U.Porto Journal of Engineering, 1:2 (2015) 30-38
ISSN 2183-6493
DOI 10.24840/2183-6493_001.002_0004

Received: 16 January, 2015
Accepted: 6 March, 2015
Published: 1 December, 2015

Basic Evolutionary Approach to the Traveling Salesman Problem

Débora Regina de São José[1], Mauricio Garcia Hernandez[2]
[1]Department of Industrial Engineering and Management, Faculty of Engineering University of Porto, Porto, Portugal (up201106063@fe.up.pt); [2]Department of Electrical and Computer Engineering Faculty of Engineering University of Porto, Portugal (up201400040@fe.up.pt)

Abstract
Evolutionary programming (EP) is a metaheuristic method developed as an alternative approach to artificial intelligence. The aim of this paper is to bring an introduction to EP algorithms through the implementation of the basic D. B. Fogel's Evolutionary Programing approach of 1988 and the emulation of his results in order to analyze the performance of the evolutionary programming method on solving a benchmark test case. The EP approach is implemented thru a simple simulation of natural evolution and the allowance of probabilistic survival of individuals. The novelty of this paper relies on testing the algorithm performance in some problems of well-known benchmark instances of the Traveling Salesman Problem, where that 1988 evolutionary approach was not tested. The reproduction of 1988 D. B. Fogel's approach was possible, the found average error of this method for 200000 offspring applied to the benchmark instances was found to be in the order of the 10%.

Subject Headings. Travelling Salesman, Optimization, Operational Research.
Author Keywords. TSP, Travelling Salesman Problem, Mutation Operator, Evolutionary, Programming, Evolutionary Computation.

1. Introduction

1.1. Travelling Salesman/Salesperson Problem (TSP)

The Travelling Salesman Problem (TSP) for n cities can be defined as the problem of finding a tour, visiting all the cities exactly once and returning to the starting city, such that the sum of the distances between consecutive cities (or some other cost function) is minimized (Fritzsche 2007; Fogel 1988; Fogel and Fogel 1996).

Furthermore, a TSP problem can, also, be considered symmetric if in the group of all n cities required to visit, the distance between cities i and j is the same as between cities j and i (Balachandar and Kannan 2007).

Considering that any city has a path to any other and that the distance or cost of the tour from city A to city B is the same than from city B to city A to every pair of cities in the problem, the problem will provide (n - 1)!/2 alternative tours (Michalewicz and Fogel 2004).

The choose of the problem was based on the importance of the TSP, according to Fogel and Fogel (1996) "the problem is interesting because (1) it has become a benchmark test case, (2) it is easily stated yet difficult to solve, and (3) it is broadly applicable to a number of routing and networking problems".

The solution to the problem is represented by any list of all the cities to be visited, e.g. for a small problem with five cities a representation could be the ordered list of cities "A-B-C-D-E" (Fogel 1988). For this example the solutions will be all the possible permutations of cities to

be visited in the list, assuming it as a symmetric traveling salesman problem, the number of possible solutions will be (5 - 1)!/2=12, this can looks like an easy problem, furthermore, according to Michalewicz and Fogel (2004, 13–14) the number of possible solutions rapidly increases when increasing the number of cities.

The data that were used to apply the method belong to TSPLIB, it is an open library of sample instances for the TSP from various sources and of various types, instances of problem cases such as the symmetric version of the Traveling Salesman Problem are available. It is provided by the Institut für Angewandte Mathematik of Heidelberg University.

1.2. Evolutionary Programming

Evolutionary computation has, according to Fogel and Chellapilla (1998), three main lines of investigation: (1) genetic algorithms, (2) evolution strategies, and (3) evolutionary programming. Fogel (1995) stated that "the elements in a simulated evolving population are considered to be analogous to species in evolutionary programming, individuals in evolution strategies, and chromosomes and genes in genetic algorithms", these concepts are summarized in Figure 1.

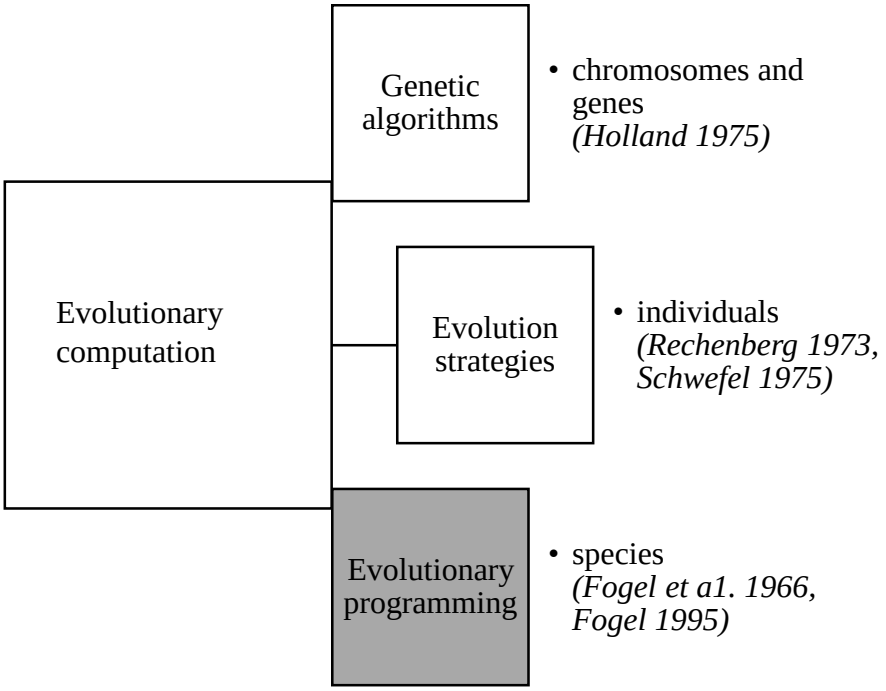

Figure 1: Evolutionary Computation's Tree

Evolutionary programming is a metaheuristic method developed by Lawrence J. Fogel while serving the National Science Foundation in 1960 (Fogel and Chellapilla 1998). According to Fogel and Fogel (1996) "the original motivation for this method was to generate an alternative approach to artificial intelligence", it "was modelled as a process that generates organisms of increasing intellect over time".

For this method, Fogel and Fogel (1996) defined the general steps as follows:

(1) Create an initial population at random (P);
(2) Evaluate the initial population;
(3) Create an offspring from this population (usually one offspring per parent) by random mutation (P');
(4) Evaluate P' using the same evaluation function that was used to evaluate P;
(5) Combine P and P';

(6) Conducted a stochastic competition including all parents and offspring;
(7) Select the winner solutions (50% of the combined population) to generate the next population.

Therefore, the pseudo code for this problem is described in Figure 2.

```
procedure evolutionary programming
begin
      t ← 0
      initialize P(t)
      evaluate P(t)
      while (not termination-condition) do
      begin
            create an offspring population P'(t) using mutation
            operator
            evaluate P'(t)
            Q(t) ← P(t)+P'(t)
            P(t+1) ← the half part of Q(t) (50% of the solutions) with
                     the better evaluation after stochastic competition
            t ← t +1
      end
end
```

Figure 2: Evolutionary Programing Pseudocode

According to Fogel and Fogel (1996), attention to two facets should be paid while creating an appropriate mutation operation: (1) it must maintain a strong behavioral link between each parent and its offspring, and (2) it must provide a (nearly) continuous range of potential new behaviors.

EP generates machine learning through automated discovery. As stated in section 1.1, the number of calculations (steps) required to solve the TSP grows at least exponentially with the amount of elements in the problem, Evolutionary Programing allows to obtain a solution by only examining a small fraction of the total number of tours examined (Fogel 1988).

This paper's goal is to bring an introduction to EP algorithms through the implementation of the basic D. B. Fogel's Evolutionary Programing approach of 1988 and the emulation of his results and it is organized as follows. In Section 2, a description of the parameters and specifications selected for the EP algorithm is provided. In Section 3, results of computational experiments are given, evolution of the current best solution provided by the EP algorithm is plotted, and quality of the tour is evaluated. Finally, Section 4 includes the concluding remarks.

2. Materials and Methods

The algorithm parameters were chosen in order to follow the implementation used by Fogel in 1988 and described as follow:

Initialization: An initial population of n parent tours is created at random. Where n is the number of cities to be visited in the tour problem.

Evaluation of parents: Euclidean length stands for measuring every parent tour quality. Equation 1 allows to compute Euclidean distance in a two dimensional space between city A and B as follows:

$$D(A,B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

Notice that distance measured from city A to city B is the same as distance measured form city B to city A.

Creation of an offspring population: Offspring are then created through random mutation of each parent by selecting a city in the parent's list and replacing it in a different, randomly chosen position.

Evaluation of offspring: Quality of generated offspring is evaluated, the same as parents quality, by equation 1.

Stochastic competition of population: Whole population is compound by parents and offspring. Every tour faces direct competition against a randomly selected 10% of the other tours. The probability of every current tour having a win after competing against one of the selected 10% of the other tours is computed by a fitness function as shown in equation 2.

$$Probability\ of\ wining = \frac{Competing\ tour\ length}{Competing\ tour\ length + Current\ tour\ length} \quad (2)$$

Therefore, i.e. if a tour of length 700 competes with a tour of length 21000 the probability of the first tour obtaining a win is

$$Probability\ of\ wining = \frac{2100}{2100 + 700} = 0.75 \quad (3)$$

Selection of new parents: The 50% of the tours with the most win scores become the parents of the next generation.

Excluding initialization step, the listed procedure was allowed to be repeated 200000 times, thus to allow the generation of the 200000 offspring used by D. B. Fogel.

At every step best tour, length is recorded in order to generate the possibility of plotting the evolution of the algorithm performance through time.

The original Fogel's implementation used tours from 16 to 100 cities, thus to know the performance of this approach on benchmark instances, the closest TSPLIB problems in number of cities available were selected: wi29, dj38, eil51, berlin52, st70, pr76, eil76 and rd100 (TSPLIB; TSPLIB). These instances are symmetric datasets, and tours are compound by a range of cities going from 29 to 100. Equation 1 verifies the validity of testing symmetric datasets.

Referred evolutionary programing approach was implemented on a Pentium Dual-Core 1.7 GHz CPU with 4 GB of RAM memory.

3. Discussion

After allowing the generation of 200000 offspring to the evolutionary programing approach, in every one of the referred instances, the obtained results are shown in Table 1:

TSP	Optimal Tour	Achieved Tour	Relative Error	Time [s]
WI29	27603	28670	0.03865522	1827
DJ38	6656	7555	0.13506611	2884
EIL51	426	472	0.10798122	4751
BERLIN52	7542	8240	0.0925484	4878
ST70	675	798	0.18222222	8540
PR76	108159	123350	0.14045063	9525
EIL76	538	610	0.13382899	9547
RD100	7910	10035	26.86472819	13337
			Average: 0.13173308	

Table 1: Comparison between the achieved solution and optimal tour

Furthermore, the figures 4 to 11 reveal that a faithful emulation of Fogel's Evolutionary Optimization (Fogel, 1988) was achieved, due to their similarity with the graphics generated by his implementation. In addition, they reveal that the convergence rate slows down as the number of cities increases.

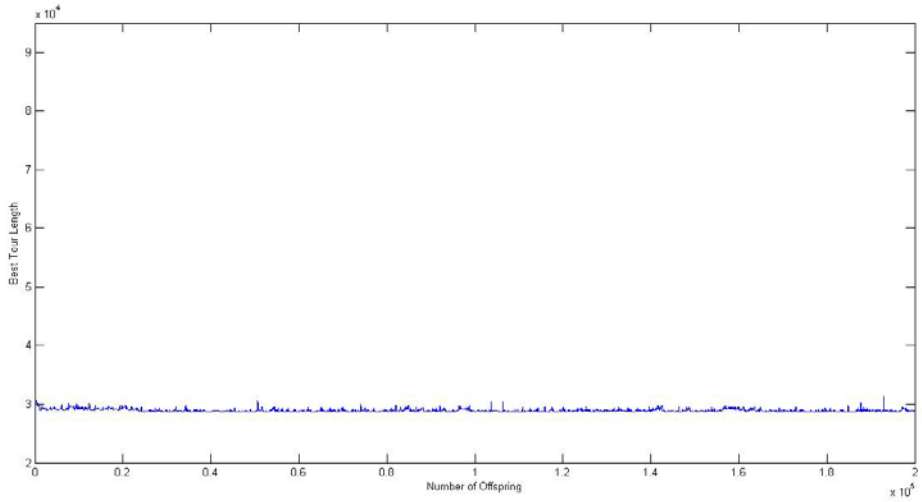

Figure 4: Evolutionary Optimization – WI29

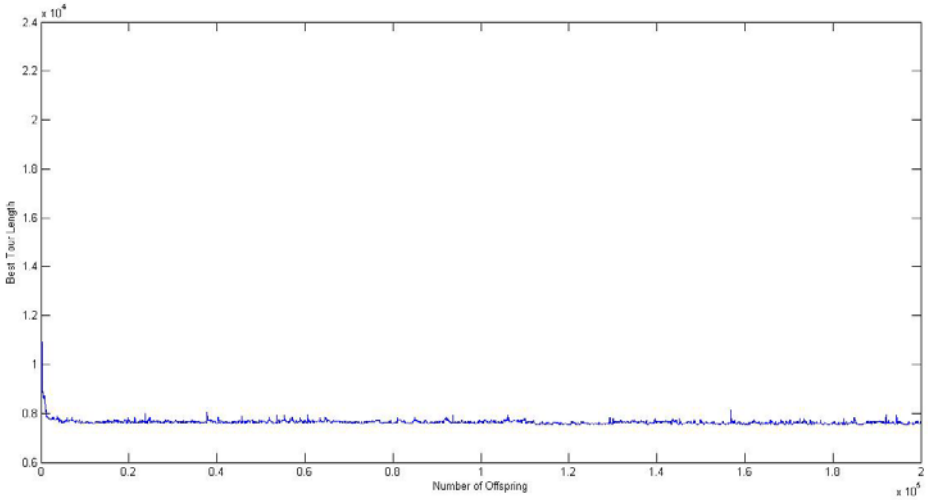

Figure 5: Evolutionary Optimization – DJ38

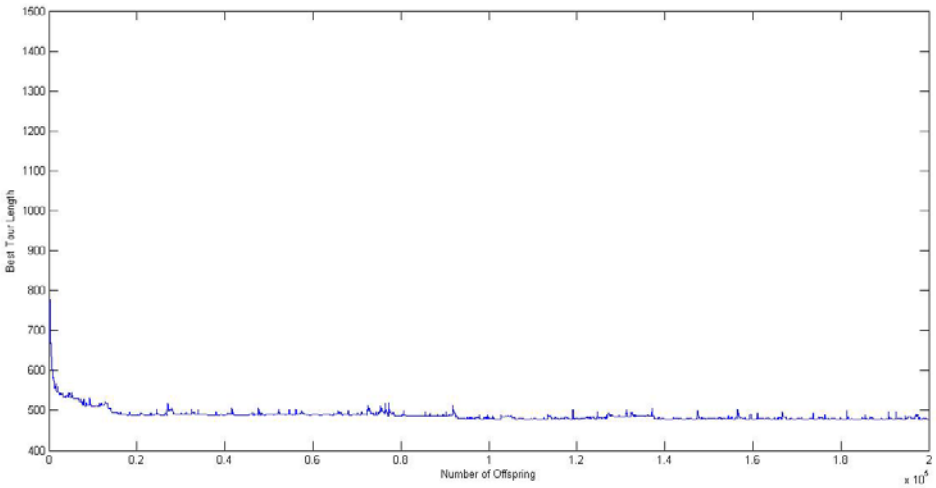

Figure 6: Evolutionary Optimization – EIL51

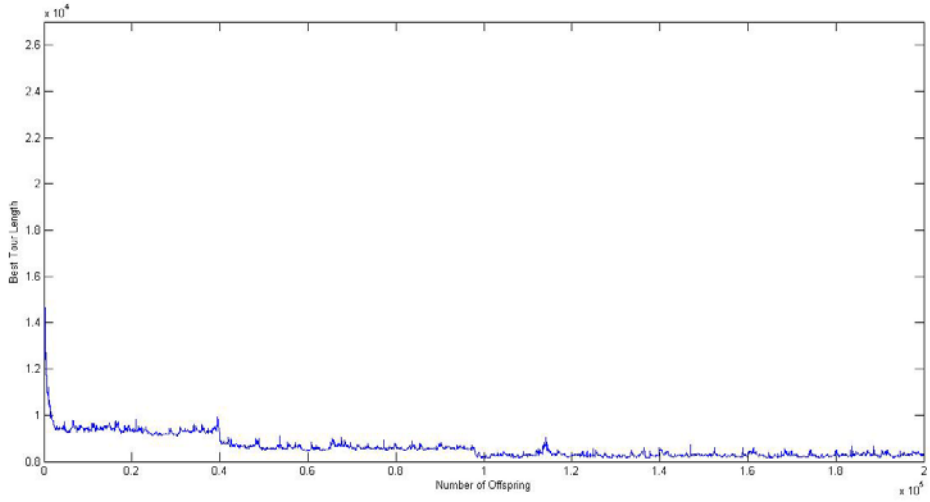

Figure 7: Evolutionary Optimization – BERLIN52

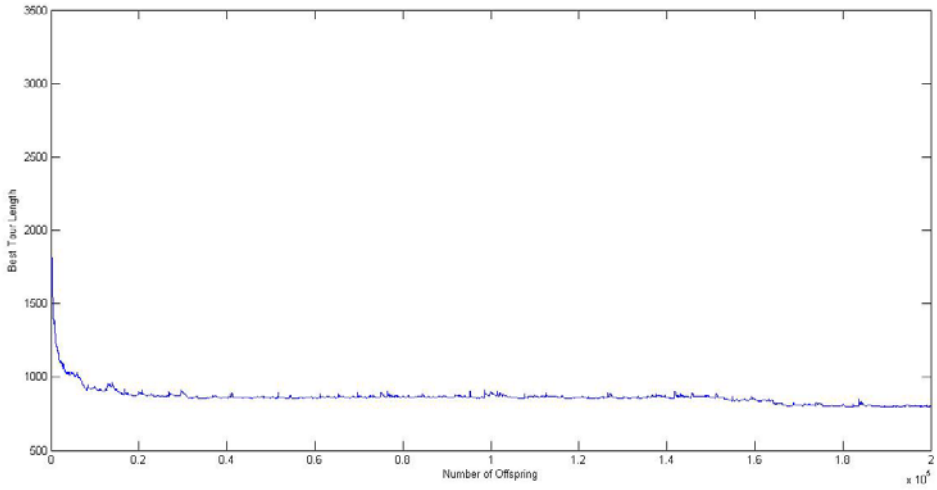

Figure 8: Evolutionary Optimization – ST70

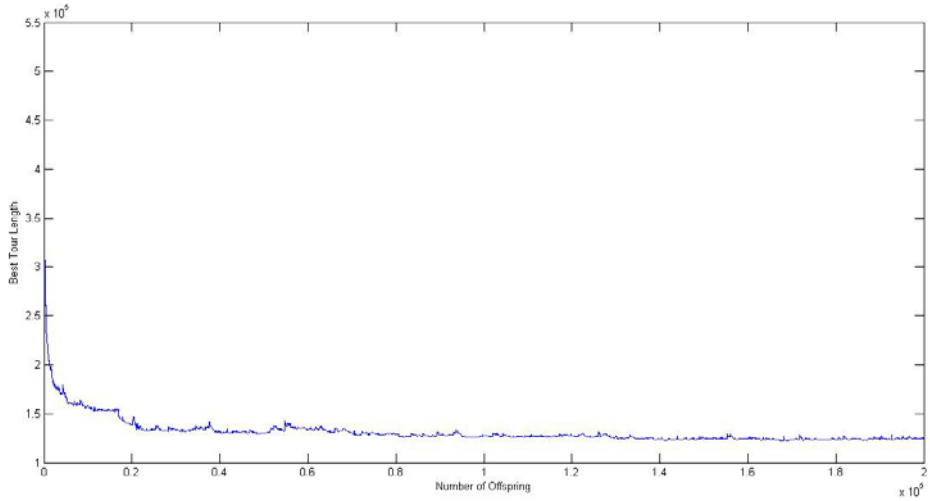

Figure 9: Evolutionary Optimization – PR76

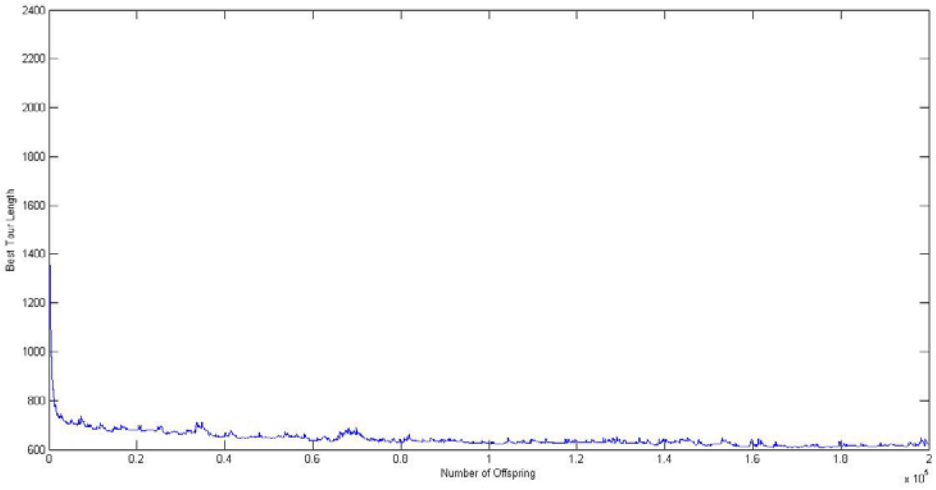

Figure 10: Evolutionary Optimization – EIL76

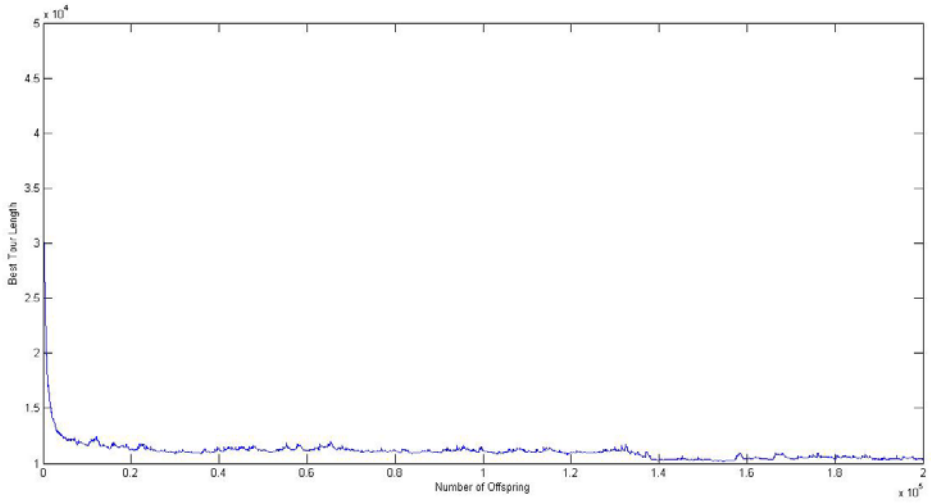

Figure 11: Evolutionary Optimization – RD100

Considering the simplicity of the implemented approach, on a 200000 offspring horizon the achieved tour remains on an acceptable range, since the average error of all the generated solutions is in the range of 10%.

Furthermore, the relative error shows that the effectiveness of the approach tends to diminish as the number of cities in the tour increments; however, this affirmation is only valid for a 200000 offspring generation. There is no reason to believe that better tour is not going to be achieved if more offspring generations is allowed, because, as shown in figures 4 to 11, and as demonstrated by Fogel (1988) the convergence rate of the EP tends to be logarithmic.

Additionally, as table 1 shows, time required for computation increments linearly as the number of cities grows up, due to the fact of the increment of computations required for every step of the cycle. For example, while on an instance of twenty-nine cities, every tour of an offspring of fifty-eight individuals competes against other three tours, an instance compounded by one hundred cities, generates the competition between every individual of an offspring of two-hundred tours versus other ten tours.

Finally, notice that the required computation increment is reflected in every step of the EP approach, not only in the provided example.

4. Conclusions

The main modelling ideas of the implemented approach can be summarized as follows: the stochastic competition of population, along with the selection of new parents represent the probabilistic survival of the simulated organisms. The inclusion of a mutation operator in this context represents a simplified way of modelling evolution of the current population, where fitness function accentuates the developmental appropriateness of the evolved solution.

The evolutionary computation community have been paying special attention to the TSP since 1980, and almost in every conference of evolutionary algorithms is still included papers on the TSP (Michalewicz and Fogel 2004). Understanding this fundamental approach helps to comprehend more sophisticated state of the art evolutionary approaches.

As studied in this work, even the method implemented is a basic one, it provides feasible/acceptable solutions for relatively small number of cities, and convergence time for this approach increases linearly as the number of cities grows up. Only some publications bring the computational time needed for solving specific instances of the TSP (Michalewicz and Fogel 2004), and convergence for this kind of approach tends to be slow (Reinelt 1994). Therefore, future work will consist in the use of combined state of the art approaches in order to generate a real idea of the EP performance in present context.

General performance of EP could be enhanced by improving different aspects of the algorithm. Some state of the art techniques which could be used to improve the EP performance may be (but are not limited to):

- Specific programing language tools such as parallel programing (The MathWorks 2015) and GPU programing (The MathWorks 2015) can improve convergence time, it also can be speeded up to 6 times by computing the fitness as reported by Xiong et al. (2011).
- Generation of better quality tours can be achieved by conducting population evolution through more sophisticated ways, i.e. distributed evolutionary strategies as reported by Huang et al. (2009) or Self-Adaptive Methods as reported by Chellapilla & Fogel (1998).

References

Balachandar, S. R., and K. Kannan. 2007. "Randomized gravitational emulation search algorithm for symmetric traveling salesman problem." *Applied Mathematics and Computation* no. 192 (2):413-421. Accessed January 16th, 2015. DOI: 10.1016/j.amc.2007.03.019.

Cook, William. 2009. "National Traveling Salesman Problems". Accessed January 16th, 2015. http://www.math.uwaterloo.ca/tsp/world/countries.html.

Fogel, D. B. 1988. "An evolutionary approach to the traveling salesman problem." *Biological Cybernetics* no. 60 (2):139-144. Accessed January 16th, 2015. DOI: 10.1007/bf00202901.

Fogel, D. B., and K. Chellapilla. 1998. "Revisiting evolutionary programming." In *Applications and Science of Computational Intelligence*, edited by S. K. Rogers, D. B. Fogel, J. C. Bezdek and B. Bosacchi, 2-11. Bellingham: Spie-Int Soc Optical Engineering. Accessed January 16th, 2015. DOI: 10.1117/12.304792.

Fogel, D. B., and L. J. Fogel. 1996. "An introduction to evolutionary programming." In *Artificial Evolution*, edited by J. M. Alliot, E. Lutton, E. Ronald, M. Schoenauer and D. Snyers, 21-33. Accessed January 16th, 2015. DOI: 10.1007/3-540-61108-8_28.

Fogel, D. B., and IEEE. 1995. *Phenotypes, genotypes, and operators in evolutionary computation*, 1995 IEEE International Conference on Evolutionary Computation, Vols 1 and 2. New York: IEEE.

Fritzsche, P. C., D. Rexachs, E. Luque, and IEEE. 2007. *TSP performance prediction using data mining*, Idaacs 2007: Proceedings of the 4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications.

Huang, S. Z., L. Zhu, F. Zhang, Y. X. He, and H. D. Xue. 2009. "Distributed evolutionary algorithms to TSP with ring topology." In *Computational Intelligence and Intelligent Systems*, edited by Z. H. Cai, Z. H. Li, Z. Kang and Y. Liu, 225-231. Accessed January 16th, 2015. DOI: 10.1007/978-3-642-04962-0_26.

Michalewicz, Zbigniew, and David B. Fogel. 2004. *How to solve it: Modern Heuristics*. Springer-Verlag Berlin Heidelberg. Accessed January 16th, 2015. DOI: 10.1007/978-3-662-07807-5.

Reinelt, Gerhard. 1994. *The Traveling Salesman: Computational Solutions for TSP Applications*. Springer-Verlag Berlin Heidelberg. Accessed January 16th, 2015. DOI: 10.1007/3-540-48661-5.

Skorobohatyj, Georg. 1995. "MP-TESTDATA - The TSPLIB Symmetric Traveling Salesman Problem Instances". Accessed January 16th, 2015. http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html.

The MathWorks, Inc. 2015. "GPU Capabilities and Performance". Accessed January 16th, 2015. http://www.mathworks.com/help/distcomp/gpu-capabilities-and-performance.html.

The MathWorks, Inc. 2015. "How Parallel Computing Products Run a Job". Accessed January 16th, 2015. http://www.mathworks.com/help/distcomp/how-parallel-computing-products-run-a-job.html.

Yan, Xiong, Zhou Wenyong, Wu Chang-an, Li Lei, and Liu Hongbing. 2011. TSP evolutionary algorithm based on fitness increment. Em *2011 Seventh International Conference on Natural Computation (ICNC)*. Accessed January 16th, 2015. DOI: 10.1109/ICNC.2011.6022549.

Acknowledgements

The authors would like to thank José F. Oliveira for his helpful comments and suggestions.

This work was supported in part by CONACYT México (Consejo Nacional de la Ciencia y la Tecnología) through “Formación de recursos humanos de alto nivel en el extranjero” program and for the ERASMUS MUNDUS through BE MUNDUS Project.