A NEW OPERATOR FOR EFFICIENT EVOLUTIONARY SOLUTIONS TO THE TRAVELLING SALESMAN PROBLEM

GEORGE G. MITCHELL DIARMUID O'DONOGHUE ADRIAN TRENAMAN

Department of Computer Science,
National University of Ireland, Maynooth.
Maynooth,
Co. Kildare,
Ireland.
{georgem, dod , trenaman} @cs.may.ie

ABSTRACT

In this paper we present two sets of empirical data evaluating the performance of a new *Cleanup* operator for evolutionary approaches to the travelling salesman problem (TSP). For raw data we have used standard road mileage charts of the USA, Great Britain and Ireland, which enable us to generate a reference table with appropriate city to city distances. A wide variety of standard genetic parameters (population size, epochs, mutation rate and selection type) is explored, and results allow the comparison of performance both with and without our cleanup operator. The cleanup operator improves the convergence speed by reducing the number of epochs required to identify a near-optimal tour; in each instance a significant reduction is convergence time was observed.

Our empirical observations show that assisting the evolutionary operators through the use of cleanup gives better performance on this evolutionary encoding. The implication of these findings run contrary to the apparent consensus towards a reduction in the number of genetic operators required [1] by a genetic system for the TSP [2,3]. In these works Fogel concluded that mutation alone was sufficient for this encoding of the TSP problem, rejecting the crossover operator because of its tendency to introduce invalid tours into the population. We have shown that by using the cleanup operator in conjunction with crossover we can effect a more efficient search than a solely mutation-driven approach.

KEYWORDS

Evolutionary Algorithms, Optimisation, Travelling Salesman Problem.

INTRODUCTION

Evolutionary algorithms are search techniques based on natural evolutionary systems, where the fittest individuals survive longer than the rest and produce more offspring. In a similar fashion, a population of solutions in an evolutionary algorithm moves in the direction of fitter solutions. In the case of the travelling salesman problem this means that the shortest tours are given a greater chance of being allowed to survive longer and produce more tours thereby favouring shorter routes.

The Genetic Algorithm (GA) as originally proposed and implemented by Holland [4] and also by Goldberg [5] consists of a string of bits representing *individuals*, which are then evolved by the application of successive actions of *crossover*, *inversion* and *mutation*.

It has been used to solve a wide range of different problems from vision systems for image classification to gas pipeline flow control systems [5].

The travelling salesman problem is a well-known member of the NP-Complete class of problems [6]. Over the last number of years it has returned to prominence with research conducted on providing near optimal solutions to large TS problems by Fogel [7, 3] and Lawler [8].

Although these GA's have undergone some alterations, in particular to the techniques of the *crossover, selection* [9] and the *mutation* operators [7], they have been shown to result in close to minimal distance TSP tour lengths. These systems have relied on the use of fully connected graphs as the method for deriving the distance between cities, (nodes). They have also relied on the use of some form of weighting to negate invalid city tours, or through the use of the implementation of a single genetic operator, such as repeated large-scale mutation, this of course maintains tour structure.

As has been noted by others who have attempted to provide solutions to the TSP using evolutionary means, the usual genetic operators of crossover, reproduction and mutation on binary strings are insufficient to solve the TSP. As Mitchell [10] points out "*some types of encoding require specially defined crossover and mutation operators for example the tree encoding used in the genetic programming, or encoding for problems like the Travelling Salesman Problem in which the task is to find a correct ordering for a collection of objects.* It is from here that we began to explore the possibilities of developing standard crossover and mutation operators that work well with the encoding scheme that we have used.

We present a solution to real world travelling salesman problems, that we have accomplished through the use of a more restrictive method for calculating distance between cities and through the introduction of a new genetic operator. This new operator *Cleanup* has been specifically designed for use in real world evolutionary TSP systems. Using *roulette wheel* selection and a combination of standard/normalised fitness together with varying mutation rates, crossover rates and different numbers of generations, it has been possible to significantly reduce the length of TSP tours.

RELATED RESEARCH

A number of differing search techniques have been applied to the travelling salesman problem, depth first search, hill climbing and neural networks [8, 11] all with varying degrees of success.

Fogel implemented a genetic solution that he called "An evolutionary approach to the travelling salesman problem". In this he proposed that as an alternative to all of the genetic operators which Holland [4] proposed in 1979, the emphasis should be on the behavioural appropriateness of the evolved trial solutions.

Tours were constructed as strings of cities, and the initial population was set at fifty. Evolution was driven by a single mutation operation. This mutation operation was loosely modelled on L. Fogel's "*Evolutionary Programming restricted to single state machines*"[7]. A small alteration of the existing tour was accomplished by selecting a city along the tour and then swapping this city with another city that had also been randomly chosen. By use of multiple mutation operations on the population Fogel was led to believe that there was no difference between widespread mutation and the use of a crossover operation. Fogel concluded that his approach, making use of mutation solely, created a random search for new offspring in the vicinity of the parents. This was in partial contradiction to the hypothesis of Holland which stated that "*if successive populations are produced by mutation alone, the result is a random sequence of structures drawn from (all possible structures)*"[4].

Fogel introduced population reduction to simulate the scarcity of resources as time proceeds. This was implemented by reducing the population size by 1 every 5000 offspring evaluations, that is, equivalent to every 100 generations.

Fogel concluded that simple mutation provided a better end result, since if a dramatic difference in the link between parent and offspring were permitted, the result may well be worse than a random search of all the coding structures.

Fogel later implemented another genetic solution in which he introduced a different approach to mutation and enhanced his simulation of behavioural methods. The tour was generated in the same manner as before. The population was set at one hundred tours, double the previous population. Each of the tours in the population produced a new offspring through mutation. The best one hundred individuals from each generation were selected for the next generation.

Here Fogel used a mutation strategy based on inversion. Two points are selected at random along a tour and the sub-string between the points is reversed. To make the evolutionary algorithm simulate natural systems even more closely Fogel designed a new technique of reducing the length of the inversion string in the mutation operation, linearly over time to some minimum as the number of generations reached their maximum. This he claimed simulated the decrease in behavioural difference across generations that is visible in natural systems, as they become better predictors of their environment.

The results found by Fogel were that for 30, 50 and 75 city tours his genetic algorithm found solutions which were better or at worst matched the previous best known tour length. On a trial 1000 city tour the result was found to be less than 5% to 7% worse than the previous best tour length.

THE PROBLEM OF TOUR VALIDATION

It is important to note the distinction between representational conventions of evolutionary algorithms for the travelling salesman problem. In some approaches, tours are directly related to the cities in a distance chart and in others they are constructed from a random graph. Regardless of the data, both approaches experience the problem of invalid tours. A tour is invalid in the TSP when a city is visited more than once.

One solution has been the PMX *partially matched crossover system* [12]. This technique reduces the effect of crossover by matching a section in each individual and then performing a limited crossover. This technique is restrictive and results have only been published for 10 and 33 city tours. It is for this reason that a new form of check is required so as to manage the calculation of solutions for more significantly size tours.

DATA

The evolutionary system used here makes use of real world distance data in the form of inter-city mileage charts. The charts in these experiments were for the USA, UK and Ireland. The coding for each tour was constructed as a list of cities represented by an integer value. The city-city routing distance was simply calculated by selecting the two appropriate cities and then cross-referencing to determine the distance, this then was used in the calculation of the raw fitness value for each individual tour. The use of this data source also enables us to permit travel between two cities in a particular direction but not in the reverse direction, as in the airline industry and 1-way streets. This differs significantly from the type of routing data normally used in Travelling Salesman solutions such the graph data used by Fogel [2,3].

CROSSOVER & CLEANUP

Firstly, the genetic operators operate in the classic genetic method we can out line these as follows:

1. Generate the initial population P(0) at random and set i = 0;

2. Repeat until convergence or time up.
   - Evaluate the fitness of each individual in P(i);
   - Select parents from P(i) based on their fitness.
   - Apply Crossover, followed by Cleanup and then according to mutation rate, apply mutation.

Examining the crossover mechanism in detail, we can see in Figure 1 parent strings (i) and (ii) with their randomly selected pivot points around which the crossover will be applied. This produces the offspring depicted in (iii). This new offspring of the two parent strings clearly is of the correct length as that of its parent strings but does suffer from the problem of replication of cities within the tour. This occurs in the majority of tours generated from crossover as implemented by our system and it is here where the use of Cleanup is needed. Firstly cleanup pin points the errors within the tour, that is those tours which are in contravention of the TSP rules: tours must be of equal length and cities can only be visited once.

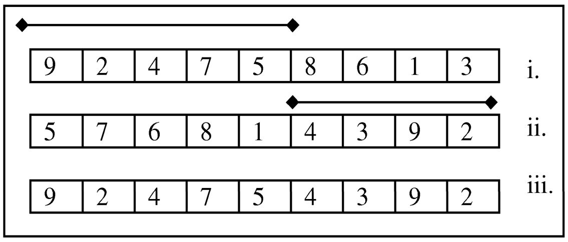

Figure 1. Standard Crossover mechanism

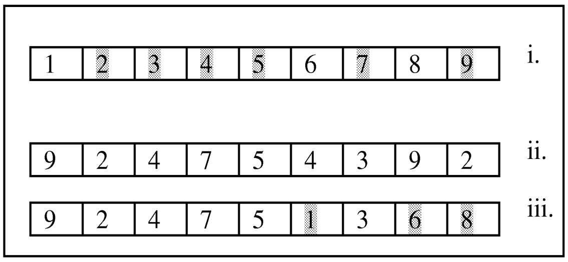

Figure 2. Cleanup operator operation

A check of the tour length is firstly completed and if the tour is found to be of insufficient length the cleanup operator flags this and then continues to the next level of checks. The replication of cities within the tour is tested for by reference to template tour, this template tour is constructed at the very initialisation of the program and importantly is an ordered tour of the numerically referenced cities.

This template, the new offspring and the genetically repaired offspring are depicted in Figure 2. (i), (ii) and (iii) respectively. By flagging those cities that have been encountered in the offspring tour in a left to right manner, cities that have been omitted in the new offspring can be identified. Following this we enter the third phase of cleanup, the reintroduction of those cities excluded in the tour. The fourth and final phase of cleanup is a re-test of the entire tour to finally guarantee the correctness of the final offspring population.

EXPERIMENT 1

Experiments were carried out using two models; one with cleanup, and one without. The data used was a set of TS problems, principally 30, 50, 75, 100 and 150 city distance charts. Throughout these experiments the mutation rate was fixed at 3%, roulette-wheel selection was used, crossover occurred on ever generation, and the initial population of the system was set at 100.

From figure 3, it is clear how the operation of the system varies depending on the inclusion of the clean up operator. This chart compares the number of epochs required to generate tours of (near) identical length. Overall, Cleanup out-performs the standard GA by up to a factor of four, and the smallest improvement reduced the required number of generation evaluations by just under quarter.

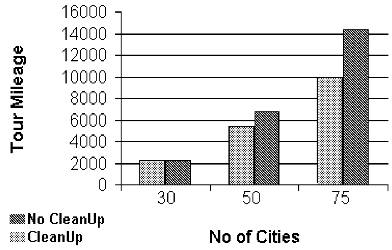

Figure 3: Effect of "Cleanup" operator

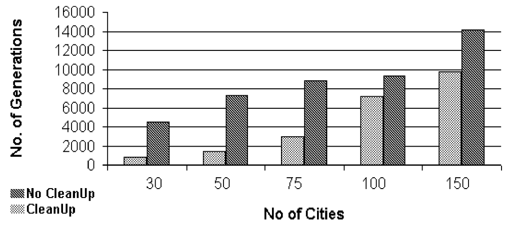

Figure 4: Shortest tour comparison for 30-75 city

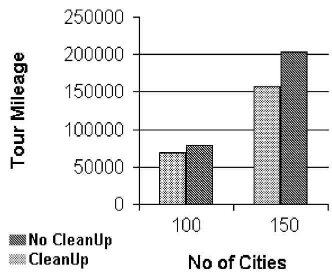

Figure 5: Shortest tour comparison for 100-150 city

More convincingly, the shortest tour produced by Cleanup in every instance, out-performed that of the standard evolutionary algorithm mechanism (see Figures 4 and 5). Thus, not only does Cleanup converge faster on all given comparisons, but produces better results in every instance.

EXPERIMENT 2

In a separate comparison with Fogel [3] the initial population was set at 20. This is the only alteration with the previous experiments. A smaller population promoted a faster convergence on the shortest tour, while not unduly affecting the distance of the shortest tour. This is in agreement with findings of Mitchell [10] and Grefenstette [13], both of whom have also found that the use of small population sizes can be advantageous for selected optimisation problems.

We now compare the best results of our cleanup system with the best results found by Fogel. We see that the use of a combined cleanup and crossover operator reduces the number of offspring required for convergence. These figures are for the 30, 50 and 75 city TSP: for more direct comparison with Fogel [2, 3]. Figure 6 displays the number of offspring generated, before convergence was achieved for both models. Again the cleanup operator drives the search towards convergence more efficiently. Comparison is made using the *Number of Offspring* metric favoured by Fogel rather than the epochs and generations metrics.

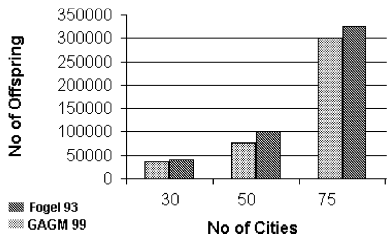

Figure 6: Fogel. Vs. Cleanup (GAGM)

All these experiments indicate that the cleanup operator has a very beneficial influence upon the operation of evolutionary algorithms for the Travelling Salesman Problem.

Perhaps the reasons for cleanup's success may lie in the fact that "unfit" offspring undergo genetic repair before being returned to the gene pool. Thus, our evolutionary system combines evolutionary mechanisms with "genetic engineering" to focus on useful solutions.

CONCLUSION

Evolutionary algorithms provide an efficient means of discovering near-optimal solutions to the travelling salesman problem, although some validation of tours is necessary. In this paper we have introduced cleanup, a new genetic operator for the TSP, and evaluated its performance on a variety of tour sizes. The results indicate that the use of the cleanup operator has a marked influence on the number of generations necessary to converge on a "shortest" tour. Furthermore, cleanup also has a marked improvement on the final distance of the shortest tour, in companion with standard GA solutions. The cleanup only validates and repairs tours and thus the underlying genetic structure remains largely intact. We deliberately chose to use mileage charts rather than artificial graph tours, to simulate real world situations such as 1-way streets, diversions, or invalid connections.

When compared to previous work, our evolutionary solution gives better tours faster, resulting in up to an 8% reduction on the required number of offspring to be evaluated.

REFERENCES

[1] W.M. Spears, *Adapting Crossover in Evolutionary Algorithms,* Navy Centre for Applied Research in Artificial Intelligance, NCARAI Report AIC-95-043, Febuarary 1995, 367-384.

[2] D.B. Fogel, *Empirical Estimation of the Computation Required to Discover Approximate Solutions to the Travelling Salesman Problem Using Evolutionary Programming,* Procedings of 2nd Annual Conference on Evolutionary Programming, 1993, 56-61.

[3] D.B. Fogel, *Applying Evolutionary Programming to Selected Travelling Salesman Problems*, Cybernetics and Systems: An International Journal, 1993, 24:27-36.

[4] J.H. Holland, *Adaptation in natural and artificial systems*, (University of Michigan Press, 1975).

[5] D. Goldberg, *Genetic Algorithms in Search, Optimisation and machine learning*, (Reading MA, USA: Addison Wesley, 1989).

[6] J. Leeuwen, *Handbook of Theoretical Computer Science*, (Cambridge USA, London UK: MIT Press, 1990).

[7] D.B Fogel. *An Evolutionary Approach to the Travelling Salesman Problems*, Biological Cybernetics, 1988, 139-144.

[8] Lawler. et al., *The Travelling Salesman Problem*, (New York: John Wiley and Sons Inc, 1986).

[9] J.R. Koza, *Genetic Programming: on the programming of computers by means of natural selection*, (Cambridge USA, London UK: MIT Press, 1994).

[10] M. Mitchell, *An Introduction to Genetic Algorithms,* (Cambridge USA, London UK: MIT Press, 1999).

[11] D.W. Paterson, *Artificial Neural Networks, Theory and Applications*, (London UK: Prentice Hall, 1996).

[12] D. Goldberg & R. Lingle, *Alleles, Loci, and the Travelling Salesman Problem*, Proceedings of an International Conference on Genetic Algorithms and their Applications, 1985, 154-159.

[13] J.J. Grefenstette, *Optimization of control parameters for genetic algorithms*, *IEEE* Transactions on Systems, Man, and Cybernetics, 1986, 16, 1:122-128.