Explorations on Template-Directed Genetic Repair using Ancient Ancestors and other Templates

Donagh Hatton
Dept. of Computer Science
National University of Ireland, Maynooth
Maynooth, Co. Kildare, Ireland
+353-1-7086099

donagh.hatton@nuim.ie

Diarmuid P. O'Donoghue
Dept. of Computer Science
National University of Ireland, Maynooth
Maynooth, Co. Kildare, Ireland
+353-1-7083851

diarmuid.odonoghue@nuim.ie

ABSTRACT

Handling constraints for combinatorial optimization problems is a classic challenge faced by genetic and evolutionary algorithms. This paper explores a naturally inspired genetic repair process to enforce constraints on evolutionary search. Lolle *et al.* (2005) controversially claim that the model plant *Arabidopsis thaliana* appears to repair genetic errors using information inherited from ancestors other than the immediate parents [10] (i.e. *non*-Mendelian inheritance). We adapt this natural template-driven genetic repair process (GeneRepair) to help solve constraint problems. Building upon previous results [6][7][8] this paper explores repair templates that originate across a range of ancestors, between one and many thousands of generations old. The fitness of resulting populations are presented and compared to a benchmark technique using a random repair template [9]. The results show that very ancient (ancestral) repair templates perform best for larger problems, significantly outperforming the benchmark. The impact of background mutation rates on solution quality is also explored. Results suggest that ancestral repair is a good general-purpose constraint handling technique – helping to explain why this strategy might have evolved in nature.

Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search – Heuristic Methods; G.1.6 [Numerical Analysis]: Optimization – Constrained Optimization.

General Terms

Algorithms.

Keywords

Genetic Algorithms, Evolutionary Optimization, Constraint Search, Genetic Repair.

1. INTRODUCTION

Mendelian inheritance is fundamental to our understanding of both biological and computational evolutionary systems. However, Lolle *et al.* [10] found that the model plant *Arabidopsis*

thaliana makes use of ancestral genetic information in order to repair genetic errors in individual plants. This repair mechanism appears to be activated only when a genetic defect is detected, creating an individual that contains some genetic information that was not present in either parent - but which was present in a previous generation. This appears to be a natural constraint handling mechanism, ensuring that certain harmful mutations are not expressed and are not passed on to subsequent generations.

Following from Lolle's discovery, the plausibility and effectiveness of this repair mechanism for constraint handling on the Travelling Salesman Problem (TSP) was investigated [6][7][8]. Using the GeneRepair technique, the viability of using ancestral repair templates was established. However, only three alternative versions of the GeneRepair process were compared, using repair templates originating in the *parent, grandparent* and *great grandparent* generations. These results showed that using non-Mendelian repair templates (*grandparent* or *great grandparent*) generally produced better results than the Mendelian (*parent*) alternative.

This paper builds on these results by stochastically updating the repair template, allowing the exploration of a diverse range of ancestral templates. A variety of repair templates are investigated and these results are compared to the use of a randomly generated template [9], which is used for benchmarking purposes. It may seem that reversion to older and less fit templates would impede evolution and delay convergence towards the global optimum. However Lolle *et al.* [10] have posited that *A. thaliana* does perform genetic repair using ancestral information.

2. CONSTRAINT HANDLING AND GENEREPAIR

4 different methods have been applied to the problem of handling constraints within evolutionary algorithms [4][16]. The first is a penalty point system, where invalid individuals are assigned an arbitrary fitness penalty to decrease or eliminate the probability that they are selected for crossover. The second is to make use of specially modified representations, crossover and mutation operators so that invalid individuals are never generated. This method is heavily problem specific as it requires crossover and mutation operators to be explicitly tailored to suit each problem. The third is to use multi-objective optimization, so that solution fitness and solution validity are considered separately. The final method is genetic repair, where errors in individuals are corrected by a repair algorithm.

The standard problem associated with the genetic repair approach is that it generally relies on heuristics to carry out repair [1][2][3][13][14][18]. This leads to two main problems; first,

repair is seen to be highly problem dependent because of these heuristics. Second, these repair heuristics are seen to reduce the population's diversity which results in poorer quality solutions. The genetic repair approach that we explore in this paper does not suffer from either of these problems because the repaired information originates from the individuals ancestors rather than being the result of a heuristic.

This paper focuses on biologically inspired approaches to constraint handling. We adapt a recently discovered genetic repair process so that it can solve a variety of combinatorial optimization problems. All the results that are presented are for instances of the Traveling Salesman's Problem (TSP).

GeneRepair is a constraint handling technique that makes use of a template in order to repair errors in individual solutions in an evolutionary algorithm. These templates are often supplied from either parent solutions of an individual, another individual in the current population or through the random generation of a valid template. It has been shown that the use of a random template usually produces better results than templates derived from another individual or from the parent [12]. Similar approaches that make use of template based error correction have shown that a random template can serve as a good general purpose repair choice as it is problem agnostic and straightforward to implement [9][17].

The GeneRepair operator repairs errors that have been introduced by the crossover or mutation operators. Invalid alleles are replaced with corrective alleles that are sourced from some repair template. These corrective alleles repair the identified genetic defect, thereby removing the constraint violation. For combinatorial optimization problems like the TSP, defects are detected by the presence of missing or duplicate alleles. Mitchell *et al.* [12] compared the use of randomly generated templates as well as the use of templates derived from parent individuals in the repair process, finding that randomly generated templates tended to provide the best results. We also highlight that with GeneRepair; invalid individuals are always repaired, contrasting with Orvosh and Davis who suggest that repairs should be carried out only 5% of the time [14]. It has been shown [4] that the GeneRepair approach produces significantly better results than the penalty point method.

2.1 The GeneRepair Algorithm

The genetic repair process is typically invoked when the process of generating a phenotype from the genotype highlights the presence of some genetic defect. GeneRepair is an adjunct process that can convert invalid genomes into valid ones, but repair does not generate valid solutions *ab initio*. The basic algorithm is composed of 6 discrete steps as listed below:

1. *Creation*
2. *Evaluation*
3. *Selection*
4. *Crossover*
5. *Mutation*
6. ***GeneRepair***

In the GeneRepair step, genetic errors are detected and then corrected. Genetic errors are defined as any alleles that violate the constraints imposed by the problem. For the TSP, these errors take the form of duplicate cities on any tour. Figure 1 provides a pseudo-code description of the GeneRepair algorithm.


```
template ← Existing valid solution
solution ← Solution being repaired
errorList ← Ordered list of defective genes in solution
missingList ← List of genes missing from solution
for each item in template do:
        if template[item] in missingList then
                repairList.add(template[item])
repairCount ← 0
for each item in solution do:
        if solution[item] in errorList do:
                errorList.remove(item)
                solution[item] ← repairList[repairCount]
                repairCount ← repairCount + 1
        if repairCount = count( repairList)
                end
```

Figure 1. Pseudo-code describing the GeneRepair algorithm as applied to the TSP

After the repair step is completed, all defective alleles in the solution have been repaired and replaced with valid genes derived from the appropriate repair template. The choice of which template to use is a key issue that has a measurable impact on solution quality. This issue is discussed further in the following sections. Algorithmic parameters for GeneRepair include: changing the direction in which the repair proceeds and varying the starting locus of the correction phase. Randomly selecting the ancestor for use as a repair template produces better results than selecting the fittest of the available ancestors [6].

2.2 Deterministic Ancestral Repair

Previous work [6][7][8] on deterministic GeneRepair compared the performance of three repair templates: *parent, grandparent* and *great grandparent*. Figure 2 demonstrates the workings of the GeneRepair operator using the *grandparent* repair template. Here, an error in the F3 generation is repaired using genetic information originating in the F1 generation, as discussed in [10].
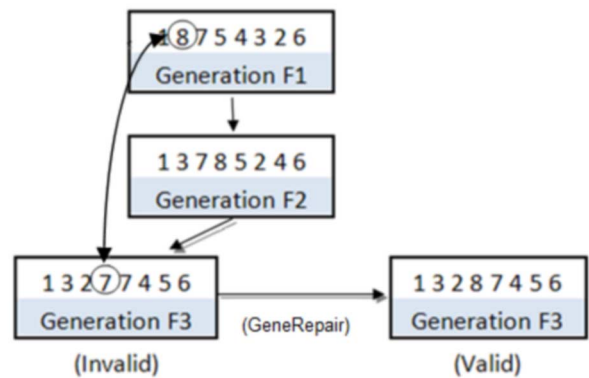

Figure 2. Example of a GeneRepair operation using the *grandparent* template.

The direction in which repair is carried out can influence the fitness of solutions found, with repair operating in randomly varying directions producing the best results [7]. All experiments discussed in this paper use a randomly varying repair direction.

2.3 Stochastic Ancestral Repair

Deterministic GeneRepair using the *grandparent* or *great grandparent* template generally produces better results than the *parent* template [6][7]. By replacing these deterministic rates with stochastic replacement rates however, we can examine a much broader range of ancestor generations. This paper explores rate between 1.0 and 0.0001. This greatly extends the range of templates that can be examined, as a rate of 0.333 approximates the oldest template used by [6][7]. These low replacement rates result in repair templates that are many hundreds (or thousands) of generations removed from the erroneous individual[1].

The stochastic replacement rate works as follows: Rather than use a specific ancestor as the repair template for an invalid individual, a single complete copy of an ancestor template is stored for each individual in the population. At the crossover stage, this ancestor template is updated with either of the parents of the individual with probability $p = replacement\ rate$. If the ancestor template is not updated, the ancestor template of either of the parents is assigned to the new individual. In this manner it is possible for very distant ancestor templates to be passed down, dependant on the replacement rate. Note that with this template update rule, a replacement rate of 1.0 corresponds to the use of the *parent* template in deterministic ancestral repair, as the ancestral template will be updated each generation. A replacement rate of 0.0001 however, can result in the use of templates that are an average of 10,000 generations behind the current generation.

This mechanism of stochastic replacement allows us to explore repair templates from generations much older than the parent, grandparent or great grandparent. Stochastic replacement allows use of these older generations without the memory and related overheads. Additionally, it seems more plausible that *A. thaliana* would store just one repair template to meet its needs – rather than storing multiple redundant repair templates as in [7].

3. RESULTS

This section presents the results of varying the replacement rate and the corresponding impact this has on solution quality. These ancestral templates are compared to the random template where a new template is generated for each invalid individual being repaired. Due to the impact of the background mutation rate on these results, we compare performance for a variety of mutation rates, both above and below the optimum rate.

The results discussed in this paper were all produced using datasets from the TSPLIB library [15]: 101 cities (eil101), 1,379 cities (nrw1379) and 18,512 cities (d18512). All of the experiments were conducted using tournament selection with a tournament size of 2, single point swap mutation and single point crossover. The experiments using 101 cities and 1,379 cities used a population of 100 while the 18,512 city problem used a population of 10. Mutation rate differed according to problem size, and the identification of the optimal mutation rate for each instance is discussed in section 3.1. All

---

[1] Personal communication with S.J. Lolle - While Lolle *et al.* (2005) say that the repair template existed in the grandparent genome, they indicate that it may have originated in a very distant generation.

---

mutation rates discussed in the following section are expressed as a percentage of the genes in each solution. Unless otherwise stated, mean values are derived from 10 separate experiments for each parameter combination. Note also that the TSP is a minimization problem, thus the lower fitness score is always better.

3.1 Optimal Mutation Rate

First, we determined the optimal mutation rate for each instance of the TSP under consideration. The optimal mutation rate was necessary to allow later comparisons within the optimal rate and outside of it. As a baseline result we compared the parent repair template and the random template approach.

A series of experiments was run for 50,000 generations with mutation rates ranging from 0 to 0.1. For each problem, data were collected for both stochastic GeneRepair with a replacement rate of 1.0 (corresponding to deterministic parent repair) and for random template generation. In all cases the optimal mutation rate was consistent for both types of repair. For the 101 problem, the ideal range was between 0.1 and 1. Figure 3 shows the fitness distribution for this range in steps of 0.1. A mutation rate of 0.6 produced the best results.
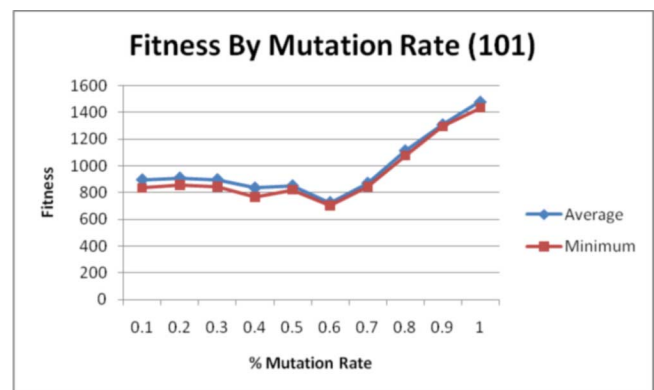

Figure 3. Fitness by mutation rate for 101 city TSP.

For the 1,379 problem, the ideal range was between 0.001 and 0.06. Figure 4 shows the fitness distribution for this problem (note the use of a non-linear scale). A mutation rate of 0.3 produced the best results.
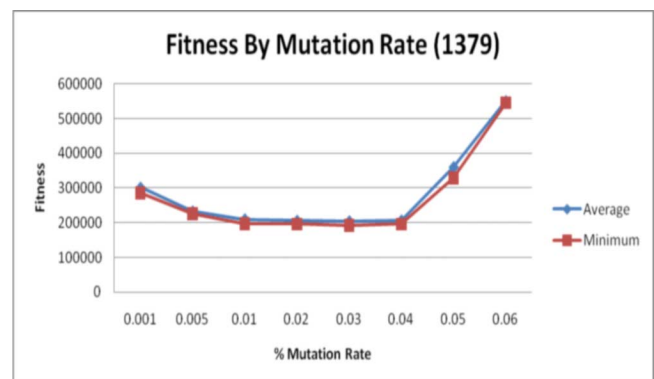

Figure 4. Fitness by mutation rate for 1,379 city TSP.

For the 18,512 problem, the ideal range was between 0.0001 and 0.06. Figure 5 shows the fitness distribution for this problem. A mutation rate of 0.001 produced the best results.
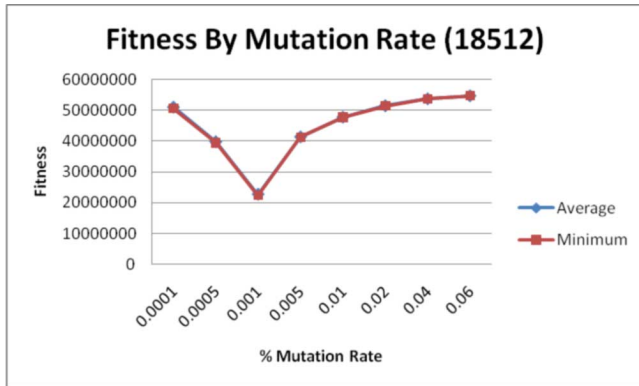

Figure 5. Fitness by mutation rate for 18,512 city TSP.

3.2 Ancestral Distance of the Repair Template

In this section we explore the impact of using more ancient repair templates on fitness when used at the optimal mutation rate. Lolle *et al.* [10] mention that Arabidopsis appears to use the grandparent genome for repair. We examined repair templates originating up to many thousands of generations old.

A range of stochastic rates was chosen to allow for a variety of ancestral distances. The replacement rates examined were: 0.0001, 0.001, 0.01, 0.1, 0.25, 0.5 and 1.0. The performance of each of these rates was examined along with the performance of a randomly generated template. In the following diagrams, "Random" on the replacement rate axis refers to the use of a randomly generated template.

Figure 6 shows the results obtained for the 101 city problem over 500,000 generations at the optimal mutation rate of 0.6. The random template achieved the best global fitness score and the best mean fitness score. A Mann-Whitney U test [5] shows that the random template reaches a lower fitness score than the stochastic rates with a confidence level of $p < 0.026$.
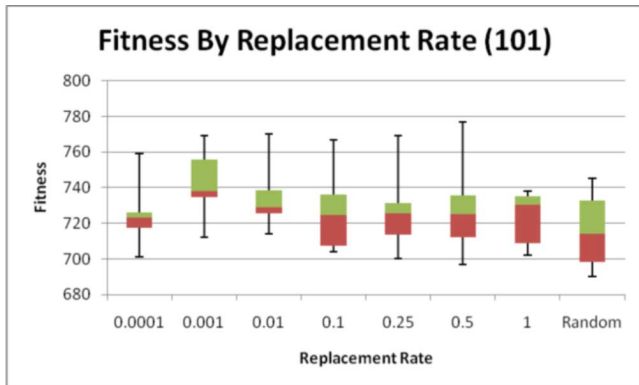

Figure 6. Fitness by replacement rate over 500,000 generations for 101 city TSP at a mutation rate of 0.6%.

Figure 7 shows the results obtained for the 1,379 city problem over 500,000 generations at the optimal mutation rate of 0.03. Here it can be seen that the randomly generated template significantly outperforms every tested stochastic replacement rate with a confidence level of $p < 0.0001$.
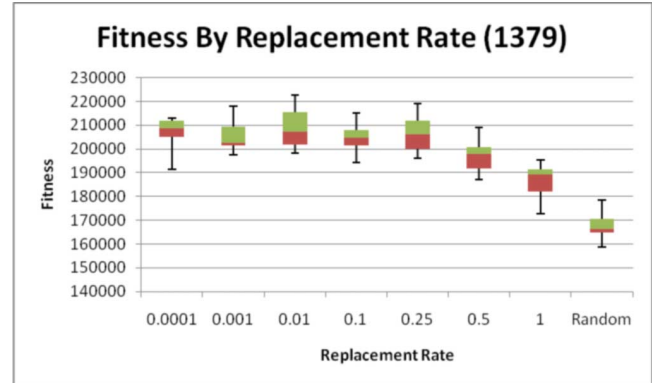

Figure 7. Fitness by replacement rate over 500,000 generations for 1,379 city TSP at a mutation rate of 0.03%.

Figure 8 gives the results obtained for the 18,512 city problem over 2,000,000 generations at the optimal mutation rate of 0.001. Here the random template has actually produced the worst results, being outperformed by the best stochastic replacement rate of 0.1 with a confidence level of $p < 0.0001$.
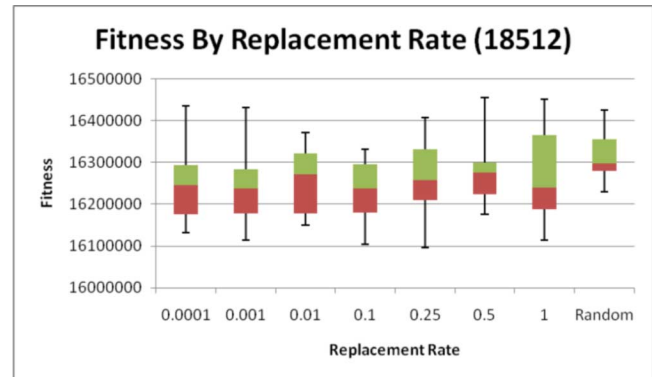

Figure 8. Fitness by replacement rate over 2,000,000 generations for 18,512 city TSP at a mutation rate of 0.001%.

The failure of the random template to be competitive in the instance of the 18,512 city TSP is possibly due to the increased problem size. With a randomly generated template, invalid genes will effectively be replaced with an arbitrary one from the set of permissible cities. Larger problems will have a larger set of these permissible replacements. It is possible that ancestral templates on the other hand, are less likely to break up the efficient ordering of genes that already exists in previous generations, as genes are used in the repair process in the order in which they occur in the template.

Having examined the results at the optimal rate, the following two sections examine the results below and above this rate respectively.

3.3 Replacement Rates below the Optimal Mutation Rate

Next, we examined the results generated when the rate of background mutation was greater than the optimal rate, for the same experimental conditions outlined above.

Figure 9 gives the results obtained for the 101 city problem over 500,000 generations at the lower than optimal mutation rate of 0.4. In this instance, the random template again outperforms several of the stochastic rates with a confidence level of $p < 0.02$ but the difference between the random template and the stochastic rate of 0.1 is not enough to be significant ($p = 0.071$). Again the random template achieved the best global fitness score.
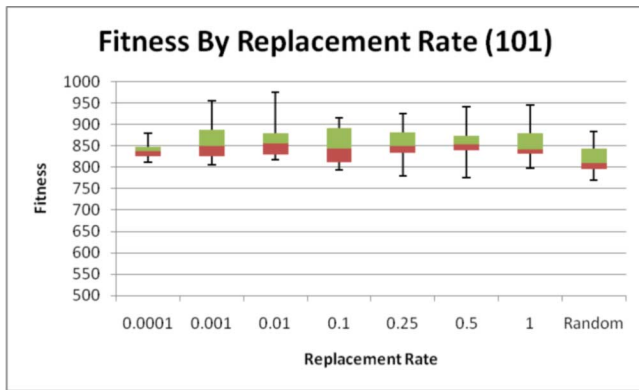

Figure 9. Fitness by replacement rate over 500,000 generations for 101 city TSP at a mutation rate of 0.4%.

Figure 10 shows the results obtained for the 1,379 city problem over 500,000 generations at the lower than optimal mutation rate of 0.005. In this case there is no statistical significance between the random template and the best stochastic replacement rate of 0.25. The rate of 0.25 achieved the best global fitness score.
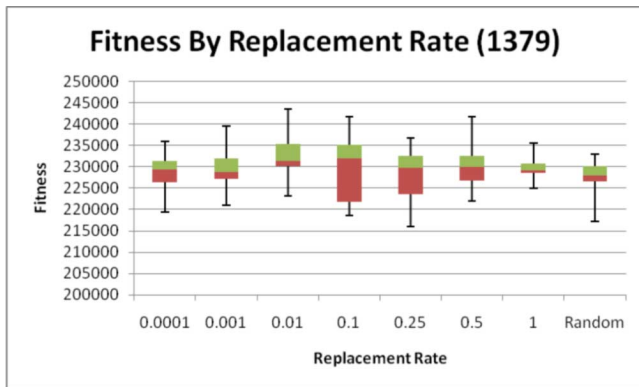

Figure 10. Fitness by replacement rate over 500,000 generations for 1,379 city TSP at a mutation rate of 0.005%.

Figure 11 gives the results obtained for the 18,512 city problem over 2,000,000 generations at the lower than optimal mutation rate of 0.0005. There is no statistical significance between the

random template and the best stochastic replacement rate of 0.0001 ($p = 0.0708$). The replacement rate of 1.0 achieved the best global fitness score due to an extreme outlier while the rate of 0.0001 achieved the best mean fitness score.
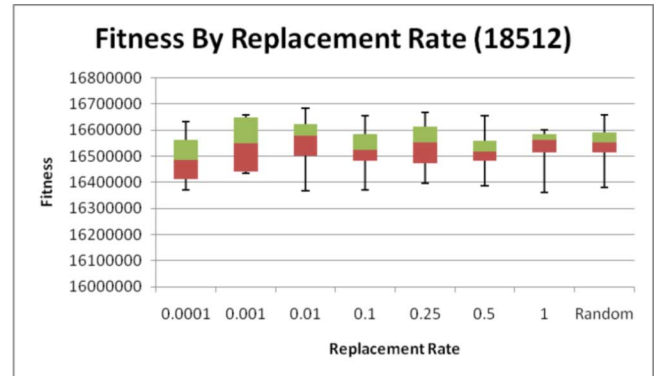

Figure 11. Fitness by replacement rate over 2,000,000 generations for 18,512 city TSP at a mutation rate of 0.0005%.

These results indicate that there is likely no particular advantage to the use of a random template at mutation rates lower than the optimal, with the effectiveness of the random template being reduced relative to stochastic replacement as the problem becomes larger.

3.4 Replacement Rates above the Optimal Mutation Rate

The results below the optimal mutation rate were broadly in line with those at the optimal mutation rate. However, the results above the optimal mutation rate were strikingly different.

Figure 12 shows the results obtained for the 101 city problem over 500,000 generations at the higher than optimal mutation rate of 0.8. The difference in results here is very pronounced. Every single stochastic replacement rate used outperformed the random template with a confidence level of $p < 0.0001$. The best global fitness score was achieved with a replacement rate of 0.01.
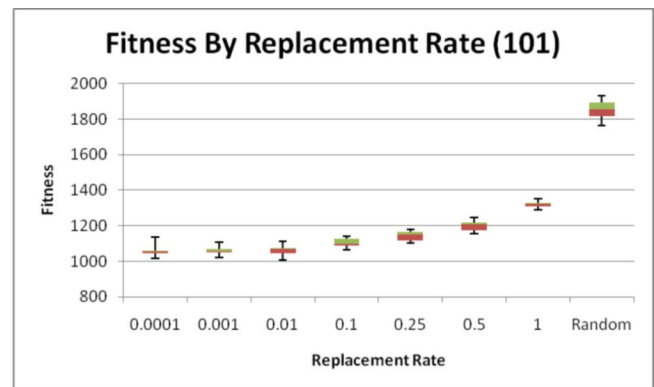

Figure 12. Fitness by replacement rate over 500,000 generations for 101 city TSP at a mutation rate of 0.8%.

Figure 13 gives the results obtained for the 1,379 city problem over 500,000 generations at the higher than optimal mutation rate of 0.5. Random template generation underperformed significantly compared to each of the stochastic replacement rates ($p < 0.0001$) shown. The best global fitness score was achieved with a replacement rate of 0.01.
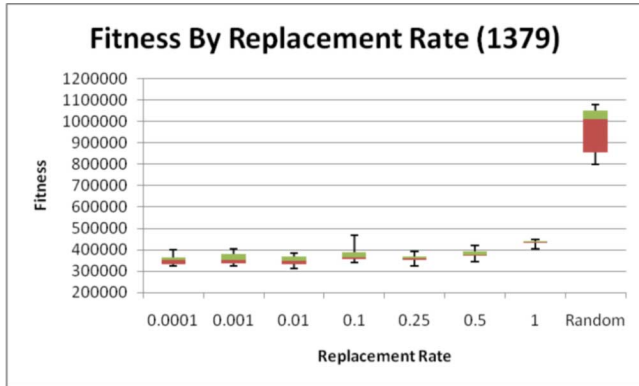

Figure 13. Fitness by replacement rate over 500,000 generations for 1,379 city TSP at a mutation rate of 0.05%.

Figure 14 gives the results obtained for the 18,512 city problem over 2,000,000 generations at the higher than optimal mutation rate of 0.01. These results are in line with the others that used a higher than optimal mutation rate. The random template repair was outperformed by every stochastic replacement rate with a confidence level of $p < 0.0022$. The best global fitness score was achieved with a replacement rate of 0.25 while the best mean fitness was achieved with a replacement rate of 0.01.
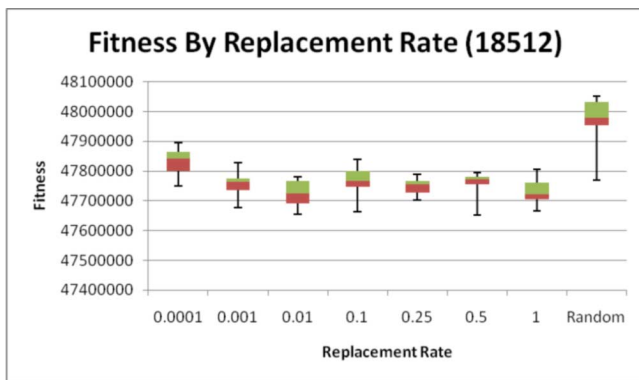

Figure 14. Fitness by replacement rate over 2,000,000 generations for 18,512 city TSP at a mutation rate of 0.01%.

3.5 Comparison of Ancestral and Random Repair Templates

The results obtained at rates of mutation outside the optimal show a clear pattern. With lower mutation rates, there does not appear to be any significant advantage to using a random template. With higher mutation rates, every stochastic replacement rate achieved a better mean fitness score than the random template in every

experiment performed. This indicates that the effectiveness of the random template is severely reduced outside of the ideal mutation rate. In addition to the mutation rate, it appears that problem size plays a significant role in the effectiveness of the random template. All of the results for the 18,512 city problem show that a random template consistently underperforms compared to stochastic template replacement. This suggests that as a problem becomes larger, the random template becomes a less effective basis for repair.

Outside of a small "Goldilocks Zone" where the algorithmic parameters are correctly set for optimal results, the random template repair either fails to perform, or performs no better than ancestral repair. We hypothesize that stochastic replacement rates produce fitter solutions than random templates for both larger problems and for parameter combinations outside of this Goldilocks Zone because they exert an implicit clamping factor on the amount of mutations that can occur. With low mutation rates, genetic repair is required less often, as shown by Mitchell *et al.* [12]. In practice, this means that the likelihood of ancestral templates preserving useful sequences of genes is diminished, bringing the utility of stochastic ancestral repair in line with that of random template repair. With high mutation rates or with larger problems, the size of the search space examined with a random template is increased. In these instances, stochastic ancestral templates are more likely to preserve useful sequences of genes and in effect, reduce the negative impact that these higher mutation rates can cause in the population.

These results show that with the TSP, a random template produces fitter solutions when used with small problems at the ideal mutation rate. For larger problems and for problems where the optimal mutation rate is unknown, random template directed genetic repair is less effective when compared to ancient ancestor driven template repair. It is likely that this approach would be applicable to wide range of combinatorial optimization problems. The real strength of the GeneRepair operator is that it is largely problem agnostic. Ancestor based GeneRepair can be applied to any combinatorial problem for which an arbitrary valid solution can be generated. We suggest that a stochastic replacement rate of between 0.01 and 0.001 is a better general purpose repair choice for this set of problems than a random template driven approach, as these replacement rates consistently produce good results.

4. CONCLUSION

In 2005, Lolle *et al.* published work [10] indicating the existence of non-Mendelian inheritance in *Arabidopsis thaliana*. This repair mechanism makes use of ancestral genetic information to repair genetic errors in individual plants, where each individual records its own ancestral template. We build on previous work [6] by examining two main modifications to the naturally inspired GeneRepair operator. First we explored the use of repair templates that originated up to many thousands of generations old, using a stochastic template replacement approach. Second, we compared the effectiveness of various stochastic replacement rates to using randomly generated repair templates. All results relate to 3 instances of the Traveling Salesman's Problem (TSP).

While results varied between experimental conditions, some general observations can be made. First, the best results were produced by repair templates that are approximately 100-1000 generations old, especially for the larger problems. This was a

relatively surprising finding as Lolle's original paper [10] indicated the repair template of *A. thaliana* was just two generations old. Furthermore we note that very recent repair templates rarely produced the best results - and only did so at the optimal mutation rate for the smaller problems.

We identify a pattern across the results produced by the random template. At the optimal mutation rate and for the small problems the random template produced the best results. However, when the rate of background mutation is above the optimal rate, the random template produces the worst results. For rates below the optimal, the random template produced results no better than those produced with any ancestral repair strategy.

Overall our results indicate that ancestral repair templates seem to provide a better general-purpose repair mechanism than random templates. For large TSP problems or for problems for which the optimal parameters (population size, mutation rate *etc.*) are unknown, using ancient repair templates produces superior results. The approach outlined in this paper can trivially be applied to any combinatorial optimization problem for which an arbitrary valid solution can be generated, regardless of the quality of that solution. Finally, these results indicate that the use of an ancestor driven repair mechanism could be a feasible method for genetic repair in the biological domain, agreeing with Lolle *et al*. [10] and lending support to the plausibility of the existence of such a mechanism in the natural world. Further developments in genetics may yield even more inspiration for genetic algorithms.

6. REFERENCES
[1] Ahn, C.W. and Ramakrishna, R.S., A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Transactions on Evolutionary Computation*, 6(6), (2002), 566-579.

[2] Arroyo, J. M. and Conejo A. J., A parallel repair genetic algorithm to solve the unit commitment problem. *IEEE Transactions on Power Systems*. 17(4), (2002), 1216-1224.

[3] Bäck, T., Schütz, M. and Khuri, S., A comparative study of a penalty function, a repair heuristic, and stochastic operators with the set-covering problem. In *European conference on Artificial Evolution*, (Berlin, Germany, 1996), 320-332.

[4] Coello Coello C., Theoretical and numerical constraint handling techniques in evolutionary algorithms: A survey. *Comp. Methods in App. Mathematics and Engineering,* 191(11, 12) (2002), 1245–1287.

[5] Conover W.J., 1999, *Practical Nonparametric Statistics* (3rd Ed.), John Wiley & Sons.

[6] Fitzgerald A. and O'Donoghue D.P., Genetic repair for optimization under constraints. In *10th Intl Conference on Parallel Problem Solving From Nature*, (Dortmund, Germany, 2008), Springer, 399-408.

[7] Fitzgerald A., O'Donoghue, D.P. and Xinyu, L., Genetic repair strategies inspired by *Arabidopsis thaliana*, *Lecture Notes in Artificial Intelligence (LNAI)*, 6206, (2010), 61-71.

[8] Fitzgerald A. and O'Donoghue, D.P., Biologically inspired non-Mendelian repair for constraint handling in evolutionary algorithms. In *GECCO - Workshop on Evolutionary Computation Techniques for Constraint Handling*, (Portland, Oregon, USA, 2010) 1817-1825.

[9] Lichtblau D., 2010, *Differential evolution in discrete and combinatorial optimization*, Mathematica Tutorial Notes, MathSource, Wolfram Research Inc.,

[10] Lolle S.J., Victor J., Young J. and Pruitt R. Genome-wide non-Mendelian inheritance of extra-genomic information in Arabidopsis. *Nature*, 434(1) (2005), 505–509.

[11] Michalewicz, Z. and Fogel, D.B., 2000, *How to solve it: Modern heuristics*. Berlin, Heidelberg, New York: Springer Verlag.

[12] Mitchell G.G., O'Donoghue, D.P., Barnes D. and McCarville M., GeneRepair - A repair operator for genetic algorithms. In *The Genetic and Evolutionary Computation Conference*, (Chicago IL, USA, 2003), ACM, 235-239.

[13] Nakano R. and Yamada, T. Conventional genetic algorithm for job shop problems. In *Proceedings of the 4th International Conference on Genetic Algorithms*. (San Diego, CA., USA, 1991), 474-479.

[14] Orvosh D. and Davis L., Shall we repair? Genetic algorithms, combinatorial optimization and feasibility constraints. In *Proceedings of the 5th International Conference on Genetic Algorithms*, (San Francisco CA., USA, 1993), Morgan Kaufmann Publishers Inc., 650.

[15] Reinelt G., TSPLIB - a travelling salesman problem library, *ORSA Journal of Computing* 3(4) (1991), 376 – 384.

[16] Salcedo-Sanz S., A survey of repair methods used as constraint handling techniques in evolutionary algorithms, *Computer Science Review*, 3(3) (2009), 175-192.

[17] Tate D. and Smith A., A genetic approach to the quadratic assignment problem, *Computers and Operations Research*, 22(1) (1995), 73−83.

[18] Walters, T., Repair and brood selection in the traveling salesman problem. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature - PPSN* (Amsterdam, The Netherlands, 1998) 813-822.