Evolutionary Divide and Conquer (II) for the TSP

Christine L. Valenzuela
Department of Computer Science, MS 4A5, George Mason University
Fairfax, VA 22030-4444
cvalenzu@cs.gmu.edu

Abstract

Results presented in recent papers demonstrate that it is possible to produce high quality solutions to TSP instances of up to several hundred cities using simple greedy heuristics when a Genetic Algorithm (GA) is used to perturb the city coordinates. The present paper extends the earlier studies to larger problems and a divide and conquer algorithm in the style of Richard Karp. Using a GA to perturb city coordinates in conjunction with a divide and conquer algorithm the feasibility of solving large problems to within a few percent of optimality is demonstrated. The exceptionally rapid execution times of Karp's algorithms together with their almost linear run-time scaling at $O(n\log n)$ set them apart from most other heuristic algorithms.

1 INTRODUCTION

Simple tour construction heuristic algorithms applied to the travelling salesman problem (TSP) tend to produce rather poor quality solutions about 20% above the optimum tour length. Results presented in recent papers, however, demonstrate that it is possible to produce high quality solutions to TSP instances of several hundred cities using simple greedy heuristics such as nearest neighbour (Bradwell 1997) and multifragment (Valenzuela 1997a) when a Genetic Algorithm (GA) is used to perturb the city coordinates. The present paper extends the earlier studies to a divide and conquer algorithm in the style of Richard Karp (Karp 1977). Using a GA to perturb city coordinates in conjunction with a divide and conquer algorithm the feasibility of solving large problems to within a few percent of optimality is demonstrated. The exceptionally rapid execution times of Karp's algorithms together with their almost linear run-time scaling at $O(n\log n)$ set them apart from most other heuristic algorithms. It is the impressive run-time scaling combined with the simplicity of the approach which motivates the study.

The TSP is probably the best known combinatorial optimization problem, and it has provided a challenging testbed for GA researchers for many years. Most authors have chosen a genetic representation based on permutations of cities, and several clever genetic operators have been devised which are capable of combining useful features from two parental tours into an offspring tour. Recent work by Nagata and Kobayashi (Nagata 1997) has produced some extremely impressive results using a crossover which they devised called edge assembly crossover.

Current versions of my algorithms are not competitive with state-of-the-art GAs for the TSP in terms of solution quality. Although results for a few hundred cities are impressive, the solution quality tends to 'drift' a little as the problems become larger. The large numbers of experiments needed to establish various parameters required for Karp's algorithms provides a serious barrier to progress. I believe, however, that the approach is worthy of perseverance and further study for the following reasons:

1) The scaling of Karp's algorithms at $O(n\log n)$ means the approach holds promise for very large instances of the TSP.

2) The simplicity of the approach: no complicated heuristics or operators are employed, yet the solution quality produced using Karp's algorithm is lifted from 20% excess to a few percent at most.

3) The encoding scheme is novel: most GAs for the TSP use representations based on permutations.

4) Finally, the divide and conquer paradigm has important parallels in nature, for example the DNA complement of a complex multicellular organism includes a blueprint for cellular differentiation: different cell groupings forming different organs to perform different functions in the organism.

Karp's algorithms subdivide a TSP instance in the Euclidean plane by recursively bisecting the space to produce smaller and smaller rectangles. Whenever the process encounters a rectangle containing fewer cities than some predetermined threshold, it stops bisecting the rectangles and solves the subproblems. The solved subproblems are patched together to produce a global tour through all the cities.

In the present study I use a greedy heuristic algorithm to solve the subproblems, and a very simple patching procedure to join the subproblems together in pairs as the recursive process unwinds.

An earlier approach which combined a GA with Karp's algorithm, which I shall refer to as *Evolutionary Divide and Conquer (I)* (*EDAC(I)*) (Valenzuela 1994, 1995), proved effective on instances of about 5,000 cities. Unfortunately Karp's algorithms in their original form typically produce solutions that are 20-30% above the optimal tour lengths (Johnson 1990). The *EDAC(I)* algorithms, however, are able to improve upon the solution quality quite considerably by using a genetic algorithm to explore the space of *problem subdivisions*, and by employing a range of *enhanced repair heuristics* in an attempt to correct the typical 'errors' that are inherent in Karp's divide and conquer scheme. Whilst the addition of enhanced repair heuristics would appear to maintain the near linear scaling qualities of the execution time for problems in the range of 500 - 5,000 cities, they unfortunately increase the actual run-time quite considerably. For example a 500 city problem runs about five times faster *without* the enhanced repair heuristics. Due to the excessive run-time requirements, the scaling qualities of *EDAC(I)* were not investigated beyond instances of 5,000 cities. Fortunately it would appear that the new approach based on perturbed coordinates does not need *any* repair heuristics and so it becomes feasible to explore its application to much larger instances of the TSP. In this preliminary paper the largest instance to which the new techniques have been applied is 5,000 cities. Work is currently underway, however, to extend this to much larger instances.

Weaknesses in many of the standard TSP heuristic algorithms are easy to spot by examining typical solutions produced by such techniques. Tour construction heuristic algorithms, for example, tend to start off well, adding short edges in the early stages of the construction process, but are often left only with very long edges from which to select the final portions of the tour. **Figure 1**shows a typical tour produced using a tour construction algorithm called the *greedy* (or *multi-fragment*) heuristic algorithm (see (Johnson 1990) for a survey of tour construction heuristics).

'Good parts' and 'bad parts' of the tour are clearly visible in **Figure 1**, illustrating the reduced effectiveness of the *greedy* heuristic when only a few cities remain to be incorporated in the tour. I use the multifragment algorithm in the present study to solve the subproblems in the rectangles generated by Karp's divide and conquer algorithm.

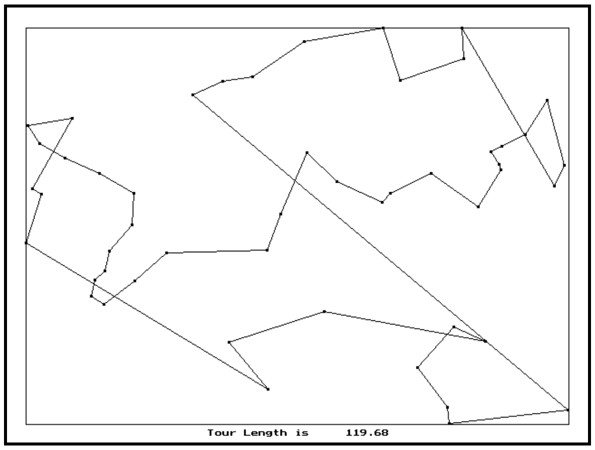

Figure 1 Typical solution to a 50 city problem produced by the greedy heuristic algorithm

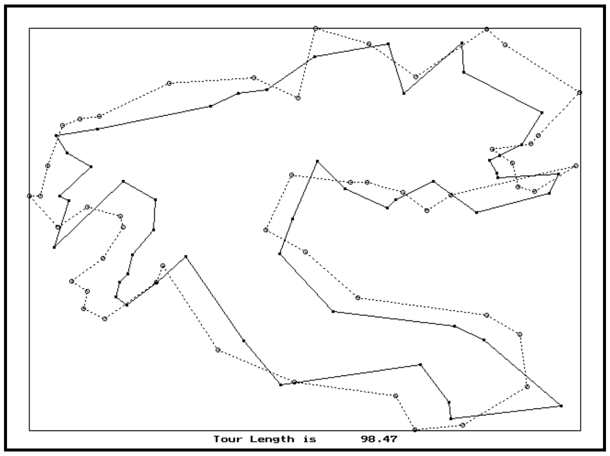

Figure 2 Improved solution obtained by perturbing the city coordinates of the 50 city problem

Figure 2 illustrates how the tour from **Figure 1** can be improved by running a GA to perturb the city coordinates to 'fool' the greedy heuristic algorithm. In **Figure 2** the best tour is denoted by solid lines, and the corresponding tour through the 'perturbed cities' is denoted by the broken lines. (Further details of the mutifragment study can be found in (Valenzuela 1997a)).

In the present study I use a genetic algorithm to perturb city coordinates for the TSP in such a way that a very simple divide and conquer algorithm is 'fooled' into producing excellent solutions. The use of perturbed coordinate sets for solving the TSP was first suggested by Codenotti, Manzini, Margara and Resta (Codenotti 1993, 1996) who incorporated the technique into their version of *iterated local search*. Perturbing the cities only to enable their algorithm to escape local optima, their approach remained primarily focussed on the original coordinate set. My technique, on the other hand, uses a GA to breed perturbed coordinate sets, and the heuristic algorithms are applied exclusively to these, the original coordinates serving only to evaluate intercity distances for computing tour lengths.

2 THE DIVIDE AND CONQUER PARADIGM FOR THE TSP

The divide and conquer algorithms pioneered by Richard Karp for the TSP (Karp 1977) give some useful guarantees of solution quality and time complexity.

Theorem (Karp 1977). *For every $\epsilon > 0$ there is an algorithm $A(\epsilon)$ such that $A(\epsilon)$ runs in time $C(\epsilon)n+O(n\log n)$ and, with probability 1, $A(\epsilon)$ produces a tour of length not more than $1+\epsilon$ times the length of a minimal tour.*

Although in practice the early implementations of this approach by Karp and others gave rather poor solutions, typically 30% excess, the probabilistic asymptotic guarantees of solution quality together with the time complexity of $O(n\log n)$ provides a promising starting point. The challenge is to develop a suitable genetic search procedure which, using a variation of Karp's algorithm as its heuristic engine, is able to improve the solution quality considerably and do so with a minimum of effort.

In essence Karp's algorithms repeatedly partition the rectangle holding the TSP problem until the whole region consists of a patchwork of small rectangles each containing about t cities. An exact or heuristic method is then applied to each subproblem and the resulting subtours are finally patched together to yield a single tour through all the cities.

Imagine a rectangle containing a TSP instance. If we wish to bisect this rectangle on area we can chose one of two directions of cut. Either we bisect in an horizontal direction or a vertical direction. Both produce two new rectangles, the former an upper and a lower rectangle, and the latter a left and a right rectangle. This process can be repeated for each of the two new rectangles, then applied recursively to the results. The choice a horizontal or a vertical cut has to be made each time.

I refer the reader to an earlier paper (Valenzuela 1994) for details of the various bisection techniques that were explored when *EDAC(I)* was developed. In the present study I use bisection method 3 from this paper, in which rectangles are bisected through the city nearest to the true area bisection line. The presence of a shared city between each pair of subsolutions means that it is possible to employ a very simple patching algorithm.

The patching algorithm used to join subtours together in pairs involves breaking the four edges incident with the 'shared city' on the boundary of the two adjacent rectangles, two edges in each of the rectangles. Once these links have been severed, there are four different ways in which links can be reconnected to form a single tour through all the cities contained in the two adjacent rectangles. All four ways are tried and the set of links which results in the shortest tour is the one chosen.

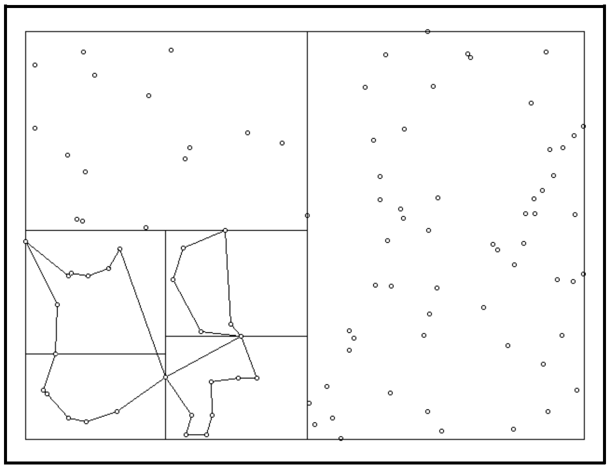
Figure 3 100 city problem in the early stages of bisection.

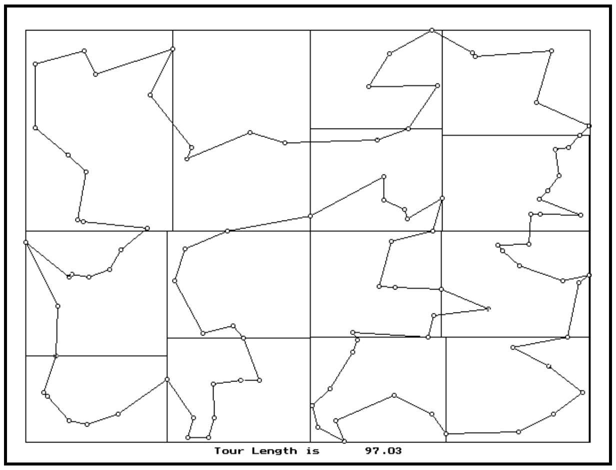

Figure 4 Completed solution of 100 city problem.

Figure 3 and **Figure 4** depict stages in the recursive bisection of a 100 city TSP instance and subsequent construction of the global tour. The direction of cut is always parallel to the shorter side of the bisected rectangle.

Karp showed that by minimizing the lengths of the perimeters of the rectangles in this way it was possible to minimize the expected lengths of the tours. In the *EDAC(I)* algorithms the decision to make horizontal or vertical cuts was made by the genetic algorithm. In the present study Karp's original criterion for bisection is adhered to and the genetic algorithm is used to perturb the city coordinates in order to 'fool' the algorithm into eliminating the typical errors inherent in Karp's approach.

3 THE METHOD OF PERTURBED COORDINATES

The basic idea is to perturb the city coordinates slightly, and use these perturbed coordinates to produce a tour using the chosen heuristic algorithm. The cities in the permutation list resulting from this tour are then moved back to their original positions and a 'true tour' is produced.

As mentioned above Codenotti *et al* incorporated a similar technique into their version of iterated local search, randomly perturbing the city coordinates of the TSP instance I by small amounts to give a new instance I' every time a locally optimal tour, T, was found on I. T will not normally be locally optimum with respect to I', so local optimization is then performed with respect to I' to give a new tour T'. T' then provides a new starting point for locally optimizing the TSP with respect to I. In this way the perturbed coordinates provide a simple 'mutation' enabling the local search algorithm to escape local optima. About half of the time is spent applying the heuristic algorithm to the original coordinates and half to the perturbed coordinates.

The present approach differs from that of Codenotti *et al* in three very important ways:

- In the present study the TSP heuristic is applied *only* to perturbed coordinate sets.
- A genetic algorithm is used to breed perturbed coordinate sets.
- The present approach can be applied to tour construction heuristics as well as to local search heuristics.

The perturbed coordinate sets are the chromosomes in these experiments, and Karp's algorithm produces an individual tour for each of the perturbed coordinate sets in the population at any one time. These tours can then be evaluated with respect to the original city coordinates and actual intercity distances. The 'true' tour lengths form the basis for the fitness function of the GA. All the TSP instances that I use here are uniform random points in a square region of the Euclidean plane. These problems are deliberately chosen in favour of standard problems from the literature (which often consist of pathologically distributed cities in straight lines or clusters) at this stage because it is vital that the distributions of cities should vary in a predictable manner with increasing problem size/city density. Such properties are essential if we are to attempt to draw any general conclusions about the algorithms with regard to solution quality or produce a formula for scaling the perturbation zone (see later).

4 THE GENETIC ALGORITHM

The genetic algorithm used in this study is a very simple GA based on (Holland 1975) and uses Cavicchio's preselection paradigm (Cavicchio 1970). Although it is very similar to the GA described in the earlier papers on breeding perturbed coordinate sets (Bradwell 1997) (Valenzuela 1997), in which an offspring either replaces a weaker parent or dies, I found that the inclusion of some simple selection probabilities (suggested by Gorges-Schleuter (Gorges-Schleuter 1990)) based on a population ranked for fitness gave slightly better results when used for mate selection than did selection of mates from the uniform random distribution used previously.

Selecting individuals for reproduction on the basis of rank is a nonparametric procedure first explored by Baker (Baker 1985). It involves first sorting the population according to the objective function. In this scheme we then assign the individual producing the *worse* phenotype (tour) a score of 1, and the individual producing the *best* phenotype a value of N, where N is the population size. All other individuals are assigned integer scores in between such that the *second worse* receives a score of *2* and the *second best* a score of (N-1) etc. The scores are then converted into selection probabilities by dividing each score by the sum of all the scores for the population.

The chromosomal representation used is a string of (x, y) coordinates. A two-point crossover is used and the coordinate pairs are sequenced according to the ordering of cities in the best tour found so far. The process for generating an initial population of perturbed coordinates is essentially the same as we employed in (Bradwell 1997) and (Valenzuela 1997a). In this scheme the city coordinates are perturbed within a small, uniform preset rectangular region surrounding each city, letting a suitable size for this rectangular region be determined by some early experimentation. Mutation is applied to randomly selected x or y coordinates by perturbing them within the small perturbation region surrounding each city. However, in the case of mutation the points defined by the *virtual* coordinate sets are considered to be located centrally in the perturbation rectangles, in favour of the points defined by the real coordinate sets.

5 SCALING THE PERTURBATION ZONE

By how much should the cities be allowed to 'move' when the coordinates are perturbed? Applying simple heuristic algorithms, such as the greedy algorithm, to random uniform points, the size of the ideal perturbation zone is likely to be related to *R*, the area of the (square) region containing all the cities, and the problem size, *n*. More formally if *R* represents the area of the region, and *n* the size of the problem then,

$$l = k\frac{R^p}{n^q} \qquad (1)$$

relates the required dimension of the side of the perturbation zone, *l*, to *R* and *n* through a constant, *k*. Earlier investigations (Valenzuela 1997a) suggested values of $p = ½$ $q = 3/2$ and $k = 60$ gave the best results for our GA when using the multifragment heuristic algorithm

New investigations are required in order to establish a suitable regime for scaling the perturbation zone for Karp's algorithm. The additional effect of varying the average subproblem size, *t*, also needs to be taken into consideration. Thorough empirical investigations of this nature are bound to be extremely time consuming, so a rather less demanding approach is taken for this initial study.

Some initial execution time measurements suggest values of *t* between 15 and 50 to be reasonable as the run-times for my version of Karp's algorithm are found to vary very little when using subproblem sizes within this range.

Intuitively values of $p = ½$, $q = ½$ in equation (1) would seem appropriate for Karp's algorithm when *t* is kept constant, as these values would relate the linear proportion of the perturbation zone directly to the inverse of the city density. At least this scaling would seem applicable to the solution of the subproblems, how it would relate to the patching algorithm is rather less clear.

Full details of the experiments are omitted from the paper to save space, but, in summary, the choice of $t = 40$, $p = ½$, $q = ½$ produces good results for a 100 city problem with $k = 0.6$. As the problem size is increased, however, better results are produced with lower values for *k*. I use k = 0.6, 0.4, 0.3, 0.2 and 0.1 for problems of size 200, 500, 1000, 2000 and 5000 cities respectively. It would appear that the ideal value for *k* converges with increasing problem size, although more extensive studies are needed to establish this with certainty.

4 RESULTS

Table 1 Results for the new GA on random uniform points

Problem size	HK lower bound	Karp's algorithm	New GA (*EDAC(II)*)	# generations	*EDAC(I)*
100	96.37	109.58	96.7	243	96.8
200	99.04	121.85	101.9	394	102.3
500	96.01	122.54	100.9	560	100.6
1,000	95.22	124.21	101.9	1095	99.6
2,000	94.30	124.33	101.3*	2148	98.7
5,000	93.59	126.50	(108.8*)	(633)	99.1

* denotes single run, ( ) indicates incomplete run.

Some results for the new genetic algorithm on instances of random uniform points are presented in **Table 1**. The test instances vary in size between 100 and 5,000 cities (column 1). The Held-Karp lower bound (Held 1970, 1971) in column 2 represents a problem specific under-estimate of the optimal tour lengths. The value of the lower bound is estimated to be 0.08 % below the true optimum, on average, see (Johnson 1996) and (Valenzuela 1997b). The tour length produced running Karp's algorithm on the original city positions is tabulated in column 3, and the best tour lengths produced by running the new genetic algorithm can be found in column 4. The best tour lengths are averaged over five runs except where otherwise indicated. The result for 5000cities is enclosed in brackets because the experiment was incomplete at the time of going to press. Population sizes of 1,000 are used and mutation rates of 0.01 for problems of 100 - 500 cities, and 0.005 for problems of 1,000 and larger. The number of generations the GA is allowed to run is presented in column 5. The GA is halted when it has run for 200 consecutive generations with no new improvement found in the best tour. The average values of tour lengths obtained with the most sophisticated version of *EDAC(I)* (with heuristics based on the *3-opt* algorithm) on the same instances can be found in column 6.

5 CONCLUSIONS

My simple genetic algorithm which perturbs city coordinates (*EDAC(II)*) has succeeded in lifting the solution quality produced by an implementation of Karp's algorithm by about 20%. I believe this approach holds promise for very large instances of the TSP (for 100,000 cities and beyond) because Karp's algorithms scale at $O(n\log n)$. Although *EDAC(II)* is not quite able to match the results of

EDAC(I) as yet for larger instances of the TSP, I believe that some parameter tuning will easily fill this gap. *EDAC(II)*, unlike its predecessor, uses no repair heuristics and the scaling properties of Karp's heuristic engine are thus guaranteed at $O(n\log n)$ for *all* values of n.

More generally it is possible that the idea of combining a genetic algorithm with a divide and conquer paradigm is potentially useful in other problem domains where run-time scaling properties are an issue.

Acknowledgments

I should like to thank the University of Teesside for their support for this work.

References

(Baker 1985) J. E. Baker. *Adaptive selection methods for genetic algorithms*. Proceedings of an International Conference on Genetic Algorithms and Their Applications, Hillsdale, N.J: Lawrence Erlbaum Associates, 101 - 111.

(Bradwell 1997) R. A. Bradwell, L. P. Williams and Christine L. Valenzuela. *Breeding Perturbed City Coordinates and `Fooling' a Travelling Salesman Heuristic Algorithm*. Third International Conference on Artificial Neural Networks and Genetic Algorithms, (ICANNGA97) Norwich, 2 - 4 April 1997, pp 241-249, Springer Verlag.

(Codenotti 1993) B. Codenotti, G. Manzini, L. Margara and G Resta. *Global strategies for augmenting the efficiency of TSP heuristics*. Proc. 3rd Workshop on Algorithms and Data Structures, Lecture Notes in Computer Science, Vol. 709, Springer-Verlag, Berlin, 1993, 253-264.

(Codenotti 1996) B. Codenotti, G. Manzini, L. Margara and G Resta. *Perturbation: an efficient technique for the solution of very large instances of the Euclidean TSP*. IFORMS Journal on Computing, Volume 8, Number 2, spring 1996. pp 125.

(Cavicchio 1970) D.J. Cavicchio. *Adaptive search using simulated evolution*. Unpublished doctoral dissertation, University of Michigan, Ann Arbor.

(Gorges-Schleuter 1990) Martina Gorges-Schleuter. *Genetic Algorithms and Population Structures, A Massively Parallel Algorithm*. Ph.D. Thesis, Department of Computer Science, University of Dortmund, Germany.

(Held 1970) M. Held and R. M. Karp. *The travelling salesman problem and minimum spanning trees.* Operations Research **18**:1138-1162.

(Held 1971) M. Held and R. M. Karp. *The travelling* salesman problem and minimum spanning trees: part II. Mathematical Programming **1**:6-25.

(Holland 1975) J.H. Holland. *Adaptation in natural and artificial systems*. Ann Arbor:The University of Michigan Press.

(Johnson 1990) D.S Johnson. *Local Optimization and the Travelling Salesman Problem*. Automata Languages and Programming: 17th International Colloquium Proceedings. 1990.

(Johnson 1996) D. S. Johnson, L. A. McGeoch and E. E. Rothberg. *Asymptotic experimental analysis for the Held-Karp travelling salesman bound.* Proceeding 1996 ACM-SIAM symp. on Discrete Algorithms.

(Karp 1977) R. M. Karp. *Probabilistic analysis of partitioning algorithms for the Traveling-Salesman Problem in the plane*. Mathematics of Operations Research, **2**(3):209-224, August 1977.

(Nagata 1997) Yuichi Nagata and Shigenobu Kobayashi. *Edge assembly crossover: A high power genetic algorithm for the travelling salesman problem*. In T. Bäck, editor, Proceedings of the 7th International Conference on Genetic Algorithms, pp 450-457. Morgan Kaufmann 1997.

(Valenzuela 1994) Christine L. Valenzuela and Antonia J. Jones. *Evolutionary Divide and Conquer (I):a Novel Genetic approach to the TSP.* Evolutionary Computation 1(4): 313-333. MIT Press.

(Valenzuela 1995) Christine L. Valenzuela. *Evolutionary Divide and Conquer: a Novel Genetic approach to the TSP*. Ph. D thesis. University of London 1995.

(Valenzuela 1997a) Christine L. Valenzuela and L. P. Williams. *Improving Simple Heuristic Algorithms for the Travelling Salesman Problem using a Genetic Algorithm*. Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97), pp 458-464. Morgan Kaufmann, San Francisco, CA.

(Valenzuela 1997b) Christine L. Valenzuela and Antonia J. Jones. *Estimating the Held-Karp lower bound for the geometric TSP.* European Journal of Operational Research 102(1): 157-175, October 1997.