Arabidopsis thaliana Inspired Genetic Restoration Strategies

Donagh Hatton *donagh.hatton@nuim.ie*
Dept. of Computer Science
National University of Ireland, Maynooth
Maynooth, Ireland

Diarmuid P. O'Donoghue *diarmuid.odonoghue@nuim.ie*
Dept. of Computer Science
National University of Ireland, Maynooth
Maynooth, Ireland

Abstract

A controversial genetic restoration mechanism has been proposed for the model organism *Arabidopsis thaliana*. This theory proposes that genetic material from non-parental ancestors is used to restore genetic information that was inadvertently corrupted during reproduction. We evaluate the effectiveness of this strategy by adapting it to an evolutionary algorithm solving two distinct benchmark optimization problems. We compare the performance of the proposed strategy with a number of alternate strategies – including the Mendelian alternative. Included in this comparison are a number of biologically implausible templates that help elucidate likely reasons for the relative performance of the different templates. Results show that the proposed non-Mendelian restoration strategy is highly effective across the range of conditions investigated – significantly outperforming the Mendelian alternative in almost every situation.

Keywords: Evolutionary Algorithms, Genetic Restoration, Arabidopsis thaliana, Constrained Optimization

1. INTRODUCTION

The inspiration for this paper lies in a recently proposed genetic repair process in the model plant *Arabidopsis thaliana* [1]. *A. thaliana* is widely used in genetic studies, having a very fast life-cycle of around 6 weeks from germination to mature seed, making it ideal for longitudinal study and allowing comparison of multiple gnomes from the same species. It has a relatively short genome with about 157 million base-pairs encoding 27,000 genes and was the first plant to have its entire genome sequenced in the year 2000.

The reported study attests that some offspring of this plant inherit genetic material from individuals *other than* the direct parents. This controversial non-Mendelian hypothesis originated from studies of the HOTHEAD (*HTH*) gene, which impacts on formation of the epidermis and flower of this plant. The mutant *hth* form of the HOTHEAD gene is recessive and plants were studied that were homozygous for this recessive mutant allele (*hth/hth*). When these *hth/hth* mutants were allowed to self-fertilize, amazingly over 10% of the resulting progeny were of the normal wild type (*HTH/hth* or even *HTH/HTH*) - even though the normal *HTH* gene was not detected in either parent. This 10% rate of reversion is far higher than can be accounted for by random point mutations, which would generally occur with a frequency of the order of 1 per billion per allele per generation [2].

These findings are not consistent with the standard Mendelian model of inheritance and led to the controversial non-Mendelian theory. While these findings have been attributed to "pollen contamination" [3] [4], this claim itself has been contested [5].

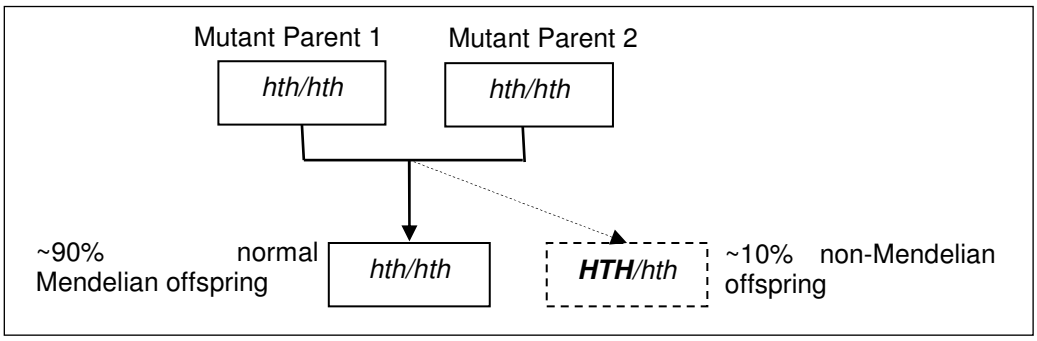

FIGURE 1: Approximately 10% of the offspring of hth/hth mutants revert back to the normal wildtype (either HTH/hth or even HTH/HTH). This finding is not consistent with normal Mendelian inheritance.

2. EVOLUTIONARY ALGORITHMS

Evolutionary algorithms, genetic algorithms [6] and related areas are based on Darwin's theory of natural selection [7]. They are widely used to find near optimal solutions to complex problems, wherein a population of (initially random) solutions are allowed to evolve under the guidance of a fitness function. Fitter genomes survive and reproduce more frequently, so that surviving individuals tend towards the optimal solution for a given problem.

While Evolutionary Algorithms (EA) are very effective and adaptable across a range of different problem types, they have a fundamental difficulty in dealing with constrained evolution. Constrained evolution addresses problems where the evolutionary process can generate genotypes that do not correspond to valid phenotypes. These infeasible solutions are inadvertently generated by either the process of reproduction or as a result of random mutations. These invalid (or infeasible) solutions are somewhat analogous to abortion events in natural evolution, where a viable phenotype cannot be produced from the genotype.

Some notable facets of our evolutionary strategy should be made clear at this point. First, our model is more obviously akin to the inheritance of sequences of genes, rather than the inheritance of lower-level base-pairs *per se*. Second, for our problems we can evaluate fitness directly from the genome sequence due to the simple and direct relationship between the genotype and phenotype for the investigated problems. Third, EA applied to permutation problems converge very slowly due to the irregular nature of the problem space. Finally, unlike the genetic restoration process proposed for *Arabidopsis thaliana*, we immediately repair genetic defects rather than repairing after a mutant solution creates a new solution.

2.1 Constrained Optimization Problems

Particularly challenging for EA are the permutation and combinatorial optimization problems, such as the Travelling Salesman's Problem (TSP) and Quadratic Assignment Problem (QAP). Solutions to these problems can be stated as identifying the optimal ordering of an initial list of items. The issue with employing an EA to solve such problems concerns ensuring that the core genetic operators of crossover and mutation do not introduce omission errors into the representation of solutions. Furthermore, due to the tightly constrained nature of these problems, the number of invalid solutions vastly out-numbers the valid solutions. Figure 2 demonstrates that as the size of the problem increases (across the horizontal axis), the invalid solutions vastly outnumber the valid solutions – note the exponential scale on the vertical axis. Thus, repairing these infeasible solutions is crucial to the effective performance of an evolutionary algorithm.

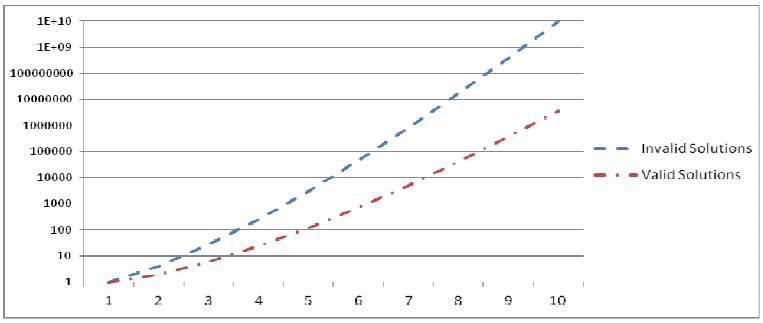

FIGURE 2: The Invalid Solutions Vastly Outnumber The Valid Solutions.

Before examining the restoration process, we briefly outline how genetic defects may be introduced into representations that have not been specifically designed for each problem type. For general representations of combinatorial optimization problems, invalid phenotypes can be easily identified by examining the genotype itself. Each duplicate gene denotes a genetic defect (see figure 3). In addition, these errors are accompanied by omission errors, as they occur simultaneously for this general representation scheme.

FIGURE 3: The processes of recombination/crossover (left) and mutation (right) easily generate invalid solutions to combinatorial optimization problems.

2.2 Template Driven Restoration in Evolutionary Optimization

Four distinct approaches have been adopted to enforce constraints on evolutionary searches [8] and this paper explores a novel strategy within the "genetic repair" approach. One approach to genetic repair has centred on the use of template solutions, which serve to restore genetic information that was accidentally removed during the reproduction or mutation phases of the algorithm. Template driven repair was introduced by Mitchell *et al.* [9], who compared the performance of static repair templates against a number of Mendelian-like templates. Later work showed that random repair templates [10] [11] generally produced better results, leading to their use in some standard application programs.

We now describe the template driven repair process as used for the combinatorial optimization problems. An erroneous genome is located by scanning its genes and locating any duplicate genes (we will return to the implicit dependency between the direction in which this process is conducted and the specific gene that is deemed to be the duplicate). Next, by comparison with the repair template we identify genes contained in the repair template, but which are not present in the invalid solution. These are highlighted in boldface on the lower line of figure 4. The restored solution in this case would be a b c d e f. However, if a different restoration template were used (say, a b e d c f) this would generate a different solution (a b e d c f). Thus, different repair templates have an impact on the fitness of the restored solutions. The rest of this paper concerns the impact that using different repair templates has on the quality of the solutions generated.

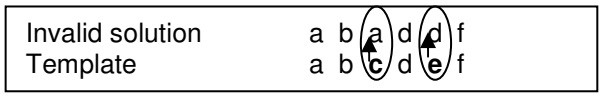

FIGURE 4: The repair template is used to restore information that was missing from the invalid solution.

The repair templates that are the focus of this paper are:

1. Parental templates. Repair templates are sourced from the parents of the invalid solutions. These restoration templates represent an alternative and more parsimonious Mendelian repair theory.

2. Ancestral templates. Templates are retrieved from ancient archived ancestors of the invalid solution. Repaired solutions are non-Mendelian as they contain genomic information from an individual other than the immediate parents. This paper investigates templates ranging from 10 generations to one thousand generations old.

3. Random templates. A new random template is generated for each invalid individual.

This paper explores the impact that a number of parameters have on the relative fitness of these genetic restoration templates. The main parameters investigated are problem type (TSP and QAP), problem size (ranging from 26 to 18152 genes in length), and the rate at which mutations impact on the population. We characterize this final parameter (mutation) into three distinct categories – low, tuned and high. Mutation rates have a very significant impact on solution quality, but manual tuning is generally required to find the optimal rate. In the remainder of this paper, the tuned mutation rate is near optimal. The low mutation rate conditions describe template performance below this rate while the high condition represents performance at mutation rates above the optimal. These three categories serve to adequately describe the (relative) performance of each of the different restoration templates. It should also be noted that optimal mutation rates vary widely between problem type, problem size, as well as other factors.

3. RESULTS

Results for two distinct problem types are presented in this section. The Traveling Salesman Problem (TSP) is the problem of visiting each of n cities once, with the goal of minimizing the total distance travelled. Solutions to the TSP are represented as a permutation of integers, representing the order in which cities are visited. The cost function for a TSP instance is the total route distance of the tour represented by a candidate solution. The TSP is a minimization problem, and so for all of the following results, a lower cost is better. The Quadratic Assignment Problem (QAP) is the problem of assigning *n* facilities to *n* different locations with the goal of minimizing the sum of the distances multiplied by a corresponding flow weights between the facilities. As with the TSP, a solution is represented as a permutation of integer values, representing the location of assignment of each facility in order. The QAP is also a minimization problem, so smaller values on the fitness axis are better. Problem instances were selected from the widely used TSPLIB [12] and QAPLIB [13] problem libraries. Specific benchmark problems are references, such as eil101. This is a 101 gene (city) instance of the TSP problem. Similarly, d18512 is a TSP problem whose solutions contain 18512 genes.

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 eil101, nrw1379 and bur26a used a population of 100 while the 18,512 city problem used a population of 10. Mutation rate was selected to provide a rate that would be considered low, tuned and high according to problem size. All mutation rates discussed in the following section are expressed as a percentage of the genes in each solution, and represent the probability that a given gene will be mutated at each crossover event. Unless otherwise stated, mean values are derived from 10 separate experiments for each parameter combination. We compare ancestral templates from a range of ages. 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 same genetic restoration process was used for both the TSP and QAP problems, indicating some generality for the approach adopted.

The key parameter that controls the age of the repair templates used is the Template Update rate (TU). TU controls the probability at each reproduction event that the current parent will be preserved as the ancestor to use for genetic restoration. For example, a TU of 1 means that the

parent of the new individual will be retained with probability p = 1, effectively resulting in the ancestor template always being constructed from the parent of the solution. A TU of 0.001 however, means that the ancestor template will be updated with that probability, extending the average age of the ancestor template to approximately 1000 generations.

3.1 Low Mutation Rates

The first set of results consists of the experiments conducted at rates of mutation lower than the optimal for problem type. In a typical EA, these conditions would result in insufficient mutation, reducing the speed of convergence of the population and impairing solution quality through a lack of diversity [6]. Figure 5 shows the results obtained for the 101 city TSP over 200,000 generations at the lower than optimal mutation rate of 0.01. A lower Template Update rate, corresponding to older ancestral templates, produced the best results (p-value < 0.01).

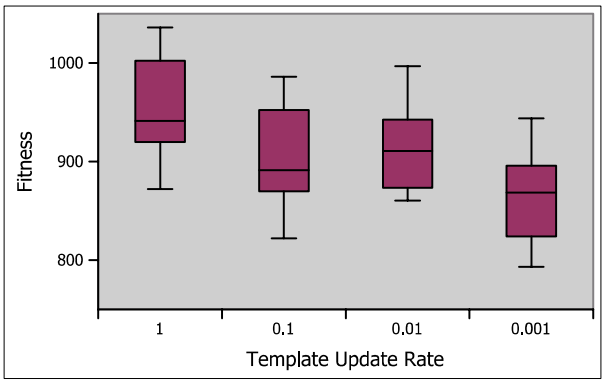

FIGURE 5: Effect of varying replacement rate on solution quality at a low mutation rate for TSP instance eil101.

Figure 6 shows the results obtained for the 1379 city TSP over 200,000 generations at the lower than optimal mutation rate of 0.01. Once again a lower Template Update rate produced the best results (p-value < 0.01), although the quality of results does at first decrease until the Template Update rate becomes lower and reverses that trend.

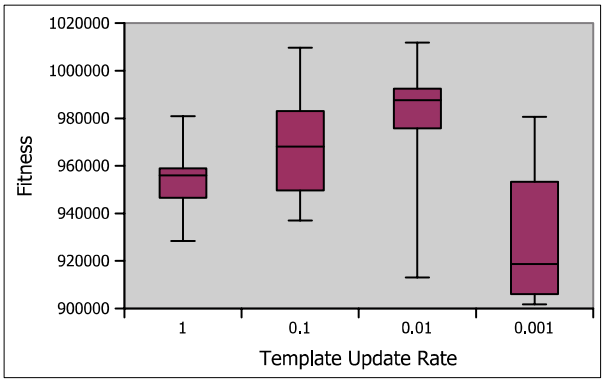

FIGURE 6: Effect of varying replacement rate on solution quality at a low mutation rate for TSP instance nrw1379.

Figure 7 shows the results obtained for the 18512 city TSP over 200,000 generations at the lower than optimal mutation rate of 0.001.

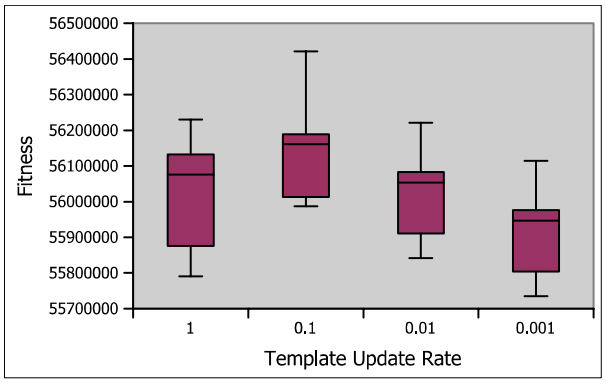

FIGURE 7: Effect of varying replacement rate on solution quality at a low mutation rate for TSP instance d18512.

Figure 8 shows the results obtained for the 26 facility QAP instance over 200,000 generations at the lower than optimal mutation rate of 0.01, although this time the difference in results is less strongly significant (p-value < 0.047)

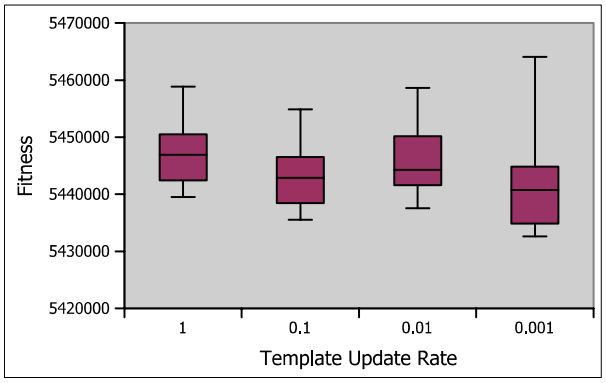

FIGURE 8: Effect of varying replacement rate on solution quality at a low mutation rate for QAP instance bur26a.

In all of the 4 problem instances investigated, a lower Template Update rate produced solutions with a better overall fitness cost. This indicates that using older ancestral templates to restore genetic defects in conditions of lower than optimal mutation frequency results in better convergence towards the optimum. In the case of the bur26a QAP problem, the upper and lower bound of the best result set were equivalent to the upper and lower bound of all the results obtained for that problem set. This indicates that the use of older ancestral templates can serve to introduce much greater genetic diversity into the population in the absence of sufficient diversity via mutation.

3.2 High Mutation Rates

This set of results consists of the experiments conducted at rates of mutation higher than the optimal for problem type. In a typical EA, these conditions would result in too much mutation, reducing the speed of convergence of the population and impairing solution quality [6]. Figure 9 shows the results obtained for the 101 city TSP over 200,000 generations at the higher than optimal mutation rate of 0.5. As with the experiments at a lower Mutation Rate, a lower Template Update rate produced the best results (p-value < 0.01).

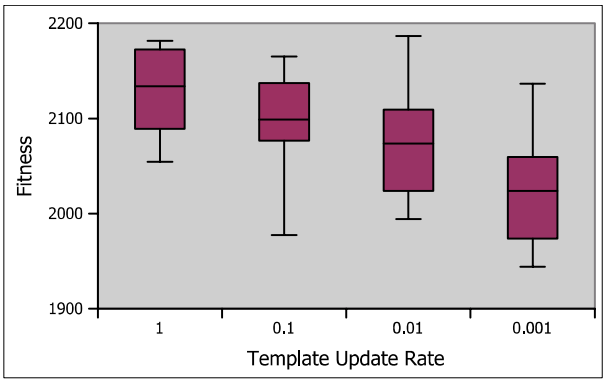

FIGURE 9: Effect of varying replacement rate on solution quality at a high mutation rate for TSP instance eil101.

Figure 10 shows the results obtained for the 1379 city TSP over 200,000 generations at the higher than optimal mutation rate of 0.5. The experiments using the oldest ancestral templates produced the best results (p-value < 0.01).

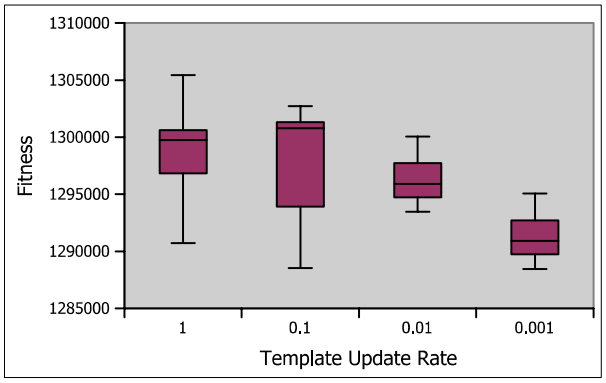

FIGURE 10: Effect of varying replacement rate on solution quality at a high mutation rate for TSP instance nrw1379.

Figure 11 shows the results obtained for the 18512 city TSP over 200,000 generations at the higher than optimal mutation rate of 0.5. The experiments using the oldest ancestral templates produced the best results (p-value < 0.01).

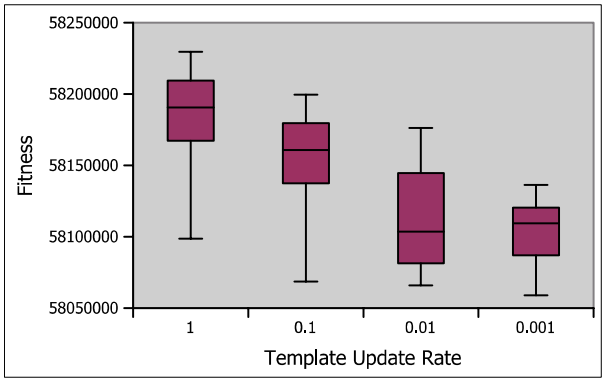

FIGURE 11: Effect of varying replacement rate on solution quality at a high mutation rate for TSP instance d18512.

Figure 12 shows the results obtained for the 26 facility QAP instance over 200,000 generations at the higher than optimal mutation rate of 0.5. In this case the difference between parameters was not enough to be statistically significant. This may be due to the excessively negative effect that a high mutation rate can have on a problem as heavily constrained as the QAP.

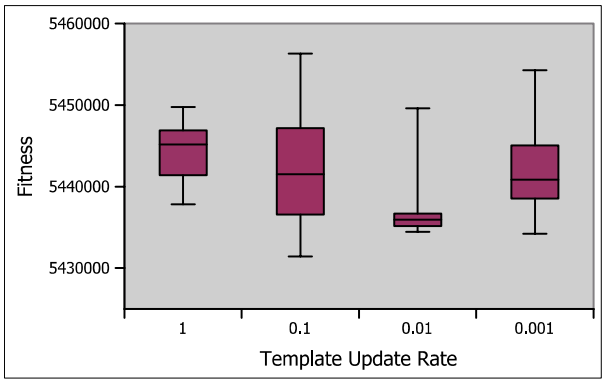

FIGURE 12: Effect of varying replacement rate on solution quality at a high mutation rate for QAP instance bur26a.

In 3 of the 4 problem instances investigated, a lower Template Update rate produced solutions with a better overall fitness cost. This indicates that using older ancestral templates to restore genetic defects in conditions of higher than optimal mutation frequency also results in better convergence towards the optimum.

3.3 Tuned Mutation Rates

This set of results consists of the experiments conducted at rates of mutation that were manually tuned per problem to produce stronger results. Typically, this approach can take a long time and is error prone [14]. For the TSPLIB and QAPLIB libraries, this is an easier task, as these particular problem instances have been extensively studied. Ordinarily, a tuned mutation rate would result in an optimal amount of mutation, maximizing the solution quality produced and minimizing the time required to arrive at a solution. Figure 13 shows the results obtained for the 101 city TSP over 200,000 generations at the tuned mutation rate of 0.1. In this situation there is no statistically significant difference between the test cases.

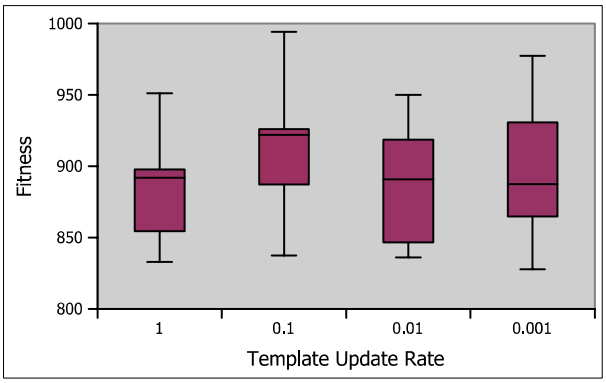

FIGURE 13: Effect of varying replacement rate on solution quality at a tuned mutation rate for TSP instance eil101.

Figure 14 shows the results obtained for the 1379 city TSP over 200,000 generations at the tuned mutation rate of 0.1.

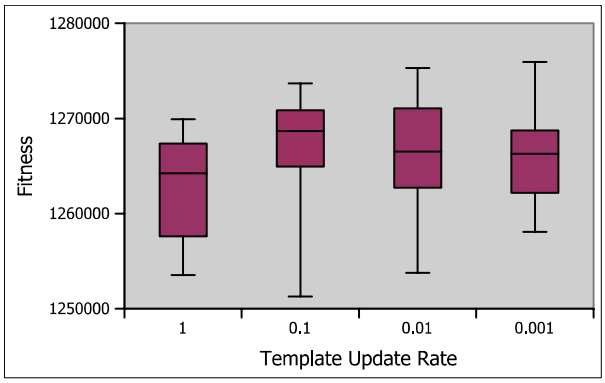

FIGURE 14: Effect of varying replacement rate on solution quality at a tuned mutation rate for TSP instance nrw1379.

Figure 15 shows the results obtained for the 1379 city TSP over 200,000 generations at the tuned mutation rate of 0.1.

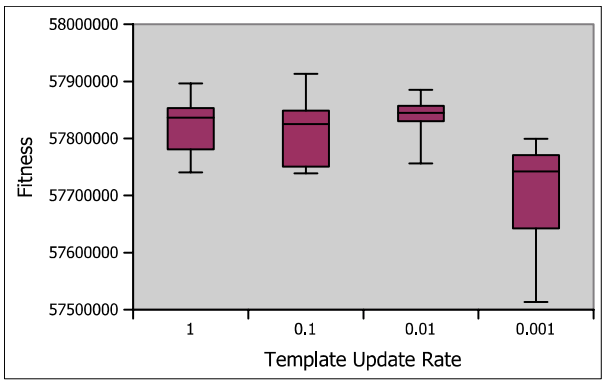

FIGURE 15: Effect of varying replacement rate on solution quality at a tuned mutation rate of 0.01 for TSP instance d18512.

Figure 16 shows the results obtained for the 26 facility QAP instance over 200,000 generations at the tuned mutation rate of 0.1. In this case the lower template update rate corresponding to older ancestral templates produced the best results.

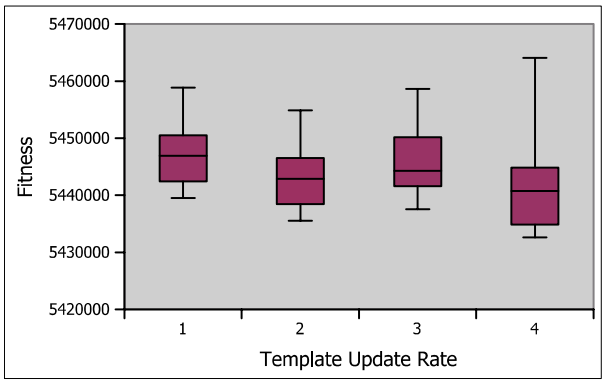

FIGURE 16: Effect of varying replacement rate on solution quality at a mutation rate of 0.1 for QAP instance bur26a.

3.4 Diachronic Analysis of Convergence

Figure 17 shows the results of a diachronic analysis of the improvement of solution quality for the eil101 TSP instance. Corresponding with figure 5, the lowest Template Update rate and therefore the oldest non-Mendelian ancestral templates are producing solutions that reach a better local optimum in fewer generations.

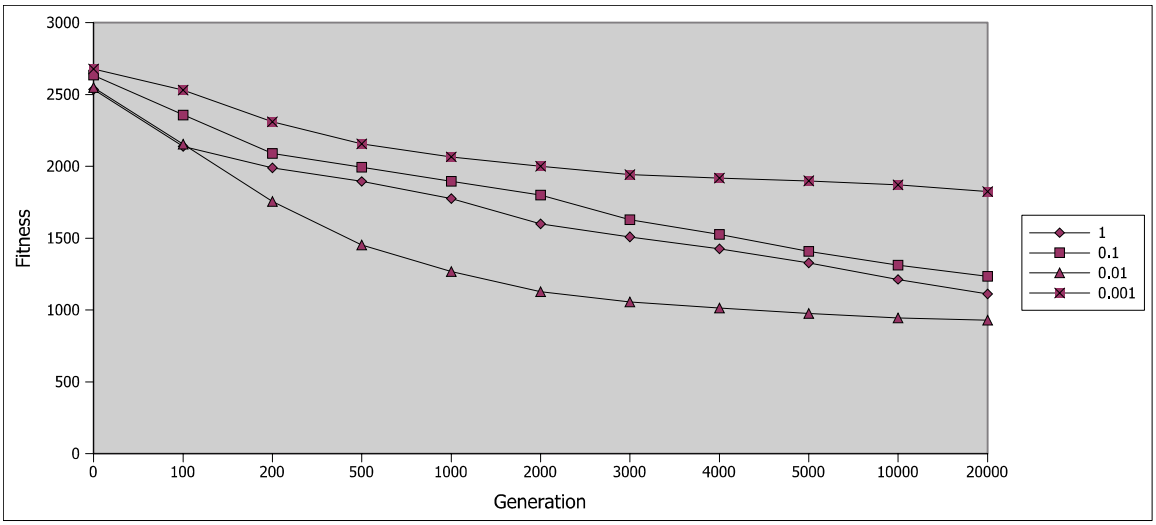

FIGURE 17: Rates of convergence of solutions for 101 city TSP instance by Template Update Rate.

3.5 Retention Policy and Repair Direction

In addition to a series of experiments in which the Mutation Rate and Template Update rate were varied, we evaluated the effect of using a different template retention policy and repair direction on the quality of results obtained. The template retention policy refers to the criteria that are used to select which template out of a possible set should be used for the purposes of Genetic Restoration. The policies investigated were; Best, Worst and Random. With Best, the ancestor with the lowest and therefore best fitness cost was selected to be used as the template in the event that an invalid solution was generated. With Worst, the ancestor with the highest and therefore worst fitness cost was selected to be used as the template and with Random the choice of policy was split between Best and Worst with equal probability of either occurring. Figure 18 gives the results of this comparison, using solutions generated for the nrw1379 TSP instance. There is no statistical significance attributed to deciding to use one policy over another.

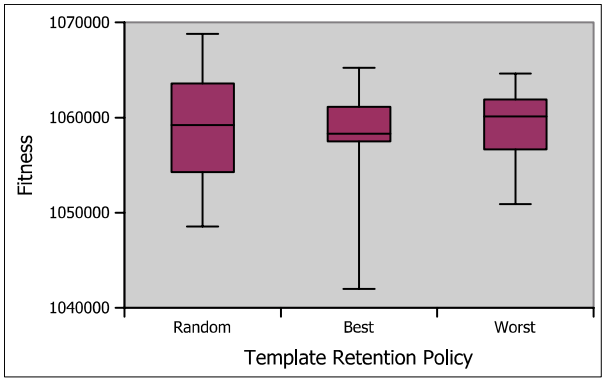

FIGURE 18: Effect of varying template retention policy on solution quality.

Repair direction refers to the order in which repair operations are carried out during the restoration phase. There are 3 choices for repair direction; Left to Right, Right to Left and

Random. The difference between these choices refers to the order in which the repair template is read. Left to Right will scan the repair template from left to right, and if it finds a valid replacement gene it will replace the defective gene with it. Right to Left scans the template in the opposite order. Defective genes are always replaced from left to right; the repair direction refers only to the order in which the repair template is scanned. With a Random strategy the choice of direction is split equally between Left to Right and Right to Left. It is important to note that the difference in repair direction only has an impact in situations where more than 2 genes in the solution are defective, as if only 1 gene is problematic, the order in which repair occurs will have no impact on the choice of replacement gene. Figure 19 shows the results of varying repair direction for instances of the nrw1379 problem, the Random strategy producing the best results with $p < 0.01$. It is likely that the Random strategy is more effective due to the additional diversity introduced to the population by varying repair direction.

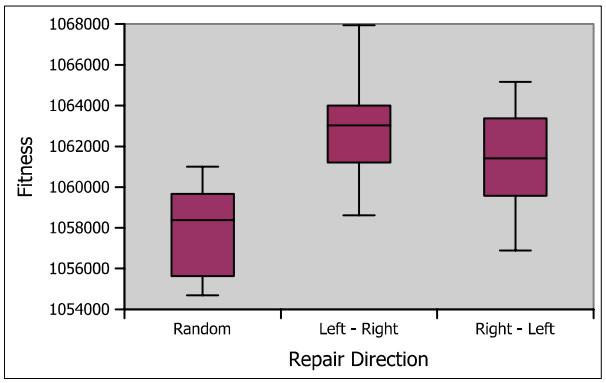

FIGURE 19: Effect of varying repair direction on solution quality.

3.6 Comparison to a Random Template

One alternate approach that has been used in the literature for sourcing repair templates is that of generating a randomized genetic sequence to use as repair data [10]. Figure 20 gives results comparing the effectiveness of random template generation as opposed to ancestral repair templates at low, tuned and high mutation rates respectively. For both low and high mutation rates, the ancestral template update policy produced fitter solutions (p-value < 0.001). At the tuned mutation rate the best results were still obtained with the ancestral repair template but the results were less statistically significant (p-value < 0.05).

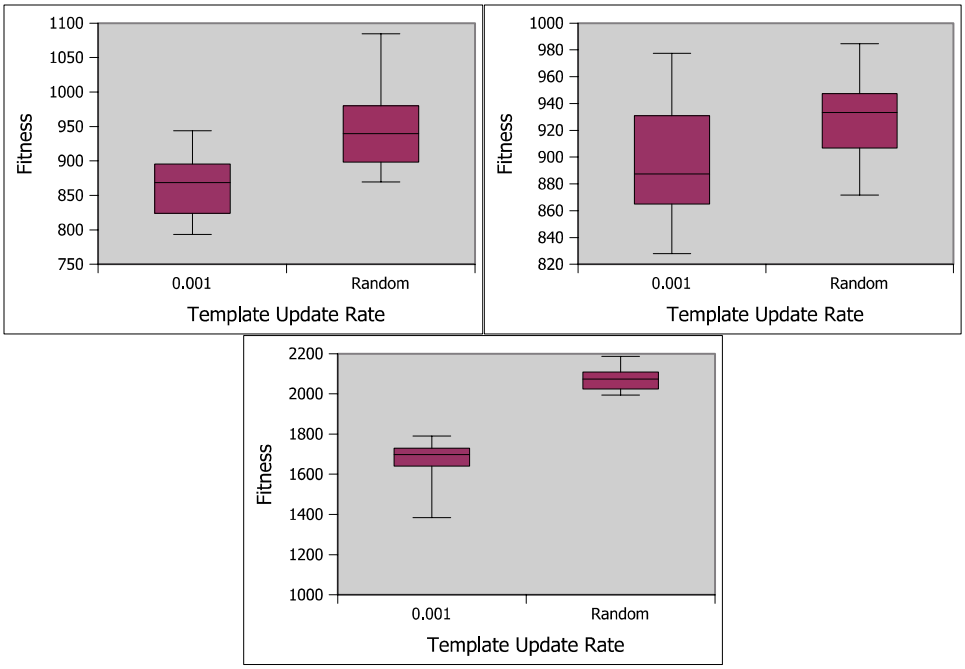

FIGURE 20: Comparison of ancestral template repair vs. random template repair across instances of the 101 city TSP problem at low (top left), tuned (top right) and high (bottom) mutation rates.

3.7 Analysis of Results

The results obtained at rates of mutation outside a tuned rate display a clear pattern. With both lower and higher mutation rates, older non-Mendelian repair templates produced the best results in each experiment. Outside of a small zone where algorithmic parameters such as mutation rate are manually tuned for optimal results, the Mendelian repair template either fails to perform, or performs no better than non-Mendelian ancestral repair. Even within optimal parameter configurations the non-Mendelian approach produces results as good as the Mendelian. We hypothesize that older templates produce fitter solutions than more recent templates for parameter combinations outside of this tuned zone because they exert an implicit clamping factor on the amount of mutations that can occur. In the presence of low mutation rates, the diversity in a population is limited. In practice, this means that the likelihood of ancestral templates introducing diversity is heavily influenced by their genomic distance from the candidate solution. In the presence of high mutation rates, the size of the search space examined is increased, as mutations that result in larger genomic changes are more likely. In these instances, repair is required more often, as shown by Mitchell et al. [9]. In this situation, older ancestral templates are able to correct the genetic defects which appear in greater amounts, while still retaining enough diversity to allow effective exploration of the search space.

3.8 Discussion

Genetic restoration appears to offer a new approach to managing population diversity by leveraging the contents of archived individuals. Ancestral genes are introduced into the current population to increase diversity, with the introduced genes being combined with genes from the current population. Introducing ancestral genes ensures that the included information is reasonably fit, avoiding the penalty associated with introducing true randomness. For complex optimization problem such as those discussed in this paper, we can think of genetic restoration as being somewhat akin to backtracking in traditional search.

Local search algorithms typically use backtracking to overcome local minima and reach the global maximum. An EA implicitly performs this sort of backtracking through the process of generating additional candidate solutions that converge in parallel [6]. Genetic restoration introduces an additional level of backtracking at the level of individual genes rather than at the level of the entire

genome. Implicit backtracking is facilitated by the re-introduction of genes originating from ancestors of the current population – creating mosaic individuals that combine current genes with some ancient genes. As far as the authors are aware this is the first use of ancient ancestral template repair in evolutionary computation, building on [15].

Further research possibilities for ancestral genetic repair include investigating its use to optimize real valued functions as well as deriving alternate approaches to other population based optimization methods such as differential evolution or simulated annealing. Additionally, ancient repair templates show good performance across a wide range of mutation rates, indicating that they may have a potential application in automatic parameter tuning.

4. CONCLUSIONS

In 2005, Lolle et al. published work [1] that indicated the possible existence of non-Mendelian inheritance in Arabidopsis thaliana. This repair mechanism makes use of ancestral genetic information to repair genetic errors in individual plants. We investigate an application of this strategy to handling constraint violations in an Evolutionary Algorithm (EA). In our EA, each individual records a collection of ancestral templates for the purposes of correcting genetic defects in the form of invalid candidate solutions. Results are presented relating to three instances of the Traveling Salesman's Problem (TSP) and one instance of the Quadratic Assignment Problem (QAP).

Some general observations can be made across the results. First, at relatively low rates of background mutation, where a typical EA suffers with stagnation due to a lack of genetic diversity, the best results were consistently produced by the most ancient non-Mendelian repair templates (around 1000 generations old). Second, at relatively high rates of background mutation, where a typical EA suffers from too much diversity, the best results were again produced by older repair templates. Third, when mutation rates were tuned by hand to provide a good rate of convergence for the problem instances examined, older non-Mendelian repair templates produced results that were as good as the Mendelian repair templates.

Overall our results indicate that older (and thus non-Mendelian) ancestral repair templates provide a better general-purpose repair mechanism than the Mendelian alternative. On problems for which optimal mutation rates are unknown, using ancient repair templates produces reliably better results. It appears that the strategy proposed in [1] is surprisingly effective across a range of problems and conditions.

5. REFERENCES

1. S. J. Lolle, J. Victor, J. Young, and R. Pruitt, "Genome-wide non-mendelian inheritance of extra-genomic information in arabidopsis," *Nature*, vol. 434(1), pp. 505–509, 2005.

2. D. Weigel and G. Jurgens, "Genetics: Hotheaded healer," *Nature*, vol. 434(443), pp. 443–443, 2005.

3. P. Peng, S. Chan, G. Shah and S. Jacobsen, "Increased outcrossing in hothead mutants." *Nature*, vol. 443, pp. E8–E8, 2006.

4. R. Mercier, S. Jolivet, J. Vignard, S. Durand, J. Drouaud, G. Pelletier, and F. Nogue, "Outcrossing as an explanation of the apparent unconventional genetic behavior of arabidopsis thaliana hth mutants," *Genetics*, vol. 180(4), pp. 2295–2297, Dec. 2008.

5. M.T. Hopkins, A.M. Khalid, P.C. Chang *et al.* (2013) "*De novo* genetic variation revealed in somatic sectors of single Arabidopsis plants", *F1000Research* vol. 2 no. 5, 2013,

6. D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

7. C. Darwin, "On the origin of species", New York :D. Appleton and Co, 1861

8. S. Salcedo-Sanz, "A survey of repair methods used as constraint handling techniques in evolutionary algorithms," *Computer Science Review*, vol. 3, pp. 175–192, 2009.

9. G. G. Mitchell, D. O'Donoghue, and A. Trenaman, "A new operator for efficient evolutionary solutions to the travelling salesman problem," in *Applied Informatics*, 2000.

10. D. Lichtblau, "Discrete optimization using Mathematica," in *Proceedings of the World Conference on Systemics, Cybernetics, and Informatics (SCI 2002)*, vol. 16, 2002.

11. G. G. Mitchell, D. P. O'Donoghue, D. Barnes, and M. McCarville, "Generepair - a repair operator for genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2003, pp. 235–239.

12. G. Reinelt, "Tsplib - a travelling salesman problem library," *ORSA Journal of Computing*, vol. 3, pp. 376–384, 1991.

13. R. E. Burkard, S. E. Karisch, and F. Rendl, "Qaplib - a quadratic assignment problem library," *J. of Global Optimization*, vol. 10(4), pp. 391–403, Jun. 1997.

14. Z. Michalewicz and M. Schmidt, "Evolutionary algorithms and constrained optimization," in *Evolutionary Optimization*, R. Sarker, M. Mohammadian, and X. Yao, Eds. New York: Kluwer Academic Publishers, pp. 57–86, Feb. 2002, .

15. A. Fitzgerald and D. P. O'Donoghue, "Genetic repair for optimization under constraints inspired by Arabidopsis thaliana," in *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature (PPSN)*, pp. 399–408, 2008,.