the diagnosis, the improvement plans of the centers were analyzed. The objective of this test was to evaluate the ratio between the improvement actions associated with the Good Practices and the recommendations proposed by KAIRÓS for the minimum scenario. There were no significant differences (significance level of 0,065).

- Post-test 1: the processing results of KAIRÓS were presented to the experimental group. After 15 days, the recommendations proposed by the system in the Improvement Plan of the experimental group had been incorporated, which didn't happen in the control group. Significant differences were identified between the groups (significance level of 0,003).

- Post-test 2: after the application of the stimulus, it was observed that the ratio between improved Critical Success Factors measurements and measurements that should be intentional according to KAIRÓS, ranged between 0,14 y 0,31 in control group, and in the experimental group between 0,87 and 1,00. In the control group, four successful minimum scenarios were predicted and after two months, only the center with initial status predicted as success could maintain this condition. In the experimental group, four successful minimum scenarios were predicted and after two months, successful states were reached by these centers. Significant differences were identified between the groups (significance level of 0,004).

To assess the applicability and satisfaction, six quality consultants and seven managers of software development centers were surveyed. The variables evaluated were customer satisfaction, applicability and utility through the use of Iadov. A group satisfaction index of 0,92 was obtained. There was a concordance of 84,62% with "Excellent" qualification for the utility and a 92,31% with "Excellent" qualification for the applicability in real environments. About its contribution to the decision-making at the beginning of SPI, there was a concordance of 92,31% with "Excellent" qualification. The rest of qualifications was "Good".

5 Conclusions

Based on the results obtained, it is considered that experiences reuse for the scenarios recommendation and the forecast before the investment in SPI, favor the decision-making in SPI. For the analysis of the information associated with Good Practices and Critical Success Factors combined, it is necessary to lean on artificial intelligence techniques, which facilitate the information processing for decision-making support in SPI.

KAIRÓS intelligent system, automates the processing of Critical Success Factors and Good Practices combined, through the integration of artificial intelligence techniques. The implementation of a GA favors the optimization towards better scenarios in SPI. The association rules allow to identify dependencies between Good Practices and Critical Success Factors measurements. The use of an evolutionary ANN, helps to predict the results of organizations in SPI.

The validation results of the solution corroborate that its application contributes to support decision-making at the beginning of SPI, through the combination treatment of Critical Success Factors and Good Practices. A high satisfaction with the solution is evidenced, in the positive criteria about the contribution of the system and in the evaluation of its implementation effect.

3 References

[1]   DOUNOS, P. and G. BOHORIS (2010). Factors for the design of CMMI-based software process improvement initiatives. Conference on Informatics (PCI), 2010 14th Panhellenic. Tripoli IEEE Xplorer Digital Library: 43-47. 1424478383. https://doi.org/10.1109/pci.2010.46

[2]   MONTONI, M. A. and A. R. ROCHA (2010). Applying grounded theory to understand software process improvement implementation. Conference on the 2010 Seventh International Quality of Information and Communications Technology (QUATIC). IEEE Computer Society: 25-34. 1424485398. https://doi.org/10.1109/quatic.2010.20

[3]   NIAZI, M.; M. A. BABAR and J. M. VERNER (2010). Software Process Improvement barriers: A cross-cultural comparison. Information and software technology, 52(11): 1204-1216. https://doi.org/10.1016/j.infsof.2010.06.005

[4]   NIAZI, M.; D. WILSON and D. ZOWGHI (2006). Critical success factors for software process improvement implementation: an empirical study. Software Process: Improvement and Practice, 11(2): 193-211. https://doi.org/10.1002/spip.261

[5]   TRUJILLO-CASAÑOLA, Y.; A. FEBLES-ESTRADA and G. LEÓN-RODRÍGUEZ (2014). Modelo para valorar las organizaciones al iniciar la mejora de procesos de software. Ingeniare. Revista chilena de ingeniería, 22(3): 412-420. https://doi.org/10.4067/s0718-33052014000300011

[6]   FERNÁNDEZ DÍAZ, H.; N. MILÁN CRISTO; A. M. GARCIA RODRÍGUEZ and Y. TRUJILLO CASAÑOLA. (2016). Bases teóricas para un procedimiento que evalúe cuantitativamente la influencia de los Factores Críticos de Éxito en la Mejora de Procesos. Informática 2016. VII Taller Internacional de Calidad en las Tecnologías de la Información y las Comunicaciones. La Habana, XVI Convención y Feria Internacional INFORMÁTICA 2016.

[7]   CLARKE, P. and R. O'CONNOR (2010). Harnessing ISO/IEC 12207 to Examine the Extent of SPI Activity in an Organisation. European

Conference on Software Process Improvement. Springer: 25-36. https://doi.org/10.1007/978-3-642-15666-3_3

[8] NIAZI, M.; D. WILSON and D. ZOWGHI (2005). A maturity model for the implementation of software process improvement: an empirical study. Journal of systems and software, 74(2): 155-172. https://doi.org/10.1016/j.jss.2003.10.017

[9] NIAZI, M.; D. WILSON and D. ZOWGHI (2005). A framework for assisting the design of effective software process improvement implementation strategies. Journal of systems and software, 78(2): 204-222. https://doi.org/10.1016/j.jss.2004.09.001

[10] GOLDBERG, D. E. (1989). Genetic Alogorithms in Search. Optimization & Machine Learning. Choice Reviews Online, 27(2): 27–0936 – 0927-0936. https://doi.org/10.5860/choice.27-0936

[11] PAVEZ-LAZO, B.; J. SOTO-CARTES; C. URRUTIA and M. CURILEM (2009). Selección determinística y cruce anular en algoritmos genéticos: aplicación a la planificación de unidades térmicas de generación. Ingeniare. Revista chilena de ingeniería, 17(2): 175-181. https://doi.org/10.4067/s0718-33052009000200006

[12] MARTÍN, D.; A. ROSETE; J. ALCALÁ-FDEZ and F. HERRERA (2014). QAR-CIP-NSGA-II: A new multi-objective evolutionary algorithm to mine quantitative association rules. Information Sciences, 258: 1-28. https://doi.org/10.1016/j.ins.2013.09.009

[13] OVIEDO CARRASCAL, E. A.; A. I. OVIEDO CARRASCAL and G. L. VÉLEZ SALDARRIAGA (2015). Minería de datos: aportes y tendencias en el servicio de salud de ciudades inteligentes. Revista Politécnica, 11(20): 111-120.

[14] TALLÓN-BALLESTEROS, A. J. (2014). New training approaches for classification based on evolutionary neural networks. Application to product and sigmoidal units. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, 17(54).
https://doi.org/10.4114/intartif.vol17iss54pp30-34

[15] TALLÓN-BALLESTEROS, A. J.; C. HERVÁS-MARTÍNEZ; J. C. RIQUELME and R. RUIZ (2013). Feature selection to enhance a two-stage evolutionary algorithm in product unit neural networks for complex classification problems. Neurocomputing, 114: 107-117. https://doi.org/10.1016/j.neucom.2012.08.041

[16] TOSTADO SÁNCHEZ, S. E.; M. ORNELAS RODRÍGUEZ; A. ESPINAL JIMÉNEZ and H. J. PUGA SOBERANES (2016). Implementación de Algoritmos de Inteligencia Artificial para el Entrenamiento de Redes Neuronales de Segunda Generación. JÓVENES EN LA CIENCIA, 2(1): 6-10.

[17] YANG, G.; H. ZHAO; L. WANG and Y. LIU (2009). An implementation of improved apriori algorithm. Conference on 2009 International Machine Learning and Cybernetics. IEEE: 1565-1569. 1424437024. https://doi.org/10.1109/icmlc.2009.5212246

[18] SINGH, J.; H. RAM and D. J. SODHI (2013). Improving efficiency of apriori algorithm using transaction reduction. International Journal of Scientific and Research Publications, 3(1): 1-4.

[19] YABING, J. (2013). Research of an improved apriori algorithm in data mining association rules. International Journal of Computer and Communication Engineering, 2(1): 25-27. https://doi.org/10.7763/ijcce.2013.v2.128

[20] BADDOO, N. and T. HALL (2002). Motivators of Software Process Improvement: an analysis of practitioners' views. Journal of systems and software, 62(2): 85-96. https://doi.org/10.1016/s0164-1212(01)00125-x

[21] BLANCO, K. R.; A. S. BATISTA; D. P. MONTALVÁN; D. N. AGÜERO; A. F. ESTRADA; R. D. MARTÍNEZ and M. M. ROJA (2011). Experiencias del programa de mejora de procesos en la Universidad de las Ciencias Informáticas. Revista Cubana de Ciencias Informáticas, 5(2).

[22] CAPOTE, J.; C. J. LLANTÉN; C. PARDO; A. J. GONZÁLEZ and C. A. COLLAZOS (2008). Gestión del conocimiento como apoyo para la mejora de procesos software en las micro, pequeñas y medianas empresas. Ingenieria e investigacion, 28(1): 137-145.

[23] CONRADI, R.; T. DYBÅ; D. I. SJØBERG and T. ULSUND (2003). Lessons learned and recommendations from two large norwegian SPI programmes. European Workshop on Software Process Technology. Springer: 32-45. https://doi.org/10.1007/978-3-540-45189-1_4

[24] DEL VILLAR, B. L. D. and M. A. M. MATA (2016). Selección de estrategias para la implementación de Mejoras de Procesos Software. ReCIBE, 2(3).

[25] DYBA, T. (2000). An instrument for measuring the key factors of success in software process improvement. Empirical software engineering, 5(4): 357-390.

[26] GONZALO, C.; M. JEZREEL; M. MIRNA and S. F. TOMÁS (2010). Experiencia en la mejora de procesos de gestión de proyectos utilizando un entorno de referencia multimodelo. RISTI-Revista Ibérica de Sistemas e Tecnologias de Informação(6): 87-100.

[27] JALOTE, P. (2002). Lessons learned in framework-based software process improvement. Software Engineering Conference, 2002. Ninth Asia-Pacific. IEEE: 261-265. 0769518508. https://doi.org/10.1109/apsec.2002.1182995

[28] MAS, A. and E. AMENGUAL (2005). La mejora de los procesos de software en las pequeñas y medianas empresas (pyme). Un nuevo modelo y su aplicación a un caso real. Revista Española de Innovación, Calidad e Ingeniería del Software, 1(2): 7-29.

[29] PANTOJA, W. L.; C. A. COLLAZOS and V. M. R. PENICHET (2013). Entorno Colaborativo de Apoyo a la Mejora de Procesos de software en Pequeñas Organizaciones de Software. Dyna, 80(177): 40-48.

[30] PINO, F.; F. GARCÍA and M. PIATTINI (2006). Revisión sistemática de mejora de procesos software en micro, pequeñas y medianas empresas. Revista Española de Innovación, Calidad e Ingeniería del Software, 2(1): 6-23.

[31] SANTOS, G.; M. MONTONI; J. VASCONCELLOS; S. FIGUEIREDO; R. CABRAL; C. CERDEIRAL; A. E. KATSURAYAMA; P. LUPO; D. ZANETTI and A. R. ROCHA (2007). Implementing software process improvement initiatives in small and medium-size enterprises in Brazil. Conference on the 6th International Quality of Information and Communications Technology, 2007. QUATIC 2007. IEEE: 187-198. 0769529488. https://doi.org/10.1109/quatic.2007.22

[32] YÉPEZ VARGAS, W.; C. PRIMERA LEAL and M. TORRES SAMUEL (2013). Mejoras al proceso de planificación de proyectos de software usando el Modelo de Madurez de Capacidad Integrado (CMMI). Compendium, 16(30).

[33] GARCIA RODRÍGUEZ, A. M.; Y. MILANÉS ZAMORA; Y. TRUJILLO CASAÑOLA; J. P. FEBLES RODRÍGUEZ and I. J. SÁNCHEZ GONZÁLEZ (2018). Asociación entre Buenas Prácticas y Factores Críticos para el éxito en la MPS. Revista Cubana de Ciencias Informáticas, 12(2): 89-103.

[34] GARCÍA RODRÍGUEZ, A. M.; Y. TRUJILLO CASAÑOLA and A. PERDOMO VERGARA (2016). Optimización de estados en la mejora de procesos de software. Enl@ ce: Revista Venezolana de Información, Tecnología y Conocimiento, 13(2): 9-27.

[35] GARCIA RODRÍGUEZ, A. M.; Y. TRUJILLO CASAÑOLA and L. ARZA PÉREZ (2016). Pronóstico de éxito en la Mejora de Procesos de Software. Revista Cubana de Ciencias Informáticas, 10: 15-30.

[36] TORRES, M.; K. PAZ and F. SALAZAR. (2006). Tamaño de una muestra para una investigación de mercado, Universidad Rafael Landívar.

[37] PRADHAN, T.; S. R. MISHRA and V. K. JAIN (2014). An effective way to achieve excellence in research based learning using association rules. Conference on 2014 International Data Mining and Intelligent Computing (ICDMIC). IEEE: 1-4. 1479946745. https://doi.org/10.1109/icdmic.2014.6954226

[38] LIN, X. (2014). Mr-apriori: Association rules algorithm based on mapreduce. Conference on 2014 5th IEEE International Software Engineering and Service Science (ICSESS). IEEE: 141-144. 1479932795. https://doi.org/10.1109/icsess.2014.6933531

[39] VIDAL, R.; Y. MA and S. S. SASTRY. (2016). Principal component analysis. en: Generalized Principal Component Analysis. Springer: 25-62.