The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.54 e-ISSN: 2251-7545

Assessing the Performance of Electimize in Solving NP-Complete Optimization Problems

Mohamed Abdel-Raheem
Assistant Professor, Dept. of Engineering Technology
Missouri Western State University
Saint Joseph, MO, USA
e-mail: maraheem@knights.ucf.edu

Ahmed Khalafallah
Assistant Professor; Dept. of Archit. & Manuf. Sciences
Western Kentucky University
Bowling Green, KY, USA
e-mail: ahmed.khalafallah@wku.edu

***Abstract*—Electimize is a new evolutionary algorithm (EA) algorithm that was introduced to overcome some limitations of existing evolutionary algorithms. Electimize simulates the phenomenon of the electrical current conductivity through the representation of solution strings as wires in closed electric circuits. Unlike some EAs, Electimize has the ability to assess the quality of each value in the solution string independently. The assessment of values in potential solution is based on Ohm's law and Kirchhoff's rule.**

One of the primary objectives of developing Electimize is to devise additional capabilities that would enable the algorithm to solve a wide range of discrete optimization problems. Specifically, this paper aims to: 1) assess the capabilities of the algorithm in solving a challenging class of discrete optimization problems, namely, NP-complete optimization problems, 2) compare the performance of Electimize to other EAs that were used to solve this class of problems. For this purpose, an instant (Bayg29) of the traveling salesman problem (TSP) was selected for the testing, application and comparison purposes.

Keywords-optimization; Electimize; evolutionary algorithms; NP-complete; traveling salesman problem

I. INTRODUCTION

Due to the limitations of some of the existing evolutionary algorithms (EAs), the main objective of developing Electimize was to devise an optimization algorithm with capabilities higher than those of existing EAs [1, 2, 3]. These capabilities includes three main aspects: 1) the ability to assess the quality of each value available in the solution space independently from other values in the solution string; 2) the ability to extensively search the solution space, and identify optimal and alternate optimal solutions –if any; and 3) the ability to solve a wide range of discrete optimization problems [1, 4, 5].

Previous studies demonstrated that Electimize has an outstanding performance when it comes to solving static and dynamic NP-hard combinatorial optimization problems. In fact, Electimize has notably outperformed many of the current EAs that are commonly used in solving these problems [6, 7]. The capability of the algorithm to independently assess the quality of each value of the solution string was investigated and demonstrated thoroughly [1, 3, 6, 7]. This unique capability enables the algorithm to effectively converge towards the optimal values, as demonstrated by investigating the change in the probability of selection of different decision variable values that showed incremental convergence towards the optimal solution(s) [3]. The capability of Electimize to extensively search the solution space was also demonstrated by the algorithm capacity to identify several alternate optimal solutions throughout experimentation [1, 3]. Electimize is also enhanced with a number of internal processes that supports the rapid convergence toward optimal solutions compared to other EAs [4, 5], as will be demonstrated later in this paper. The role of these internal processes was most remarkable in determining the optimal solutions of a benchmark time-cost tradeoff problem after the first iteration [6], and in identifying a new optimal solution for a benchmark site layout planning problem [7].

Despite the notable performance of Electimize in solving various classes of combinatorial optimization problems, it is deemed essential to further investigate its performance in solving the NP-complete optimization problems.

NP-complete optimization problems are the hardest problems in the NP-hard class. If a polynomial-time algorithm can solve an NP-complete problem, then it should solve other NP-hard problems in this class in a polynomial time [1, 8]. A very well-known example of the NP-complete problem is the traveling salesman problem. In this paper, an instance of the traveling salesman problem (Bayg29) was selected for testing and application purposes.

Probably the traveling salesman problem (TSP) is the most famous and most attempted combinatorial optimization problem. The TSP is the problem of finding the shortest tour among a number of cities in a given set. The TSP can have different objectives. The objective can be finding: 1) Shortest travel distance, 2) Shortest travel time, or 3) Least travel cost between a number of cities. TSPs vary in size (number of cities), and symmetry.

The research literature for the TSP is huge due to the generic nature of the problem, its application in various

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.54 e-ISSN: 2251-7545

engineering field, and its utilization as a benchmark problem for testing mathematical optimization models and EAs [9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20].

II. ELECTIMIZE: ALGORITHM AND INTERNAL PROCESSES

Electimize simulates the phenomenon of the flow of electrons in an electric circuit, where the wire with the least resistance has the maximum flow and the highest electric intensity [1, 3, 4, 5, 6, 7]. In Electimize, a feasible solution is represented by an electric wire. The simulation starts by a random fabrication of a population of N wires . Each wire (W_n) is composed of a number of segments (M) that represent the decision variables. Each segment (m) is assigned a value (l_{nm}) from the range of the relevant decision variable. Each assigned value is assumed to have a local resistance (r_{nm}) that pertains to its quality and relates to the physical characteristics of the segment (resistivity, cross-sectional area, and length). The fabricated wires are then connected in parallel to a source of electricity with voltage (V), a value determined by Electimize in order to differentiate between the qualities of different wires (W_n) [1, 4, 5] .

The quality of a solution (wire) is represented by its global resistance (R_n). The Global resistance is determined using Ohm's Law (R= V/I). The local and global resistances obey Kirchhoff's law, such that: at any time, the summation of the local resistances (r_{nm}) of the values (l_{nm}) of a single wire (W_n) should be equal to the global resistance (R_n) of the wire. . The intensity (I_n) of the electric current passing through each wire (w_n) is determined by substituting the value (l_{nm}) of each wire segment (m) in their corresponding optimization variable. Once the global resistance (R_n) is determined, the local resistances of segments (r_{nm}) are initially calculated, according to (1). The algorithm assumes that, at first, all values of the same wire have the same quality since there is no available information about their qualities yet. The next step is to assess the quality (r^*_{ml}) of each value (l_{nm}) in wire (W_n) independently.

$$r_{nm} = \frac{R_n}{M} \tag{1}$$

The wires are then ranked according to their global resistances (R_n). In a minimization problem, the goal is to find the wire with the maximum resistance (min I_n). The best wire (W_{best}) in the population is then used to evaluate other wires and is referred to as the control wire (CW). The top 5-25% of the wire population are then selected to undergo a one-way sensitivity analysis to assess the quality of each value (l_{nm}) independently. This is accomplished by substituting the value (l_{nm}) of each wire segment (m) in its corresponding segment in the CW. The new intensity of the CW is then recorded, the change in the CW global resistances is calculated, and the local resistances (r_{ml}) are then modified accordingly as illustrated in (2). The resistances (r_{ml}) are then updated, according to (3). The final step is the calculation of selection probability (P_{ml}) based on the updated resistance of values, as shown in (4) – for maximization problems.

$$r^*_{ml} = [r_{nm}(1-\left[\frac{R_{cw}-R_n}{R_{cw}}\right])] * R_n \Big/ \sum_{m=1}^{M} [r_{nm} \times \left[\frac{R_{cw}-R_n}{R_{cw}}\right]] \tag{2}$$

Where r^*_{ml}: modified resistance of value (l) occupying segment (m); r_{nm}: resistance of value (l_{ml}) of segment (m) in the original wire (W_n); R_n: global resistance of wire (W_n); and R_{CW}: resistance of the control wire.

$$\hat{r}_{ml} = r_{ml} + r^*_{ml} \tag{3}$$

Where $r^\wedge_{ml}$: updated resistance for value (l) of segment (m), and r_{ml}: resistance for value (l) of segment (m) from the previous iteration.

$$P_{ml} = \frac{1/\hat{r}_{ml}}{\sum_{l=1}^{L} 1/\hat{r}_{ml}} \tag{4}$$

Where P_{ml}: probability that value (l_{ml}) is selected for segment (m).

The above mentioned steps represent one iteration. At the end of each iteration, the wires are dismantled and parts are re-used to fabricate new wires. This process continues up to a specified number of iterations, and then terminates [1, 4, 5, 6, 7].

Electimize has a number of internal processes that enhance the search for optimal values in the solution space, and accelerates the rate of convergence. The first internal process is the decomposition of the local resistance (r_{nm}) into its main parameters, according to (5).

$$r_{ml} = \frac{\rho \times b_{ml}}{a_{ml}} \tag{5}$$

Where, ρ: resistivity of wire material - for simplicity, ρ=1, a_{ml}: cross-sectional area of value (l_m); and b_{ml}: length of value (l_m)

The cross-sectional area should represent a piece of information about the problem that is available beforehand and can guide the algorithm toward the optimal solution [1, 4, 5]. If this information is available, then the selection probability will be calculated according to (6). The second internal process is the utilization of the Heat Factor (HF), where the resistances (r_{ml}) are multiplied by a HF (a value less than one) if selected more than a specified number of times set by the user. The utilization of the HF prevents premature convergence and allows for the extensive search of the solution space [1, 4, 5]. Another internal process is the

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.54 e-ISSN: 2251-7545

determination of the control wire (CW) for the sensitivity analysis step. For the majority of iterations (90-95%), the CW is selected as the best wire among the current population of wires. Toward the end of simulation, the CW is selected as the best wire identified in all iterations, a process that allows for identifying the global optimal and alternate optimal solutions. Also, during the sensitivity analysis process, the selected CW is replaced by any wire of a better quality generated during this step. This process of changing the CW is yet another form of internal evolution that contributes to the robustness of the algorithm. The main nine steps of the algorithm, along with the interactions occurring with different internal processes, are shown in Fig. 1. A detailed description of Electimize and its operation is available in the literature [1, 4, 5, 6, 7].

$$P_{ml} = \frac{1/b_l}{\sum_{l=1}^{L} 1/b_l} \tag{6}$$

III. MODELING THE TRAVELING SALESMAN PROBLEM (TSP) USING ELECTIMIZE

In modeling the TSP using Electimize, N wires, each composed of M segments, are fabricated to represent possible tours. The number of segments (M) should correspond to the number of cities in the tour. For example, in a 14-city TSP, the wire will be composed of 14 segments. Each segment is randomly assigned a certain city in the tour, as shown in Fig. 2. The tour always starts from a specified city and ends at same specified city. This means that the first and last segments of the wire will always have the same yet fixed indexes. The optimization objective is to find the tour with the minimum distance. The objective function is stated as follows:

$$\text{Minimize}: \sum_{j=i+1}^{m} D_{ij}$$

where, D_{ij}: distance between cities (i) and (j), and m: total number of segments in the wire. The travel distances between cities are known before hand and stored in the distance matrix.

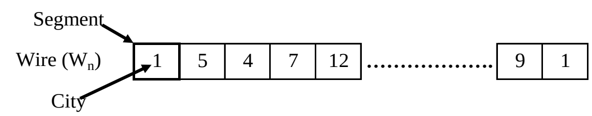

Figure 2. Wire representation of the traveling salesman problem

The cross-sectional area (a_{ml}) was calculated as total distance of the tour, divided by the distance between any two cities in the tour, as shown in Fig. 3.

	Segments								
W_n	1	?							

City	Distance Matrix								
i \ j	1	2	3	4	5	6	7	8	9	Total
1	0	23	12	2	21	15	8	6	15	102

Area Matrix									
Area	1	2	3	4	5	6	7	8	9
1	0	102/23=4.43	12/102=8.5	51.00	4.86	6.80	12.75	17.00	6.80

Figure 3. Areas of decision variable vales for the first wire segment

Electimize was coded using Visual Basic for Application as a macro in Microsoft Excel. A simple interface was developed for the user to input the data, including the number of iterations, number of wires, size of the instance, and number of wires performing the sensitivity analysis, as shown in Fig. 4. Separate multiple sheets were prepared to store distance matrixes and the output solutions. Various data were collected to trace and verify the logic in computation.

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.54 e-ISSN: 2251-7545

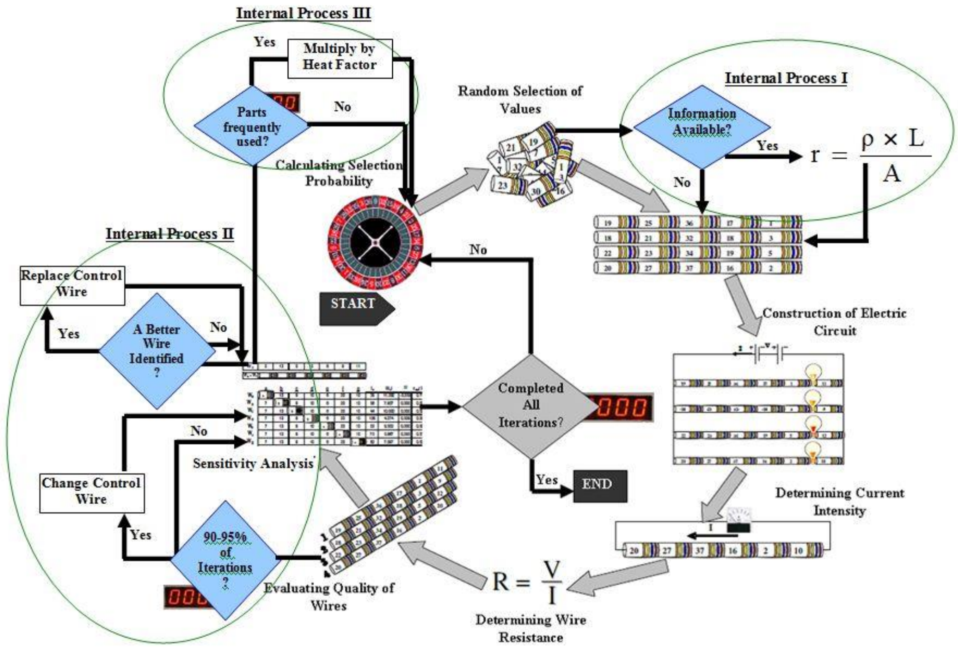

Figure 1. Main steps of optimization and the interaction with different internal processes.

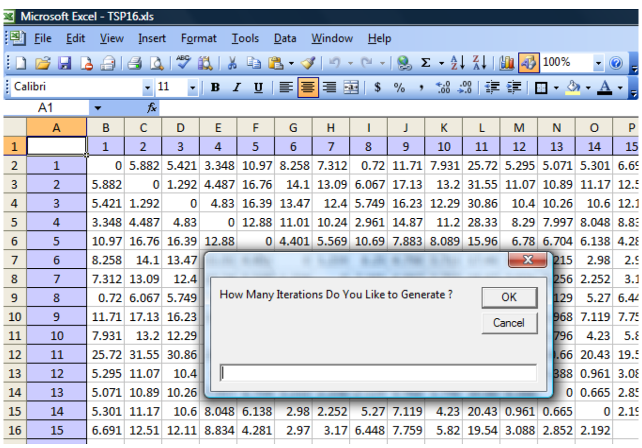

Figure 4. A simple user interface for the TSP using spreadsheets

IV. EXPERIMENTATION AND RESULTS FOR THE BAYG29 TSP

Bayg29 is a real-life benchmark symmetric TSP, with an objective of finding the optimal tour between 29 cities in Bavaria. The problem has 3.05 x 10^{29} possible solutions. The distances between cities are geographical.

The algorithm was applied successfully and determined the optimal solution for the Bayg29 TSP (1,610 km) in 17 iterations using 500 wires and the top 50 wires in each iteration for the sensitivity analysis (see Fig. 5). The Control Wire was defined as the best wire identified in each iteration in 90% of the iterations, and the overall best wire in the remaining 10% of iterations. The resistance for each decision variable value (r_{ml}) is multiplied by a Heat Factor of 0.4, if the value is selected more than 5,000 times. The optimal tour and some of the sub-optimal tours identified are listed in Table I. From Table I, it can be noticed that some of the good wires have been evolving incrementally throughout the iterations and until the optimal solution is generated.

V. COMPARISON WITH PREVIOUS RESULTS FOR BAYG29 TSP

The results obtained by Electimize were compared to the most recent attempts made to solve the Bayg29 TSP. Shimomura et al. [21] introduced a modified version of Ant Colony Optimization (ACO) referred to as Ant Colony Optimization with Intelligent and Dull Ants (IDACO). They attempted Bayg29 TSP to compare the performance of their proposed algorithm (IDACO) and traditional ACO. Although they proved that IDACO outperforms ACO, both algorithms failed to reach the optimal solution of Bayg29. The best result obtained by IDACO had an error rate of 1.38% compared to an error rate of 2.49% of ACO. The error rate is calculated according to (6). The error rates given were used to calculate the tour lengths obtained, as shown in Table II. Another interesting observation is that some of the near-optimal solutions identified by Electimize (see Table I) are better than the best values reported by IDACO, ACO, and GAs. This is also apparent in Fig. 5, which illustrates Electimize's high convergence rate demonstrated by the

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.54 e-ISSN: 2251-7545

dramatic drop in the error rate from 110.8% in the first iteration to 0.74% in the eleventh iteration.

$$\text{Error Rate (\%)} = \frac{\text{algorithm solution - optimal solution}}{\text{optimal solution}} \quad (6)$$

In an attempt to compare the performance of GAs after the introduction of a new crossover method referred to as Sequential Constructive Crossover (SCX), Ahmed [22] selected to try the Bayg29 problem. The new crossover method proved to be more efficient as it reached the optimal solution (1,610 km), outperforming the original GA that had an error rate of 9.25%, as shown in Table II.Like SCX algorithm, Electimize was able to determine the optimal solution of the problem. Unfortunately, further data about SCX such as the convergence rate and the number of generations and chromosomes used is not available in the literature. This prevented further comparison between Electimize and SCX.

VI. Conclusion

This paper presents an assessment of Electimize performance in solving NP-complete combinatorial optimization problems. An instance of the traveling salesman problem (Bayg29) was selected for application; and Electimize was able to identify the optimal tour with a high convergence rate, as illustrated by the results . The robustness of the algorithm is attributed to three main internal processes that enable it to extensively and efficiently search the solution space for the optimal solutions. This robustness is further illustrated by comparing the intermediate near-optimal values generated during the search to the final solutions obtained by other algorithms. Electimize proves to be an efficient tool for optimization and is capable of solving NP-complete optimization problems, the hardest class of NP-hard optimization problems.

References

[1] M. Abdel-Raheem, Electimize: A New Evolutionary Algorithm for Optimization with Application in Construction Engineering. Ph.D Dissertation, College of Engineering and Computer Science, University of Central Florida, USA, 2011.

[2] M. Abdel-Raheem, and A. Khalafallah, "Framework for a Multi-Level Evolutionary Algorithm for Construction Optimization," Proc. of the 23rd European Conf. on Modeling and Simulation (ECMS 2009), June 2009.

[3] A. Khalafallah, and M. Abdel-Raheem, "Electimize: A New Optimization Algorithm with Application in Construction Engineering," J. of Computing in Civil Engineering, vol. 25, 2011, pp. 192-202, 2011.

[4] M. Abdel-Raheem, and A. Khalafallah, "Introduction to a New Mechanism for Electimize," Proc. of UKSim 13th International Conference on Modeling and Simulation (UKSim2011), March 2011, pp. 8-13.

[5] M. Abdel-Raheem, and A. Khalafallah, "Modeling Combinatorial Optimization Problems Using Electimize," Proc. of Conference on Systems Engineering Research (CSER 2013), Georgia Institute of Technology, March 2013, in press.

[6] M. Abdel-Raheem, and A. Khalafallah, "Using Electimize to Solve the Time-Cost-Tradeoff Problem in Construction Engineering," Proc. of 2011 ASCE International Workshop on Computing in Civil Engineering, June 2011, pp. 250-257.

[7] M. Abdel-Raheem, and A. Khalafallah, "Application of Electimize in Solving the Construction Site Layout Planning Optimization Problem," Proc. of Construction Research Congress (CRC 2012), May 2012.

[8] V. Blondel, and J. Tsitsiklis, "Complexity of stability and controllability of elementary hybrid system," Automatica, vol. 35, 1999, pp. 479-489.

[9] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling salesman problem," Operations Research, vol. 2, 1954, pp. 393–410 .

[10] D. Rosenkrantz, R. Stearns, and P. Lewis, "An Analysis of Several Heuristics for the Traveling Salesman Problem", SIAM Journal on Computing, 1977, pp. 563–581.

[11] M. Garey, and D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, 1979 .

[12] D. Shmoys, J. Lenstra, K. Rinnooy, and E. Lawler, The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, John Wiley & Sons, 1985 .

[13] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison–Wesely Publishing Company, Inc., New York, 1989.

[14] D. Johnson, and L. McGeoch, The Traveling Salesman Problem: A Case Study in Local Optimization, in L. Aarts, and J. Lenstra, John Wiley and Sons Ltd, 1997 .

[15] S. Arora, S. (1998), "Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems", J. of the ACM, vol. 5, 1988, pp. 753–782.

[16] C. Walshaw, A Multilevel Approach to the Traveling Salesman Problem, CMS Press, 2000.

[17] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, The traveling-salesman problem: Introduction to Algorithms (2nd ed.), MIT Press and McGraw-Hill, 2001.

[18] G. Gutin, A. Yeo, and A. Zverovich, "Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP", Discrete Applied Mathematics, vol. 117, 2002.

[19] G. Gutin, and A. Punnen, The Traveling Salesman Problem and Its Variations, Springer, 2006.

[20] D. Applegate, R. Bixby, V. Chvátal, and J. Cook, The Traveling Salesman Problem: A Computational Study, Princeton University Press, 2006 .

[21] S. Shimomura, M. Sugimoto, T. Haraguchi, H. Matsushita, and Y. Nishio , "Investigation of Ant Colony Optimization with Intelligent and Dull Ants," Proc. of International Workshop on Nonlinear Circuits, Communication and Signal Processing NCSP'10, 2010, pp. 17-20.

[22] Z. Ahmed, "Genetic Algorithm for the Traveling Salesman Problem using Sequential Constructive Crossover Operator," Intr. J. of Biometric and Bioinformatics, Vol. 3, 2010, pp. 96-105.

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.54 e-ISSN: 2251-7545

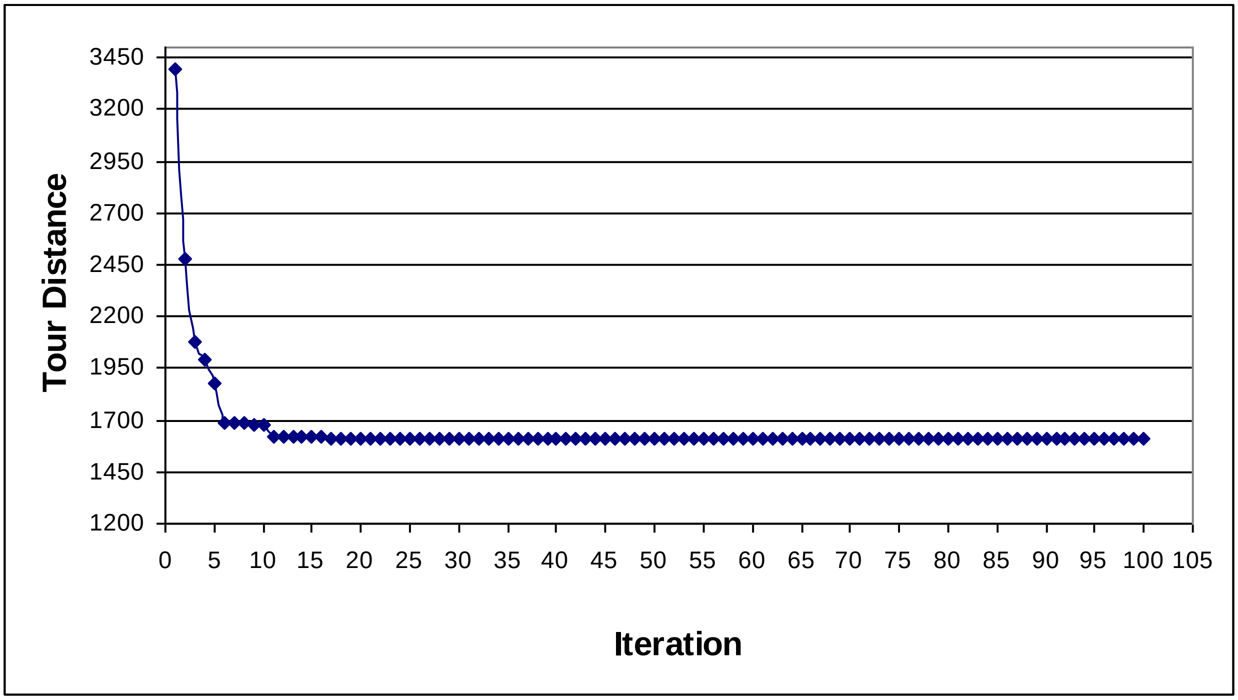

Figure 5. Convergence of Electimize toward Optimal Solution for the Bayg29 TSP

Best Tours Identified by Electimize for Bayg29

Tour	**Best Tours for Bayg 29**																													**Distance**	**% Error**
I	1	28	6	12	9	26	3	29	5	21	2	20	10	4	15	18	14	17	22	11	19	25	7	23	8	27	16	13	24	1	1610	0.00%
II	1	28	6	12	9	26	3	29	5	21	2	20	10	4	15	18	14	17	22	11	19	25	7	23	27	16	13	24	8	1	1615	0.31%
III	1	28	6	12	9	26	3	29	5	21	2	20	10	4	15	18	14	17	22	11	19	25	7	23	27	8	24	16	13	1	1620	0.62%
IV	1	28	12	6	9	26	3	29	5	21	2	20	10	4	15	18	14	17	22	11	19	25	7	23	27	16	13	24	8	1	1622	0.75%

TABLE I. BEST TOUR VALUES IDENTIFIED BY DIFFERENT ALGORITHM FOR THE BAYG29 TSP

Algorithm	**Best Solution**	**% Error**
Electimize	1,610	0.00%
IDACO	1,637	1.68%
ACO	1,650	2.48%
GAs (SCX)	1,610	0.00%
GAs	1,759	9.25%