World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:3, 2016

Discriminant Analysis as a Function of Predictive Learning to Select Evolutionary Algorithms in Intelligent Transportation System

Jorge A. Ruiz-Vanoye, Ocotlán Díaz-Parra, Alejandro Fuentes-Penna, Daniel Vélez-Díaz, Edith Olaco García

***Abstract*—**In this paper, we present the use of the discriminant analysis to select evolutionary algorithms that better solve instances of the vehicle routing problem with time windows. We use indicators as independent variables to obtain the classification criteria, and the best algorithm from the generic genetic algorithm (GA), random search (RS), steady-state genetic algorithm (SSGA), and sexual genetic algorithm (SXGA) as the dependent variable for the classification. The discriminant classification was trained with classic instances of the vehicle routing problem with time windows obtained from the Solomon benchmark. We obtained a classification of the discriminant analysis of 66.7%.

***Keywords*—**Intelligent transportation systems, data-mining techniques, evolutionary algorithms, discriminant analysis, machine learning.

I. Introduction

THE discriminant analysis [1], [2] is a multivariate statistical technique whose purpose is to analyse if significant differences between groups of objects with respect to a set of variables measured on such explaining exist in what sense procedures of systematic classification of new observations of origin not known in one of the analysed groups occur and provide. The Linear Discriminant Analysis (LDA) was introduced by Fisher [3] as a statistical procedure for the classification. LDA is concerned with classification problems where the dependent variable is categorical (nominal or ordinal) and the independent variables are metric. The objective of LDA is to construct a discriminant function that yields different scores when computed with data from different output classes. The classification is the most common task inside of generic inductive learning; in addition, it is a function of predictive learning that inside classifies a data of diverse classes [4].

In this paper, we propose to use the classification as a function of predictive learning in evolutionary algorithms. An evolutionary algorithm [5] consists:

1. Evolutionary strategies (ESs) were created and developed by Rechenberg [6] and his co-workers. ESs use real-coding of design parameters since they model the organic evolution at the level of individual's phenotypes. ESs depends on deterministic selection and mutation for its evolution. ESs use strategic parameters such as on-line self-adaptation of mutability parameters.
2. Evolutionary programming is one of the four major evolutionary algorithm paradigms. It was first used by Fogel [7] in order to use simulated evolution as a learning process aiming to generate artificial intelligence. Fogel used finite state machines as predictors and evolved them. Currently, evolutionary programming is a wide evolutionary computing dialect with no fixed structure or (representation), in contrast with some of the other dialects. It is becoming harder to distinguish from ESs. Some of its original variants are quite similar to the later genetic programming, except that the program structure is fixed and its numerical parameters are allowed to evolve. Its main variation operator is mutation; members of the population are viewed as part of a specific species rather than members of the same species therefore each parent generates an offspring, using a survivor selection.
3. GA is a search technique used in computing to find exact or approximate solutions to optimization and search problems. GAs are categorized as global search heuristics. GAs are a particular class of evolutionary algorithms (EAs) that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination).
4. Genetic Programming (GP) is an EA-based methodology inspired by biological evolution to find computer programs that perform a user-defined task. It is a specialization of GA where each individual is a computer program. Therefore, it is a machine learning technique used to optimize a population of computer programs according to a fitness landscape determined by a program's ability to perform a given computational task. The first statement of "tree-based" GP (that is, procedural languages organized in tree-based structures and operated on by suitably defined GA-operators) was given by Cramer [8]. This work was later greatly expanded by Koza [9], a main proponent of a GP who has pioneered the application of GP in various complex optimization and search problems.

In this paper, our interest is the GAs. Gas, originally developed by Holland [10], are heuristic adaptive that simulate processes of optimization with the natural evolution of genes in an organism population (vertical gene transfer or VGT).

Jorge A. Ruiz-Vanoye, Jr is with the Universidad Autónoma del Estado de Hidalgo, México, CO 42780 MX (phone: 01-771-717-2000 ext5509; e-mail: jorge@ruizvanoye.com).

Ocotlán Díaz-Parra, Alejandro Fuentes-Penna, Daniel Velez Díaz, and Edith Olaco Garcia are with the Universidad Autónoma del Estado de Hidalgo, México, CO 42780 MX (e-mail: ocotlan@diazparra.net, alexfp10@hotmail.com, dvelez@uaeh.edu.mx, Edith.olaco@hotmail.com).

The VGT is the transfer of genetic material to offspring, or the inheritance of genes by subsequent generations, is an essential basis of the evolutionary process. The most common form of gene transfer for higher organisms is sexual reproduction. In the case of higher plants, genetic information is passed along to the next generation by pollination. This is called a vertical gene transfer. The VGT occurs when an organism receives genetic material from its ancestor, e.g. its parent or a species from which it evolved [11].

GA initializes randomly population of problem solutions. The GA evaluates each one of the solutions (assign him score or fitness according to solution feasible that finds). The GA chooses of the population (the one that has a greater score). The GA applies of crossover operator and mutation operators for different solutions from the selected population, to create a new population.

The theory of the computational complexity is the part of the theory of the computation that studies the resources required during the calculation to solve a problem [12]. The resources commonly studied are the time (execution number of an algorithm to solve a problem) and the space (amount of resources to solve a problem).

A combinatorial optimization problem is either a minimization problem or a maximization problem and consists of three parts: a) a set of instances, b) candidate solutions for each instance, c) a solution value [13]. The combinatorial optimization problem that was used in this paper is the Vehicle Routing Problem with Time Windows or VRPTW (NP-hard problem) used in an intelligent system for transport.

VRP is a combinatorial optimization problem complex [13], [14]. It is considered naturally like a central problem in the areas of transportation, logistics and distribution. In some sectors of the industry, the transportation means a high percentage of value added to products. For that reason, the use of computational methods on the transportation offers good results, the savings go from a 5% to 20% in the total of costs, as Toth & Vigo report [14]. The VRP has diverse variants of the problem, one of them is the VRPTW (Vehicle Routing Problem with Time Windows), consists basically of diminishing the costs of subject transportation to restrictions of time of each route and capacity on the basis of the demand of each client [14]. Equations (1)-(11) represent the VRPTW Model:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} \tag{1}$$

$$\sum_{k \in K} \sum_{j \in \Delta^{+}(i)} x_{ijk} = 1; \qquad \forall i \in N \tag{2}$$

$$\sum_{j \in \Delta^{+}(0)} x_{0jk} = 1; \qquad \forall k \in K \tag{3}$$

$$\sum_{i \in \Delta^{-}(j)} x_{ijk} - \sum_{i \in \Delta^{+}(j)} x_{ijk} = 0 \qquad \forall k \in K, j \in N \tag{4}$$

$$\sum_{i \in \Delta^{-}(n+1)} x_{i,n+1,k} = 1 \qquad \forall k \in K \tag{5}$$

$$w_{ik} + s_i + t_{ij} - w_{jk} \le (1 - x_{ijk}) M_{ij} \qquad \forall k \in K, (i,j) \in A \tag{6}$$

$$a_i \sum_{j \in \Delta^{+}(i)} x_{ijk} \le w_{ik} \le b_i \sum_{j \in \Delta^{+}(i)} x_{ijk} \qquad \forall k \in K, i \in N \tag{7}$$

$$E \le w_{ik} \le L \qquad \forall k \in K, i \in (0, n+1) \tag{8}$$

$$\sum_{i \in N} d_i \sum_{j \in \Delta^{+}(i)} x_{ijk} \le C \qquad \forall k \in K \tag{9}$$

$$x_{ijk} \ge 0 \qquad \forall k \in K, (i,j) \in A \tag{10}$$

$$x_{ijk} \in \{0,1\} \qquad \forall k \in K, (i,j) \in A \tag{11}$$

where: 0 = The zero represent the deposit at the beginning of the route, A = the set formed by ordered pairs (i, j), a = the time window limit in a node, b = the time window limit in a node, C = Capacity of vehicle, c = cost of service, d = demand, E = beginning time window in the deposit, i = origin node, j = destiny node, K = the fleet of vehicles, k = vehicle, L = the time window limit in the deposit, M = a very big positive number, N = the set of nodes, n+1 = the deposit when the route has already been realized, S = Service time, t = the time of arrival to the following node, W = the beginning of the service time in the node, X = the execution of an operation, $i \in \Delta^{-}(j)$ = an origin i together with a destiny j with direction of j directly to i, $j \in \Delta^{+}(i)$ = a destiny j together with an origin i with direction of i to j. Solomon [15] mentions that in the problem Vehicle Routing Problem with Time Windows (VRPTW) exists classifications of type C for instances clustered, type RC for instances Random and Clustered, Type R for Random instances and in addition the instance is determining by the number of clients or CN.

An individual is the set of genes grouped in chromosomes; an individual in VRPTW (Fig. 1) is a formed route of sub-routes (chromosomes), each sub-route is formed by nodes (gene).

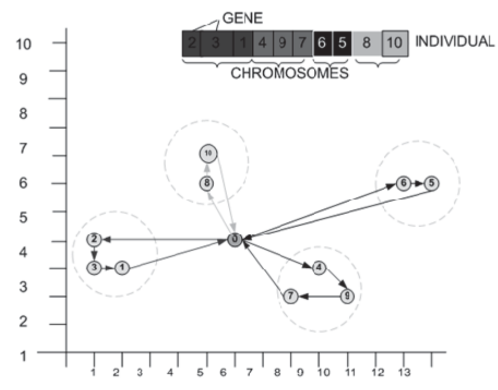

Fig. 1 Individual in VRPTW

In this paper, we propose the use discriminant analysis as a function of predictive learning to select as it is the best algorithm than it solves instances of an Intelligent Transportation Systems. At the moment the computer science systems exist to provide solutions to some problems of transportation, so is the case of the calls Intelligent

Transportation Systems or ITS [16], by means ITS is constructed to an itinerary of routes solution. ITS serves as support for the citizen and public institutions in the attempt to search for the industry of the transport the minimization of costs for their sustainability. Nevertheless, the ITS is not easily adaptable to the specific circumstances of each company, since each company of transport has different needs for transportation. By such reason the efforts realized by the scientific community to find specific solutions to real problems difficult to solve as it is the Vehicle Routing Problem with time windows.

In the experimentation to verify the hypothesis of this investigation we used the discriminant analysis as a function of predictive learning to select EAs (in an intelligent transportation system) that better solves instances of transport problem (or the vehicle routing problem with time windows) that means for Best Route Choice.

II. Related works

In the area of the data mining, Fink [17] developed a technique of algorithm selection for decision problems, which is based on the estimation of the gain of the algorithm, obtained by the statistical analysis of its performances. The investigation group that develops to the project METAL, proposed a method to select the algorithm for a set of related cases. They identify the set of old cases that she shows characteristics more similar to those of the new set. The algorithmic performance of the old cases is known and used to predict the best algorithm for the new set of cases.

Cruz [18] proposed a methodology, based on automatic learning systematically to develop mathematical models of the algorithmic performance. The proposal of Cruz consists of characterizing the performance of a set of heuristic algorithms applied to the solution of NP-hard problems, by means of the formation of algorithms domination regions.

The work of Soares & Brazdil [19] has made important advances in the use of the characterization of the cases, to integrate groups of similar cases. But, for a different case that it does not belong to a predefined class of cases, the prediction of algorithm performance will not be able to be made. The GAs to solve problems VRPTW were used pioneering by Blanton & Wainwright [20] and Thangiah [21]. Table I shows the related works more important for this investigation.

TABLE I
Related Works

Research	*Modelled of characteristics of a problem*	*Discriminant Analysis*	*Prediction*	*Selection*	*Evolutionary Algorithms*	*VRPTW*
Blanton						√
Thangiah						√
Tshang			√	√		
Hoos			√			
Fink			√	√		
Soares	√		√	√		
Cruz	√		√	√		
This	√	√	√	√	√	√

III. Discriminant Analysis as a Function of Predictive Learning to Select EAs in Intelligent Transportation System

The methodology shown in Fig. 2 uses to verify the hypothesis of this investigation, the methodology consists of: 1) sampling of VRPTW instances, 2) measurement of indicators, and 3) experimentation (discriminant analysis, algorithm results, and best algorithm).

In the phase of the sampling, we used the instances of Solomon [15] for 25 and 100 nodes. The instance parameters of the VRPTW are in Table II. Where: *VN* = Vehicle Number, *C* = Capacity of the Vehicle, *CN* = Customer Number, *XCO* = X Coord., *YCO* = Y Coord., *D* = Demand of client, *RT* = Ready time, *DT* = Due date, *ST* = Service Time.

In the problem exist additional parameters: the value 1 for small time window and small vehicle capacity, 2 big time window and big vehicle capacity, in addition to *C* = clustered data, *R* = random data, *RC* = random and clustered data.

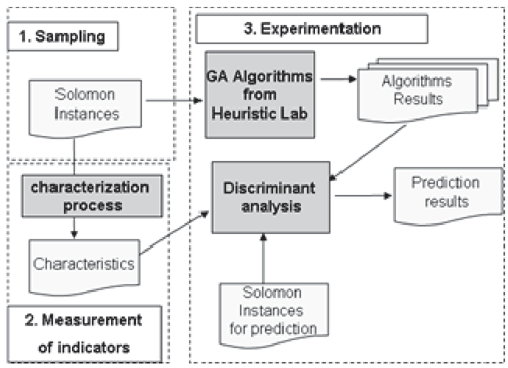

Fig. 2 Methodology used in the investigation

TABLE II
Format of VRPTW Instances

VN	C					
CN	XCO	YCO	D	RT	DT	ST
0	x_0	y_0	d_0	RT_0	DT_0	ST_0
…	…	…	…	…	…	…
100	X_{100}	y_{100}	d_{100}	RT_{100}	DT_{100}	ST_{100}

The indicators are mathematical formulations with which it looks for to reflect a certain situation. The indicators are useful for several aims: management evaluating, to identify improvement opportunities, adapting the objectives and strategies to the reality, to make preventive measures on time, to communicate ideas, thoughts and values of a summarized way. The characteristics of the indicators, serves an intention, has been designed considering this intention and the user characteristics or problem; it is useful and non-subjective (it allows to obtain the same result when the obtaining of the indicator is made by different observers in analogous circumstances), it is specific (applicable only to the situation in question), it is valid (it measures what it is tried to measure), it is consisting of the course of the time, it is precise, it is transparent (easily understood and interpreted by the users). Exist diverse indicators of other areas as they can be macroeconomic indicators, population indicators, and statistical indicators, to mention some. An indicator is a relation between quantitative variables or qualitative that allows observing the situation and the tendencies of changes generated in the object or observed phenomenon, in relation to objectives, and impacts.

The objective of the indicators proposed for VRPTW is to allow being base appropriately to select by means of machine learning an algorithm that better solves an instance of the VRPTW and that is representative, trustworthy and excellent.

The *IAREA* indicator (12) is based on a position method of the known descriptive statistic like median is applied to three instances parameters of the problem *Xcoord*, *Ycoord*, and *Demand*. Where: *IAREA* = the proposed indicator 1, *X* = X Coord., *Y* = Y Coord., *D* = Demand, *CN* = Customer Number.

$$IAREA=((\mathrm{MEDIAN}(X_0{:}X_{CN})+\mathrm{MEDIAN}(Y_0{:}Y_{CN})+\mathrm{MEDIAN}(D_0{:}D_{CN}))*CN)/100 \tag{12}$$

The justification to unite parameters of VRPTW with different units was proposed by Michie et al. [22]: to transform the attributes usually by applying a monotonic transformation of the power law type. Monotonic transformations do not affect the machine learning methods.

We propose the *ISIZE* indicator (13) as size as the instance of the problem that was previously only based on the value of customer number (*CN*). Where: *ISIZE* = the proposed indicator 2, *R* = at the restrictions of the data on the basis of the value 1 for big time window and small vehicle capacity, 2 big time window and big vehicle capacity, *A* = at the randomness of the data with *A* = 1 for data type C, *A* = 2 for data type RC, *A* = 3 for data R type:

$$ISIZE= CN*R*A \tag{13}$$

The *ITIME* Indicator (14) applies to the three time parameters (*RT*, *DT*, *ST*) of the problem instance. Where *ITIME* = the proposed indicator 3, *RT* = Ready time, *DT* = due date, *ST* = service time:

$$ITIME=((\mathrm{MEDIAN}(RT_0{:}RT_n)+\mathrm{MEDIAN}(DT_0{:}DT_n)+\mathrm{MEDIAN}(ST_0{:}ST_n))*CN)/100 \tag{14}$$

IV. Experimentation and Results

In the experimentation was used a computer Acer TravelMate 2330LC Intel Celeron, 1.5 GHz, 512 MB of RAM memory, 80 GB hard disk, 15.0" XGA. We used the algorithms: the generic GA, RS, SSGA, SXGA [20]-[24].

A simple GA works as follows [5]:

1. Start with a randomly generated population of *n* l-bit chromosomes (candidate solutions to a problem).
2. Calculate the fitness *f* (*x*) of each chromosome *x* in the population.
3. Repeat the following steps until *n* offspring have been created:

a. Select a pair of parent chromosomes from the current population, the probability of selection being an increasing function of fitness. Selection is done "with replacement," meaning that the same chromosome can be selected more than once to become a parent.

b. With probability cp (the "crossover probability" or "crossover rate"), crossover the pair at a randomly chosen point (chosen with uniform probability) to form two offspring. If no crossover takes place, form two offspring that are exact copies of their respective parents.

c. Mutate the two offspring at each locus with probability pm (the mutation probability or mutation rate), and place the resulting chromosomes in the new population. If n is odd, one new population member can be discarded at random.

4. Replace the current population with the new population.
5. Go to step 2.

Each iteration of this process is called a generation. A GA is typically iterated for anywhere from 50 to 500 or more generations. The entire set of generations is called a run. At the end of a run there are often one or more highly fit chromosomes in the population. Since randomness plays a large role in each run, two runs with different random-number seeds will generally produce different detailed behaviours. GA researchers often report statistics (such as the best fitness found in a run and the generation at which the individual with that better fitness was discovered) averaged over many different runs of the GA on the same problem. A most often requires a fitness function that assigns a score (fitness) to each chromosome in the current population, the fitness of a chromosome depends on how well that chromosome solves the problem at hand.

SSGA [23] is that replace a large proportion of the population are called generational and those replacing a single solution or only a few. A pseudocode for a typical steady-state algorithm is shown.

1. Calculate the Population, Population (M).
2. While the stopping criterion is not satisfied do
2.a. P1, P2← ParentsSelection(Population)
2.b. O1←Crossover (P1,P1)
2.c. O2←Mutation(O1)
2.d. R←SolutionOutSelection(Population)
2.e. Replace(O2,R)
3. End while

The function Population (M) generates M random solutions.

The two selection methods need to be more specified. Many selection methods are available for choosing both individuals to reproduce and also for surviving at the end of every iteration. The same parents can be chosen several times to reproduce. The selection methods use fitness values associated with each solution to compare the solutions. This is a steady-state algorithm; a crossover can be applied in every generation because a large part of the population will always be preserved in the next generation. Other operators can also be applied after or instead of Mutation. The Replace function replaces individual R in the population with the offspring O2 in order to keep the size of the population constant. Of course, it is not wise to replace the best individual in the population.

SSGA consists of the selection of parent chromosomes for reproduction, in the case of GA, is done using only one selection strategy. When considering the model of sexual selection in the area of population genetics it gets obvious that the process of choosing mating partners in natural populations is different for male and female individuals. Inspired by the idea of male vigor and female choice, Lis and Eiben [25] have proposed Sexual GA that utilizes two different selection strategies for the selection of two parents required for the crossover. The first type of selection scheme utilizes random selection and another selection strategy uses roulette wheel selection for the selection of two parents. The rest of the process is similar to that of GA.

The main idea of the RS algorithm is to generate an initial solution with moderate quality. Then, according to some predefined neighbourhood, the algorithm probabilistically selects and tests whether a nearby solution in the search space is better or not. If the new solution is better, the algorithm adopts it and starts searching in the new neighbourhood; otherwise, the algorithm selects another solution point. The algorithm stops after a specified number of search steps have elapsed or the solution does not improve after a fixed number of steps. The solution quality of a neighbourhood search technique relies heavily on the construction of the solution neighbourhood. The pseudo-code for the RS algorithm is the following:

1. Initialize.
2. Evaluate.
3. Save as best solution.
4. Repeat the following for a number of iterations or rounds.
5. Create random solution.
6. Evaluate.
7. Save if the solution is better.
8. End.

In GA algorithm we used the values: *generations* = 1000, *population size* = 100, *mutation rate* = 0.05, *replacement strategy* = elitism, *crossover rate* = 1, *n-elitism* = 1, *selection operator* = roulette, *tournament group size* = 2, *crossover operator* = OPX. In RS algorithm we used the values: *rounds* = 1000. In SSGA algorithm we used the values: *individuals* = 1000, *population size* = 100, *mutation rate* = 0.05, *replacement operator* = worst, *selection operator* = roulette, *tournament group size* = 2, *crossover operator* = OPX, *mutation operator* = random swap. In the SXGA algorithm we used the values: *generations* = 1000, *population size* = 100, *mutation rate* = 0.05, *male selection operator* = roulette, *female selection operator* = roulette, *tournament group size* = 2, *crossover operator* = OPX, *mutation operator* = random swap.

The VRPTW instances were obtained from Solomon [15] benchmark and to be used with the HeuristicLab software [26]. The HeuristicLab Optimization framework is for developing and testing optimization methods, parameters and applying these on a multitude of problems (Asymmetric Traveling Salesman Problem, GA-Based Machine Learning, Generic Genetic Programming, Genetic Regression, Job Shop Scheduling Problem, Multi-Processor Scheduling Problem, Rotor Machine Analysis Problem, Satisfiability Problem, 2-Dimensional Real-Valued Test Functions, N-Dimensional Real-Valued Test Functions, Vehicle Routing Problem). The project was started in 2002 and has evolved to a stable and productive optimization platform, it contains the algorithms: Ant Colony Optimization, Standard Evolution Strategy, Generic Genetic Algorithm, Parallel Genetic Algorithm (Island Model), Particle Swarm Optimization, Ra RS, Simulated Annealing, Segregative Genetic Algorithm, Standard Genetic Algorithm, Standard Genetic Algorithm for Machine Learning, Scatter Search, SSGA, Standard Tabu Search, and SXGA. The input parameters for the software were given random in the options: overload penalty, tardiness penalty, route-time penalty, travel time excess penalty, and distance penalty. In addition, we used the technique of discriminant analysis contained in SPSS [27] software.

The results are given based on the algorithm runtime on the given instance and the theoretical ratio (*TR*), which means how so close it was the solution that is reported in computer science Literature (15). Where *optimal know* = the best reported optimal solution, *NV* = Number of optimal vehicles reported, *Sol* = founded solution, *NVSol* = number of found vehicles:

$$TRatio = (Sol/NVsol) - (OptimalKnow\ /NV) \qquad (15)$$

In Table III are the obtained results of GA and SSGA algorithms with the VRPTW instances contained in HeuristicLab software.

TABLE III
OBTAINED RESULTS OF THE GA AND SSGA ALGORITHMS

Instances	Optimal	Vehicles	GA		SSGA	
			Time	*TR*	*Time*	*TR*
c101.100	827.3	10	25:46.7	13.91	00:54.6	24.38
c102.100	827.3	10	20:49.2	27.08	00:11.0	38.99
c201.100	589.1	3	46:56.7	0.81	00:23.3	0.81
c202.100	589.1	3	43:19.6	32.63	00:21.5	14.52
r101.100	1637.7	20	20:59.9	1.53	00:09.8	4.6784
r102.100	1466.6	18	17:41.7	6.17	00:09.9	10.37
rc101.100	1619.8	15	17:31.1	4.79	00:00.1	8.85
rc102.100	1457.4	14	18:36.6	4.76	00:10.3	16.28
rc201.100	1261.8	9	35:02.5	75.82	00:19.8	134.90
rc202.100	1092.3	8	39:49.2	76.74	00:19.7	96.60

For each instance was determined the list of algorithms that better solve it, defined the following evaluation criteria of algorithms: the run time divided between the obtained theoretical ratio (TR), and the smaller value is the algorithm champion with better performance for the instance. In Table IV are the obtained results of SXGA and RS algorithms with the VRPTW instances contained in HeuristicLab software.

TABLE IV
OBTAINED RESULTS OF THE SXGA AND RS ALGORITHMS

Instances	Optimal	Vehicles	SXGA		RS	
			Time	*TR*	*Time*	*TR*
c101.100	827.3	10	20:54.0	6.87	00:01.2	121.78
c102.100	827.3	10	19:27.6	14.68	00:00.2	115.76
c201.100	589.1	3	37:49.6	0.81	00:01.3	8.20
r101.100	1637.7	20	17:37.2	2.37	00:01.2	71.75
r102.100	1466.6	18	22:25.9	5.68	00:01.3	77.93
rc101.100	1619.8	15	18:12.4	1.36	00:00.2	95.45
rc102.100	1457.4	14	17:19.0	11.44	00:01.3	90.21
rc201.100	1261.8	9	32:55.5	105.14	00:03.3	90.31
rc202.100	1092.3	8	34:19.1	75.90	00:01.2	91.57

Table V shows the list of better algorithms for the sampling instances (obtains of Tables III and IV), as well as the calculation of the indicators proposed in each one of the instances. The values in Table IV were used like input values of the discriminant analysis.

TABLE V
OBTAINED RESULTS FROM THE INDICATORS OF THE PROBLEM INSTANCES

Instances	CN	ISIZE	IAREA	ITIME	Best Algorithm
c101.100	100	100	100.00	981.00	SXGA (3)
c102.100	100	100	100.00	956.00	SXGA (3)
c201.100	100	200	100.00	981.00	GA (1)
c202.100	100	50	95.00	3231.00	SSGA (2)
r101.100	100	75	79.00	205.00	GA (1)
r102.100	100	300	79.00	220.00	SXGA (3)
rc101.100	100	200	98.00	214.00	GA (1)
rc102.100	100	50	98.00	231.00	GA (1)
rc201.100	100	400	98.00	831.00	GA (1)
rc202.100	100	400	98.00	907.00	SXGA (3)

In order to obtain first criterion of algorithms classification, we used two indicators (CN and ISIZE) which they were used like independent variables, and the number of the best algorithm like dependent variable. The discriminant classification was trained with classic instances of VRPTW. In Table VI are the results of the discriminant analysis with the training instances (Tables III and IV), showing 38.1 % of the original group classified correctly.

TABLE VI
OBTAINED RESULTS OF DISCRIMINANT ANALYSIS 1

Group Origin	**Predicted** GA	**Group** SSGA	**Membership** SXGA	RS	TOTAL
GA	2	7	4	0	13
SSGA	1	2	0	0	3
SXGA	0	1	4	0	5
RS	0	0	0	0	0

In order to obtain the second criterion of algorithms classification, we used two indicators (*IAREA* and *ITIME*) which they were used like independent variables, and the number of the best algorithm like dependent variable. In Table VII are with the results of the discriminant analysis with the training instances, showing 57.1% of the original group classified correctly.

TABLE VII
OBTAINED RESULTS OF DISCRIMINANT ANALYSIS 2

Group Origin	**Predicted** GA	**Group** SSGA	**Membership** SXGA	RS	TOTAL
GA	7	1	5	0	13
SSGA	2	1	0	0	3
SXGA	1	0	4	0	5
RS	0	0	0	0	0

In order to obtain the third criterion of algorithms classification, we used three indicators (*IAREA*, *ISIZE*, *ITIME*) that was used like independent variables, and the number of the best algorithm like dependent variable was used. In Tables VIII and IX are with the results of the discriminant analysis with the training instances, showing 66.7 % of the original group classified correctly.

TABLE VIII
OBTAINED RESULTS OF DISCRIMINANT ANALYSIS 3

Group Origin	**Predicted** GA	**Group** SSGA	**Membership** SXGA	RS	TOTAL
GA	9	1	3	0	13
SSGA	2	1	0	0	3
SXGA	1	0	4	0	5
RS	0	0	0	0	0

TABLE IX
OBTAINED RESULTS OF THE DISCRIMINANT ANALYSIS FUNCTIONS

Functions	Wilks Lambda	Chi-square
Function 1	.590	8.977
Function 2	.866	2.446

TABLE X
EXAMPLE OF INSTANCES WITH ITS CHARACTERISTICS AND THE BEST ALGORITHM PREDICTED

I	*CN*	*ISIZE*	*IAREA*	*ITIME*	*Best Algorithm predicted*
c105.25	25	25	28.25	232.12	GA (1)
c105.100	100	100	100.00	983.00	SXGA (3)
r210.25	25	150	19.25	234.25	GA (1)
r210.100	100	600	79.00	945.00	SXGA (3)
rc208.25	25	100	20.00	217.37	GA (1)
rc208.100	100	400	98.00	841.00	SXGA (3)

If the functions are effective for the sample of VRPTW training instances, the percentage of the new observations classified correctly is an indicator of the effectiveness of the discriminant functions. In order to validate the effectiveness of the discriminant classification other instances of problem VRPTW were considered. Table X represents a fraction of the instances where is the result of the proposed indicators and the prediction of the best algorithm than solves the given instance. The prediction of the obtained classification of the discriminant analysis was of 66.7 %.

V. Conclusions

In this paper we demonstrated like true the hypothesis of the investigation, affirming that if the use of the discriminant analysis is possible as a function of predictive learning to select as it is the best EA than it solves an instance of an intelligent system for transport problem. The prediction of the obtained classification of the discriminant analysis was of 66.7 %. In this paper we proposed the use of discriminant functions that allow selecting among a set of EAs, the best one to solve a given situation.

As future works set out to improve the percentage of prediction, with the creation of new indicators to improve the performance of the algorithm selection.

Acknowledgment

This work was supported by grants of SEP-PRODEP UAEH-PTC-719 and UAEH-PTC-682, and the support of the National System of Researchers (SNI) of the National Council for Science and Technology (CONACYT).

References

[1] Huberty, C.J., Applied Discriminant Analysis, Wiley Interscience, New York, 1994.
[2] Rascón Chávez, O., Introduction of descriptive statistics. Editorial UNAM, 1983.
[3] Fisher, R.A., The use of multiple measurements in taxonomic problems. Annals of Eugenics Vol. 7, 1936, 179-188.
[4] Kantardzicv, M.M., Data mining: Concepts, Models, Methods, and Algorithms. Wiley IEEE Press, 2002.
[5] Mitchell, M., An Introduction to Genetic Algorithms. MIT Press, England, 1998.
[6] Rechenberg, I., Evolution Strategy, Frommann-Holzboog, Stuttgart, 1994
[7] Fogel, L., Owens, A., Walsh, M., Artificial intelligence through simulated evolution. Wiley, 1966.
[8] Cramer, N., A representation for the adaptive generation of simple sequential programs. In Proceedings of the International Conference on Genetic Algorithms and their Applications, 183-187, 1985.
[9] Koza, J.R. Genetic Programming: On Programming Computers by Means of Natural Selection and Genetics. MIT Press, Cambridge, MA, 1992.
[10] Holland, J.H., Adaptation in Natural and Artificial Systems. University of Michigan Press. Ann Arbor, 1975.
[11] Lederberg J., Tatum, E.L., "Novel Genotypes in mixed Cultures of Biochemical Mutants of Bacteria". Cold Spring Harbor Symposia of Quantitative Biology Vol. 11, 1946.
[12] Cook, S.A., "The Complexity of Theorem Proving Procedures", Proceedings of 3rd ACM Symposium on Theory of Computing, pp. 151-158, Shaker Heights, Ohio, United States, May 03–05, 1971, ACM, New York, NY, USA
[13] Garey, M.R. and Johnson, D.S., Computers and Intractability, a Guide to the Theory of NP-completeness. W. H. Freeman and Company, New York, 1979.
[14] Toth, P. and Vigo, D., The Vehicle Routing Problem. Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, 2001.
[15] Solomon, M.M., "Algorithms for Vehicle Routing and Scheduling Problems with Time Window Constrains", Operations Research Vol. 35 No. 2, pp. 254-265, 1987.
[16] Sussman, J.M., Perspectives on Intelligent Transportation Systems (ITS), Springer Verlag, 2005, ISBN: 978-0-387-23257-7
[17] Fink, E., "How to Solve it Automatically, Selection among Problem-solving Methods". Proceedings of the Fourth International Conference on AI Planning Systems AIPS'98, pp. 128-136, 1998.
[18] Cruz-Reyes, L., Classification of heuristics algorithms for the solution of Bin Packing Problem, Ph. D Thesis. Centro Nacional de Investigación y Desarrollo Tecnológico, 2004.
[19] Soares, C. and Brazdil, P., "Zoomed Ranking, Selection of Classification Algorithms Based on Relevant Performance Information", Principles of Data Mining in Knowledge Discovery 4th European Conference (PKDD 2000), LNAI 1910, Springer Verlag, Berlin Heidelberg New York, pp. 126-135, 2000.
[20] Blanton, J. and Wainwright, R.L., "Multiple Vehicle Routing with Time and Capacity Constrainst using Genetic Algorithms", Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, pp. 452-459, 1993.
[21] Thangiah, S.R., "Genetic Algorithms, Tabu Search and Simulated Annealing Methods for Vehicle Routing Problems with Time Windows", Practical Handbook of Genetic Algorithms: Complex Structures. L. Chambers Ed. CRC Press, 1998.
[22] Michie, D., Spiegelhalter, D.J. and Taylor, C.C., Machine Learning, Neural and Statistical Classification. Publisher Ellis Horwood, 1994.
[23] Affenzeller, M., "New Generic Hybrids Based Upon Genetic Algorithms", Advances in Artificial Intelligence. In IBERAMIA 2002, Lecture Notes in Artificial Intelligence 2527, Springer-Verlag, pp. 329-339, 2002.
[24] Dantzing, G.B. and Ramser, J.H., "The truck Dispatching Problem". Management Science Vol. 6 No. 80, 1959.
[25] Lis, J., Eiben, A., "A Multi-Sexual Genetic Algorithm for Multi-Objective Optimization", Proceedings of 1996 International Conference on Evolutionary Computing, IEEE, T. Fukuda, and T. Furuhashi (editors), Nagoyo, Japan, 1996, 59-64
[26] Wagner, S. and Affenzeller, M., "HeuristicLab: A Generic and Extensible Optimization Environment", Adaptive and Natural Computing Algorithms. Springer, Computer Science, pp. 538-541, 2005.
[27] SPSS, Inc. Headquarters, Chicago, Illinois. http://www.spss.com/es/

Jorge A. Ruiz-Vanoye was born in D.F. Mexico in 1975. He obtained his Ph.D. degree in Computer Science in 2008 from CENIDET. He has worked at the Electric Research Institute of Mexico government (IIE) and in diverse companies. He has given classes in diverse Mexican Universities since 1996. Dr Research Interests are: Algorithmic Finance, Algorithmic portfolio management, Applications and Theory of Algorithms, Bio-inspired Algorithms,Combinatorial Optimization Problems, Compilers, Computational Intelligence, Computational Complexity Theory, Computational Financial Intelligence, Complexity of Algorithms, Complexity of Instances, Computational Statistics, Computer Networks, Computer Science Security, Cybercrimes, Data-mining, Education, Evolutionary Computation, Heuristic Optimization Techniques in Bioinformatics, Hybrid evolutionary algorithms, Hybrid optimisation algorithms, Machine learning, Meta-heuristics, Operations Research, Operations Management, Parallel & Distributed Computing, Production Planning and Logistics Optimization, Project Scheduling Problems, Software Engineering, Transgenic Algorithms. He currently serves as a professor at the Autonomous University of Hidalgo State (UAEH) México and he is Mexican National Research level I (SNI 1) of the National Council of science and technology; for publications and more information see www.ruizvanoye.com.

Ocotlán Díaz-Parra received his Ph.d. in engineering and applied sciences by the CIICAp of the Autonomous University of the State of Morelos. She has worked at PEMEX and in diverse companies more. She has given classes in diverse Mexican Universities. Dr Research Interests are: Combinatorial Optimization, Genetic Algorithms, Vehicle routing, Heuristics Algorithms, Lineal Programing, Algorithmic, Evolutive algorithm, Artificial intelligence, software engineering, Computational intelligence, Computer Science Security, OpenMP/MPI Programming, Computer Education, meta-heuristics algorithms, Parallel & Distributed Computing. She is a Mexican National Research Level I (SNI 1) of the National Council of science and technology; for publications and more information see www.diazparra.net.

Alejandro Fuentes Penna has a PhD. in the Strategic planning and management of technology in 2012. He has published articles, book chapters and books in the areas of Software Engineering, education, HCI, health, strategic planning and Optimization Combinatorial level Congress, national and international indices, and JCR. Researcher in the State system of researchers of the Morelos State. He has worked in areas of development of information systems, Professor - researcher, reviewer of articles and books worldwide, consultant in areas of computer science, planning, administration and education, and as a Professor, researcher or consultant in research institutes, UNAM and other universities and consulting of information

technologies, architecture and education in Mexico. He currently works at the upper school of Tlahuelilpan - Autonomous University of the State of Hidalgo as research professor, head of the academic Area of engineering and coordinator of the Bachelor in computer systems.

Daniel Vélez-Díaz was born in Mexico City, Mexico. He received the B.S. degree in electronic engineering from the Metropolitan Autonomous University, Mexico City, Mexico, the M.S. degree in computational sciences from the Technologic Institute of Toluca, Toluca, Mexico, in 1989 and 1999, respectively; finally, he received the D.Eng. degree in electrical engineering at the National Autonomous University of Mexico, Mexico City. He worked at the National Institute of Nuclear Research on developing control algorithms based fuzzy logic for nuclear reactor, from 1991 to 2001. Currently he is researcher for the Autonomous University of Hidalgo State; his research interests include fuzzy control and intelligent control of nonlinear systems with uncertainties, and fuzzy modelling.

Edith Olaco Garcia is a student of BSc from the Universidad Autónoma del Estado de Hidalgo.