Journal of Intelligent & Fuzzy Systems 21 (2010) 57–64
DOI:10.3233/IFS-2010-0435
IOS Press

A genetic distance metric to discriminate the selection of algorithms for the general ATSP problem

J. Pérez-Ortega[a,*], R.A. Pazos R.[a], J.A. Ruiz-Vanoye[b], J. Frausto-Solís[c], J.J. González-Barbosa[d], H.J. Fraire-Huacuja[d] and O. Díaz-Parra[b]

[a] *Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Mexico*
[b] *Universidad Juárez Autónoma de Tabasco, Cunduacán, Tabasco, México*
[c] *Tecnológico de Monterrey Campus Cuernavaca, Cuernavaca, Mexico*
[d] *Instituto Tecnológico de Cd. Madero, Cd. Madero, Mexico*

Abstract. The only metric that had existed so far to determine the best algorithm for solving an general Asymmetric Traveling Salesman Problem (ATSP) instance is based on the number of cities; nevertheless, it is not sufficiently adequate for discriminating the best algorithm for solving an ATSP instance, thus the necessity for devising a new metric through the use of data-mining techniques. In this paper we propose: (1) the use of a genetic distance metric for improving the selection of the algorithms that best solve a given instance of the ATSP and (2) the use of discriminant analysis as a means for predictive learning (data-mining techniques) aiming at selecting meta-heuristic algorithms.

Keywords: Inductive learning, discriminant analysis, data-mining techniques, machine learning, genetic distance metric

1. Introduction

Discriminant analysis [13] is a multivariate statistical technique, whose purpose is to analyze if there exist significant differences between groups of objects with respect to a set of variables measured. Linear discriminant analysis was introduced by [7] as a statistical procedure for classification.

Classification is the most common task in generic inductive learning; additionally, it is a function of predictive learning that classifies data of diverse classes [15].

In this paper we propose the use of classification as a means for predictive learning in meta-heuristic algorithm selection, and a metric to improve the selection of algorithms that best solve a given instance of the Asymmetric Traveling Salesman Problem (ATSP). Metrics are mathematical formulations used for reflecting a certain situation; i.e., a relation between quantitative or qualitative variables that allows observing the situation and the tendencies of changes generated in the objects [13]. They can be used to measure the influence on algorithm performance.

The ATSP in operations research is a problem in discrete or combinatorial optimization. It is a prominent exemplification of a class of problems in computational complexity theory which are classified as Nondeterministic Polynomial time complete or NP-complete [4], the class of problems that are complete in the NP, that is to say, most difficult

*Corresponding author. E-mails: jpo_cenidet@yahoo.com.mx (J. Pérez-Ortega); r_pazos_r@yahoo.com.mx (R.A. Pazos R.); jruizvanoye@yahoo.com.mx (J.A. Ruiz-Vanoye); juan.frausto@itesm.mx (J. Frausto-Solís); jjgonzalezbarbosa@gmail.com (J.J. González-Barbosa); hfraire@prodigy.net.mx (H.J. Fraire-Huacuja); koko_diazparra@yahoo.com.mx (O. Díaz-Parra).

to solve in NP, and it is classified like decision problems.

The ATSP problem can be stated as follows: given a set of nodes and distances d_{ij} for each pair of nodes (i, j), where d_{ij} can be different from d_{ji}, find a circuit of minimal overall length that visits each of the nodes exactly once [3]. The ATSP problem is expressed by Eqs (1)–(4).

$$\min z(x) = \sum_{j=1}^{m}\sum_{i=1}^{m} d_{ij}x_{ij} \tag{1}$$

$$\sum_{j=1}^{m} x_{ij} = 1; \quad i = 1, \ldots, m \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = 1; \quad j = 1, \ldots, m \tag{3}$$

$$x_{ij} = \begin{cases} 1, & \text{if circuit traverses from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases} \quad \forall i, j \tag{4}$$

The parameters of ATSP instances can be coded in a encoding scheme: L = "nc, d_1, d_2, …, d_n", where nc represents the number of cities and d_i is the distance between pairs of cities.

The purpose of this work is to propose the use of a genetic distance metric in discriminant analysis for improving the selection of the algorithms that best solve a given instance of the general Asymmetric Traveling Salesman Problem (ATSP).

2. Related works

In the area of data mining, Fink [6] developed a technique of algorithm selection for decision problems, which is based on the estimation of the gain of an algorithm, obtained from the statistical analysis of its performance. The investigation group that worked in the METAL project [26], proposed a method to select an algorithm for a set of related cases. They find a set of old cases whose characteristics are the most similar to those of the new set. The algorithms performances for the old cases are known and used to predict the best algorithm for the new set of cases. Pérez-Cruz [20] proposed a methodology, based on automatic learning to systematically develop mathematical models for algorithm performance. The proposal consists of characterizing the performance of a set of metaheuristic algorithms applied to the solution of the Bin-Packing problem, through the determination of regions of superiority of the algorithms. Reinelt [23] mentions that the only metric that existed for the general Asymmetric Traveling Salesman Problem is based on the number of cities.

It is necessary to mention that none of the related works has performance comparison with this work because they do not approach the problem used in this investigation in some cases and in others not used the same algorithms classifier. But, we calculate the Reinelt metric and we realize comparisons of our discriminating classifier with Reinelt metric and our metric.

Table 1 shows the related works for this investigation. The purpose of this investigation consists of developing a method to improve the prediction on the algorithm that best solves the instances of a set of benchmark ATSP instances using a genetic distance metric in discriminant analysis.

At present, there exists a metric based on the number of cities; however, it is not sufficiently adequate for discriminating the best algorithm. Thus the following question arises: is it possible to improve predicting the best algorithm using additional metrics (Fig. 1).

3. Genetic distance metric for algorithm selection for the general ATSP

The only metric that existed for the general ATSP problem is based on the number of cities [23]; unfortunately, it is not sufficiently adequate for discriminating the best algorithm for solving a general ATSP instance, thus the necessity for devising a new metric.

Table 1
Related works

Research	Algorithms selection	Prediction of best algorithm	Algorithm characterization	Metric for ATSP	Discriminant analysis
Fink	√				
METAL	√	√			
Pérez-Cruz	√	√	√		
Reinelt				√	
This work	√	√	√	√	√

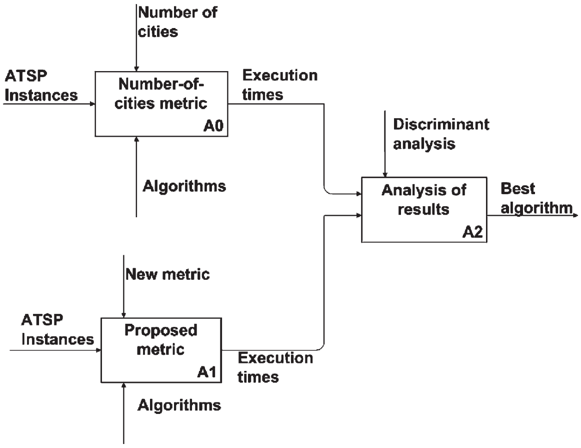

Fig. 1. Hypothesis of the investigation.

The purpose of this work is to develop a formal metric that helps improve algorithm selection for the general ATSP. To this end we used a concept from descriptive statistics called relative frequency. The index of similarity or genetic distance quantifies the similarity or difference in intercity distances (genetic is attributed to the quantification of the molecular markers; for example, in a jungle a lion and a leopard have some characteristics that are similar but there are also differences, which are reflected in their genes). The genetic term applied to ATSP indicates how much variability in the intercity distances exists; i.e. the number of occurrences in an interval of a frequency distribution.

The index of similarity S (Eq. (5)) expresses the similarity among intercity distances, which is obtained from the frequencies of the distances, where similarity of distances will exist as long as the frequencies of the distances are larger than 1. In general, the cases with many similar frequencies are considered less complex to solve by a certain type of algorithm.

$$S = \begin{cases} \sum Frequency(d_i) > 1, & \text{if} \quad Frequency(d_i) > 1 \text{ for some } i \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where S = index of similarity divided between one hundred, d_i = intercity distance, and $Frequency(d_i)$ = frequency (of occurrence) of distance d_i.

4. Experimentation

The experimentation was carried on an Acer Travelmate 2330LC computer with an Intel Celeron processor at 1.5 GHz, 512 MB of primary memory. The instances were obtained from the TSPLIB benchmark [23]. The metaheuristic algorithms tested were: a generic Genetic algorithm (GA), Standard Tabu Search (STS), Random Search (RS), and Scatter Search (SS) [1, 2, 29, 30].

A Genetic Algorithm (GA) is one of the heuristic methods used to find approximate solutions to NP-complete problems. GA is inspired by the darwinian principles of species evolution, and uses techniques pertaining to genetics, such as: inheritance, mutation, natural selection and recombination (or crossover). A simple GA works as follows [17]:

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. The selection is carried out "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 should be discarded at random.
4. Replace the current population with the new population.
5. Go to step 2.

The parameters for GA were: *selection operator* = roulette, *crossover operator* = OX, *crossover rate* = 1, *mutation rate* = 0.05, *generations* = 1000, *population size* = 100, *replacement strategy* = elitism, *n-elitism* = 1, *tournament group size* = 2.

Standard Tabu Search (STS) is a metaheuristic that guides a local heuristic search procedure to explore the solution space beyond local optimality. The concept of Tabu Search was introduced by Hansen and Jaumard [12] and Glover [10, 11]. A Standard Tabu Search works as follows [10]:

1. Choose an initial solution i in S.
   a. Set $i^* = i$ and $k = 0$.
2. Create a list of candidate solutions.
3. Evaluate solutions.
   a. Set $k = k + 1$ and generate a subset V^* of neighboring solutions in $N(i, k)$ such that either one of the Tabu conditions $t_r(i, m) \in T_r$ $(r = 1, \ldots, t)$ is violated or at least one of the aspiration conditions $a_r(i, m) \in A_r(i, m)$ $(r = 1, \ldots, a)$ holds.
4. Choose the best admissible solution.
   a. Choose the best j in V^* and set $i = j$.
5. Stopping condition satisfied?
   a. If the value of the objective function decreases $(f(i) < f(i^*))$, then set $i^* = i$.
6. If the stopping condition is met then stop; otherwise, update Tabu and aspiration conditions and afterwards go to Step 2.
7. Final solution.

where i, j: solution indexes; k: iteration index; V^*: subset of solution; $N(i, k)$: neighborhood of solution i at iteration k; $f(i)$: objective function value for solution i; m: Tabu move; $a_r(i, m)$: aspiration level; $A_r(i, m)$: threshold value. For STS the parameters were: stop criterion (*iterations* = 1000, *evaluations* = 10000), *tabu list tenure* = 10, *neighborhood size* = 10, *selection operator* = standard, *neighborhood operator* = simple inversion.

The main idea in the Random Search (RS) algorithm is to generate an initial solution with moderate quality. Then, according to some predefined neighborhood, 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 neighborhood; 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 neighborhood search technique relies heavily on the construction of the solution neighborhood. The pseudo-code for the Random Search (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.

The parameter for RS was: *rounds* = 1000.

Scatter Search (SS) was introduced by Glover [9] in a heuristic study on integer linear programming problems, which also provides a number of formative ideas of Tabu Search. Scatter Search is an evolutionary method that combines solution vectors through linear combinations to produce new ones through successive generations. The Scatter Search general procedure works as follows [9]:

1. Initial phase: a set of solutions is generated and the best solutions are chosen to become part of the reference set.
2. Evolutionary phase: repeat the following steps until some stopping criterion is met: a) New solutions are generated using weighted structured combinations of strategically selected reference subsets. b) A set of the best generated solutions will be included in the reference set.

The SS procedure is a very general process with many possible implementations. In order to specify a general framework, the SS/PR template proposed by Glover [9] is used. The template comprises 5 sub-routines, which can be sketched as follows:

I. Diversification Generation Method: starting with different seed solutions, a set of trial solutions is generated.
II. Improvement Method: given a trial solution as input, a heuristic process is applied to improve the solution; if the initial trial solution is not feasible, the heuristic procedure should be able to restore the solution feasibility during the improvement process.
III. Reference Set Update Method: maintains the reference set, consisting of a set of the best solutions found, widespread throughout the solution space.
IV. Subset Generation Method: uses a structured process that extracts solutions from the reference set and produces a subset, which will be used in the next method.
V. Solution Combination Method: combines the subset solution into one or more trial solutions.

For the SS algorithm the parameters used were: stop criterion (*steps* = 50, *evaluations* = 50000), *PSize* = 40, *HQ ref. set size* = 5, *div. ref. set Size* = 3, *diversification* = none, *similarity operator* = Hamming, *improvement method* = simple inversion, *improvement cycles* = 10, *neighborhood op* = simple inversion, *neighborhood size* = 10, *ref. set update method* = standard (more diversity), *subset generator* = pairs, *solution combinatory* = OX.

The discriminant analysis implemented in the SPSS software was used [27]. This kind of analysis is used as a machine-learning method to find the relation between the characteristics of a problem and the performance of an algorithm [15].

Table 2 shows some results obtained for GA and STS algorithms; while Table 3 shows some results for RS and SS algorithms.

The contribution of Tables 2 and 3 is to provide the results (current best or solution, time or run time, and ratio) of GA, STS, RS, and SS algorithms for each ATSP instance, with the purpose of to be base to determine the best algorithm than it solves a certain instance and to be used like input parameter for training the classifier. Where: The *ratio* or theoretical ratio obtained was the value of the current best (a solution of the algorithm) divided by the best know, *instances* = cases of TSPLIB benchmark, *Best Know* = best solution from benchmark, *time* = runtime of the algorithm, *Number of cities* = the instance size.

For each instance the set of algorithms that best solve it was determined; to this end the following criterion for algorithm evaluation was defined: the algorithm with the smallest value of the run time divided by the theoretical ratio is considered the best for the instance.

Table 4 shows the list of the best algorithms for the instances of a sample for training the classifier, as well as the calculation of the proposed metric S for each of the TSPLIB instances [23]. The contribution of Table 4 was to use the values as input for training the discriminant analysis (where *instances* = cases of TSPLIB instance, *number of cities* = the instance size, n = the index of cities and S is the index of similarity).

Table 5 shows the results from the discriminant analysis (where the origin group consists of the algorithms for training and the destination group is the object of the classification). We can observe that of a sample of 19 instances, using the discriminant analysis was assigned correctly (the GA origin group: 7 instances correctly in GA destination group, the RS origin group: 5 instances correctly in RS destination group and 1 instance incorrectly in the GA destination group), in addition for the STS origin group: 6 instances incorrectly were assigned to a GA destination group instead of to be assigned to STS destination group.

In order to obtain the classification criterion, two indicators (n and S) were used as independent variables, and the name of the best algorithm was used as dependent variable. A criterion of classification called group discriminant function (which is the criterion variable, also called the grouping variable in SPSS) was used later for classifying each new observation into the corresponding group. The percentage of the

Table 2
Sample of results obtained for GA and STS algorithms

Instances	Number of cities	Best know	GA			STS		
			Current best	Ratio	Time	Current best	Ratio	Time
br17	17	39	39	1.00	0.13	39	1.00	1.15
ft53	53	6905	10230	0.67	0.78	14418	0.47	1.44
ft70	70	38673	48888	0.79	2.29	54799	0.70	1.33
ftv33	33	1286	1482	0.86	1.99	2100	0.61	1.35
ftv35	35	1473	1926	0.76	2.18	2384	0.61	1.42
ftv38	38	1530	1949	0.78	2.17	2674	0.57	1.37
ftv44	44	1613	2135	0.75	2.23	3053	0.52	1.36
ftv47	47	1776	2410	0.73	2.26	3474	0.51	2.28
ftv55	55	1608	2721	0.59	2.12	3718	0.43	1.44
ftv64	64	1839	3789	0.48	2.44	4669	0.39	1.45
ftv70	70	1950	4239	0.46	2.33	4975	0.39	1.46
ftv170	170	2755	11007	0.25	2.88	16038	0.17	1.75
kro124p	124	36230	65223	0.55	2.54	76900	0.47	1.50
p43	43	5620	5812	0.96	2.24	5673	0.99	1.35
rbg323	323	1326	4012	0.33	3.78	5217	0.25	1.57
rbg358	358	1163	4695	0.24	4.29	6018	0.19	1.54
rbg403	403	2465	5355	0.46	4.39	6724	0.36	1.59
rbg443	443	2720	6226	0.43	5.017	7367	0.36	1.74

Table 3
Sample of results obtained for RS and SS algorithms

Instances	Number of cities	Best know	RS			SS		
			Current best	Ratio	Time	Current best	Ratio	Time
br17	17	39	90	0.43	0.97	39	1.00	1.21
ft53	53	6905	21461	0.32	1.06	13565	0.50	3.44
ft70	70	38673	65724	0.58	1.07	52935	0.73	5.10
ftv33	33	1286	3441	0.37	1.12	1887	0.68	2.18
ftv35	35	1473	3931	0.37	1.02	1945	0.75	2.50
ftv38	38	1530	4299	0.35	1.09	2106	0.72	2.61
ftv44	44	1613	4859	0.33	1.04	2887	0.55	2.98
ftv47	47	1776	5730	0.30	1.19	3491	0.50	3.30
ftv55	55	1608	6264	0.25	0.99	2898	0.55	3.81
ftv64	64	1839	7144	0.25	1.04	4377	0.42	4.72
ftv70	70	1950	8269	0.23	1.02	4659	0.41	5.06
ftv170	170	2755	23418	0.11	1.04	15171	0.18	17.29
kro124p	124	36230	163489	0.22	1.02	66145	0.54	7.74
p43	43	5620	11849	0.47	1.01	5634	0.99	3.12
rbg323	323	1326	5827	0.22	1.12	4892	0.27	53.94
rbg358	358	1163	6490	0.17	1.22	5560	0.20	01:08.6
rbg403	403	2465	7270	0.33	1.28	6157	0.40	01:17.1
rbg443	443	2720	7799	0.34	1.39	6786	0.40	01:31.8

Table 4
Results obtained using the metric on the solution of the training ATSP instances

Instances	Number of cities	n	S	Best algorithm
br17	17	17	0.53	GA
ft53	53	53	0.32	GA
ft70	70	70	0.24	RS
ftv33	33	33	0.22	STS
ftv35	35	35	0.36	STS
ftv38	38	38	0.26	GA
ftv44	44	44	0.3	STS
ftv47	47	47	0.48	GA
ftv55	55	55	0.56	STS
ftv64	64	64	0.65	STS
ftv70	70	70	0.71	STS
ftv170	170	170	1.7	RS
kro124p	124	124	1.0	STS
p43	43	43	1.03	STS
rbg323	323	323	49.28	RS
rbg358	358	358	81.15	RS
rbg403	403	403	121.04	RS
rbg443	443	443	894	RS
ry48p	48	48	0.12	STS
ftv170	170	170	1.7	RS
p43	43	43	1.03	STS

new observations correctly classified is an indicator of the effectiveness of the discriminant functions. It is assumed that if the functions are effective for the training sample, they will classify new instances correctly. To validate the effectiveness of the discriminant classifier, we used other ATSP instances.

Table 6 shows the results for a fraction of the instances. For each instance, it displays the result of the proposed metric (S), metric n, the prediction of the best algorithm that solves the given instance, the discriminant Squared Mahalanobis distance (it is based on correlations between variables by which different patterns can be identified and analyzed. It is a useful way of determining similarity of an unknown sample set to a known one) to the centroid, and discriminant functions 1 and 2 (Canonical Discrimininat Functions, the first function maximizes the differences between the values of the dependent variable. The second function is orthogonal to it (uncorrelated with it) and maximizes the differences between values of the dependent variable, controlling for the first factor). The results were obtained for validating the effectiveness of the discriminant classifier. The prediction of the classifier using the n and S indicators was of 58.9% of cases correctly classified.

Table 5
Results from the discriminant analysis

Origin group	Destination group			Total instances
	GA	STS	RS	
GA	7	0	0	7
STS	6	0	0	6
RS	1	0	5	6

5. Conclusions

The main contribution of this work is the proposal of a similarity metric or genetic distance metric that

Table 6
Example of instances with their characteristics and the best algorithm predicted

Instances	Characteristics				Squared	Function 1	Function 2
	Number of cities	n	S	Best algorithm	Mahalanobis Distance to Centroid		
ftv160	160	160	1.61	GA	2.468	0.549	−0.510
ftv110	110	110	1.1	GA	0.702	−0.145	−0.275
ftv120	120	120	1.21	GA	0.969	−0.006	−0.322
dc112	112	112	1.12	GA	0.752	−0.117	−0.284
dc126	126	126	1.26	STS	1.236	0.77	−0.350
dc134	134	134	1.34	STS	1.510	0.188	−0.388

quantifies the similarity or difference among intercity distances (through the distribution of distance frequencies), a method devised to improve the prediction on the algorithm that best solves a benchmark set of ATSP instances, which improves the prediction of the best solution algorithm. From the experimental results (47.4% of cases correctly classified with indicator n and 58.9% of cases correctly classified with indicators n and S), it follows that it is possible improve discrimination by adding another metric to the ATSP problem.

Future work would aim at devising additional metrics to increase the prediction percentage on the algorithm that best solves a given ATSP instance. The results were obtained for validating the effectiveness of the discriminant classifier.

References

[1] M. Affenzeller, New generic hybrids based upon genetic algorithms, In: F.J. Garijo, J.C. Riquelme and M. Toro (Eds), *Advances in Artificial Intelligence, IBERAMIA 2002* **2527** (2002), 329–339, Springer-Verlag. Lecture Notes in Artificial Intelligence.

[2] M. Affenzeller, New variants of genetic algorithms applied to problems of combinatorial optimization, *Proceedings of the EMCSR 2002* **1** (2002), 75–80.

[3] J. Cirasella, D.S. Johnson, L.A. McGeoch and W. Zhang, The asymmetric traveling salesman problem: algorithms, instance generators, and tests, A.L. Buchsbaum and J. Snoeyink (Eds), *Third International Workshop*, *ALENEX 2001* **2153** (2001), 32–59. Springer-Verlag, Lecture Notes in Computer Science.

[4] S.A. Cook, The complexity of theorem proving procedures, *Proceedings of 3rd ACM Symposium on Theory of Computing* (1971), 151–158.

[5] M.A. Cruz-Chávez, O. Díaz-Parra, D. Juárez-Romero and M.G. Martínez-Rangel, Memetic algorithm based on a constraint satisfaction technique for VRPTW, in: L. Rutkowski, et al. (Eds), *Artificial Intelligence and Soft Computing – ICAISC 2008* **5097** (2008), 376–387. Springer-Verlag, Lecture Notes in Artificial Intelligence.

[6] E. Fink, How to solve it automatically selection among problem-solving methods, *Proceedings of the Fourth International Conference on AI Planning Systems AIPS'98* (1998), 128–136.

[7] R.A. Fisher, The use of multiple measurements in taxonomic problems, *Annual of Eugenics* **7** (1936), 179–188.

[8] F. Glover, A template for scatter search and path relinking, in: J.K. Hao, E. Lutton, E. Ronald, M. Schoenaurer and D. Snyers (Eds), *Artificial Evolution* **1363** (1998), 1–51. Springer-Verlag, Lecture Notes in Computer Science.

[9] F. Glover, Heuristics for integer programming using surrogate constraints, *Decision Science* **8** (1977), 156–166.

[10] F. Glover, Tabu search: Part I, *ORSA Journal on Computing* **1** (1989), 190–206.

[11] F. Glover, Tabu search: Part II, *ORSA Journal on Computing* **2** (1990), 4–32.

[12] P. Hansen and B. Jaumard, Algorithms for the maximum satisfiability problem, *RUTCOR Research Report* (1987), 43–87. Rutgers University, USA.

[13] C.J. Huberty, *Applied Discriminant Analysis*, Wiley, Interscience, 1994.

[14] L.C. Jain, S.C. Tan and C.P. Lim, An introduction to computational intelligence paradigms, *Computational Intelligence Paradigms* **137** (2008), Springer-Verlag. Studies in Computational Intelligence.

[15] M. Kantardzic, *Data Mining: Concepts, Models, Methods, and Algorithms*, John Wiley & Sons, 2003.

[16] P. Larrañaga, C.M.H. Kuijpers, R.H. Murga, I. Inza and S. Dizdarevic, Genetic algorithms for the travelling salesman problem: A review of representations and operators, *Artificial Intelligence Review* **13**(2) (1999), 129–170. Springer-Verlag.

[17] M. Mitchel, *An Introduction to Genetic Algorithms*, MIT Press, 1999.

[18] T.M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

[19] J. Pérez, R.A. Pazos, J. Frausto, L. Cruz, H. Fraire, E. Santiago and N.E. García, A machine learning approach for modeling algorithm performance predictors, in: V. Torra and Y. Narukawa (Eds), *Modeling Decisions for Artificial Intelligence – MDAI 2004* **3131** (2004), 70–80. Springer-Verlag, Lecture Notes in Computer Science.

[20] J. Pérez, R.A. Pazos, J. Frausto, G. Rodríguez, D. Romero and L. Cruz, A statistical approach for algorithm selection, in: C.C. Ribero and S.L. Martins (Eds), *Proceedings of III International Workshop on Experimental and Efficient Algorithms, WEA 2004* **3059** (2004), 417–431. Springer-Verlag, Lecture Notes in Computer Science.

[21] J. Pérez, R.A. Pazos, J. Frausto, G. Rodríguez, L. Cruz and H. Fraire, Comparison and selection of exact and heuristic algorithm, in: A. Laganá, et al. (Eds), *Computational Science*

and Its Applications – ICCSA 2004 **345** (2004), 415–424. Springer-Verlag, Lecture Notes in Computer Science.

[22] G. Reinelt, Basic concepts, *The Traveling Salesman: Computational Solutions for TSP Applications* **840** (1994), 4–30. Springer-Verlag, Lecture Notes in Computer Science.

[23] G. Reinelt, TSPLIB – A traveling salesman problem library, *ORSA Journal on Computing* **3** (1991), 376–384.

[24] J.A. Ruiz-Vanoye, O. Díaz-Parra and V. Landero, A metric to discriminate the selection of algorithms for general problem ATSP, in: I. Lovrek, R.J. Howlett and L.C. Jain (Eds), *Knowledge-Based Intelligent Information and Engineering Systems, KES 2008* **5177**(Part I) (2008), 106–113. Springer-Verlag, Lecture Notes in Artificial Intelligence.

[25] J.A. Ruiz-Vanoye, J.A. Zárate, O. Díaz-Parra and V. Landero, Applied statistical indicators to the vehicle routing problem with time windows for discriminate appropriately the best algorithm, in: O. Gervasi, et al. (Eds), *Computational Science and Its Applications – ICCSA 2008* **5073**(Part II) (2008), 1131–1141. Springer-Verlag, Lecture Notes in Artificial Intelligence.

[26] C. Soares and P. Brazdil, Zoomed ranking selection of classification algorithms based on relevant performance information, in: D.A. Zighed, J. Komorowski and J. Zytkow (Eds), *Principles of Data Mining in Knowledge Discovery 4th European Conference, PKDD 2000* **1910** (2000), 126–135. Springer-Verlag, Lecture Notes in Artificial Intelligence Series.

[27] SPSS Inc. Headquarters, Chicago, Illinois, 2008.

[28] K.H. Tan, C.P. Lim, K. Platts and H.S. Koay, Managing manufacturing technology investments: an intelligent learning system approach, *International Journal of Computer Integrated Manufacturing Special Issue*, Taylor & Francis **19**(1) (2006), 4–13.

[29] S. Wagner and M. Affenzeller, The heuristicLab optimization environment. technical report, *Institute of Formal Models and Verification*, Johannes Kepler University Linz. Austria, 2004.

[30] S. Wagner and M. Affenzeller, Heuristiclab a generic and Extensible optimization environment, *Adaptive and Natural Computing Algorithms* (2005), 538–541. Springer-Verlag, Springer Computer Science.

[31] I.H. Witten and E. Frank, *Data Mining Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers, Elsevier, 2005.