

Population Diversity in Permutation-Based Genetic Algorithm

Kenny Q. Zhu¹ and Ziwei Liu²

¹ Department of Computer Science, National University of Singapore, Singapore 119260

kzhu@comp.nus.edu.sg

² Advantech Singapore Pte Ltd, #03-06 Technopark@Chaichee, Singapore 469004

lziwei@hotmail.com

Abstract. This paper presents an empirical study of population diversity measures and adaptive control of diversity in the context of a permutation-based algorithm for Traveling Salesman Problems and Vehicle Routing Problems. We provide detailed graphical observations and discussion of the relationship among the four diversity measures and suggest a moderate correlation between diversity and search performance under simple conditions. We also study the effects of adapting key genetic control parameters such as crossover and mutation rates on the population diversity. We are able to show that adaptive control of the genetic operations based on population diversity effectively outperforms fixed parameter genetic algorithms.

1 Introduction

Traditional genetic algorithms (GA) often suffer from loss of diversity due to premature convergence of the population. As a result, the search is trapped in local optima. Hence, the maintenance of diversity is one of the most fundamental issues of GA. Previous studies on population diversity can be divided into two categories: *diversity measures* and *maintenance of diversity*. A large amount of work has been devoted to diversity measures, which includes early study of variance of fitness [11,3], and uncertainty [3]. Recently, other measures such as evolution history [14], distance [2], epistasis [7] and measures in the phenotype and genotype space [15] have also been introduced. A survey of population diversity measures in genetic programming (GP) can be found in [5]. Work on diversity maintenance includes crowding and preselection [13], self-adapting mutation rates [9], etc. Some studies have been devoted to adaptive GA and population diversity control. A good survey about aspects of adaptive GA can be found in [12]. Parameter control in general evolutionary algorithms is discussed in a recent survey by Eiben, *et al.* [1], which includes references to other works on self-adaptation in GA.

The GA we are concerned with in this paper is one in which individual chromosomes are integer encoded, and the crossover operations are *permutation-based* [10]. In this type of GA, all individuals have the same set of distinct alleles (integers) in all generations. The different permutations of the individuals decode into different fitness values. Permutation-based GA is used to solve Traveling Salesman Problem (TSP) [8], Vehicle Routing Problem (VRP) [4] and its variant Vehicle Routing Problem with Time Windows (VRPTW) [16], and many other problems. TSP can be defined as: given a set

of towns and the distances between them, determine the shortest path starting from a given town, passing through all the other towns *exactly* once and returning to the first town. The objective of VRP/VRPTW is to find routes for vehicles to service all the customers at a minimal cost (in terms of number of routes and total distance traveled), without violating the capacity and the travel time constraints of the vehicles and, in the case of VRPTW, the time windows imposed by the customers. Both classes of problems are in the NP domain.

We define and compare four important diversity measures, namely phenotypes, genotypes, standard deviation of fitness and ancestral ids. These measures represent diversity from different angles, hence the behaviors are also different. We perform a comprehensive empirical study on the effects of genetic operations such as crossover and mutation on the population diversity. A simple adaptive control function is applied to maintain diversity at desirable levels through automatically varying application rates of genetic operators. Benchmarks show that adaptive diversity control is able to strike a balance between global exploration and local exploitation, and outperforms traditional fixed parameter GA's.

2 Canonical Algorithm

The basic algorithm for solving TSP/VRP uses a fixed-length integer-string representation for encoding, and a heuristic to decode the chromosomes into fitness values. The algorithm starts with an initial population of 50 random individuals unless otherwise stated, and selects individuals for reproduction. After reproduction through a number of operations, the new population replaces the whole parent populations to complete one generation. The algorithm runs for a fixed number of generations. We briefly introduce some elements of this algorithm below.

The representation of a solution is a string of distinct integers of length K , where K is the number of customers (cities). The string is known as a *chromosome*, whose length is K . Each *gene* (integer) of the chromosome is a customer's identifier. For example,

$$3 - 2 - 4 - 5 - 9 - 8 - 7 - 10 - 6 - 1 - 12 - 11$$

A problem-specific algorithm [17] is used to decode the string into solutions and to compute the fitness. A binary *tournament selection* mechanism is used in this algorithm. Three commonly used order-based *crossover* operators are applied to the mating chromosomes. They are *Partially Matched Crossover (PMX)*, *Order Crossover (OX)* and *Cycle Crossover (CX)* [10]. Each will be applied independently in our experiments. The probability of applying crossover operator to a pair of mating individuals is denoted by p_c .

We use a *sequence insertion* mutation, which is defined as relocating a subsequence from one position of the chromosome to another position. Mutation rate is denoted by p_m . Besides mutation, we use a *post-recombination* operator called *random immigrants* [6], that randomly generates chromosomes to replace randomly selected existing chromosomes. Random immigrants is applied at a rate of p_r .

The algorithm can be summarized as follows.

- GA-1** Initialize the population P with N chromosomes;
GA-2 Decode chromosomes to obtain fitness values in the population. Set crossover rate p_c , mutation rate p_m and random immigrants rate p_r ;
GA-3 Create a new population by repeating the following steps:
 1. Select two parent chromosomes from a population by Tournament Selection;
 2. With a probability p_c , crossover the parents to form two new children, otherwise copy the parents to become offspring;
 3. With probability p_m , mutate the new offspring;
 4. Place the new offsprings in a new population;
GA-4 Replace the old population with the newly generated population;
GA-5 Do random immigrants if required;
GA-6 If the stop criterion is satisfied, stop; else go to GA-2.

3 Diversity Measures

Four diversity measures, namely, the number of unique *phenotypes*, the fitness *standard deviation*, the total distance among *genotypes*, and the number of unique *ancestral ids*, are compared and studied in this paper. In what follows, P is a set of all chromosome sequences in a population, N is the number of sequences in the population, and K is the length of the sequences in P .

Phenotypes (ptype) The number of unique fitness values in the population, normalized between 0 and 1. The computation takes $O(N \log N)$ time.

$$ptype(P) = \frac{|U| - 1}{N - 1}, \quad (1)$$

where $U \subseteq P$, and $\forall f_i \in U$ and $\forall f_j \in U$, $f_i \neq f_j$ if $i \neq j$.

Standard deviation (stddev) The standard deviation of fitness values in a population ($O(N)$):

$$stddev(P) = \sqrt{\frac{\sum_{i=1}^N (f_i - \bar{f})^2}{N - 1}}, \quad (2)$$

where N is the population size and f_i is fitness of the i th individual.

Genotypes (gtype) The sum of the *edge distances* between any two genotypes (individual strings). Let s be an integer sequence that represents a genotype. We define $A(s)$ to be a set of arcs in s . The edge distance between genotype u and v is defined as:

$$\mathcal{D}_e(u, v) = |A(u) \setminus A(v)|, \quad (3)$$

In other words, edge distance is defined as the number of arcs in u but not in v . Note that this number is equal to the number of arcs in v but not in u . The edge distance here is different from the commonly used edit distance for measuring similarity among sequences. This is because we are trying to measure the similarity in subsequences, which is key in TSP and VRP.

$$gtype(P) = \frac{\sum_{i \neq j} \mathcal{D}_e(P[i], P[j])}{(K - 1)(N - 1)N}, \quad (4)$$

where $P[i]$ and $P[j]$ are the i th and j th genotypes in P . This takes $O(N^2 K^2)$.

Ancestral id (uid) Each individual in the initial population has a unique id. During crossover, two parents produce two children. One of the children inherits the mother’s uid and the other inherits the father’s uid. One’s uid changes when it’s mutated or being replaced in random immigrants procedure.

Unless otherwise noted, the benchmark problem used in this paper is R101 from Solomon’s 100-node VRPTW problem set [16]. Problem R101 is characterized by randomly distributed customer locations and short time windows for each customer. It is one of “harder” problems in the problem set that have not yielded the optimal solution as far as we know. Extract methods are required to obtain the optimal solution but the cost is prohibitive.

With a fixed, 100% random initial population, and PMX crossover only, the basic algorithm was run 10 times with the parameters: $p_c = 0.6$, $p_m = 0$ and $p_r = 0$. In each generation, the population diversity is recorded by four measures defined above. Fig. 1 through Fig. 4 demonstrate the natural evolution of these measures over 201 generations, without any mutation.

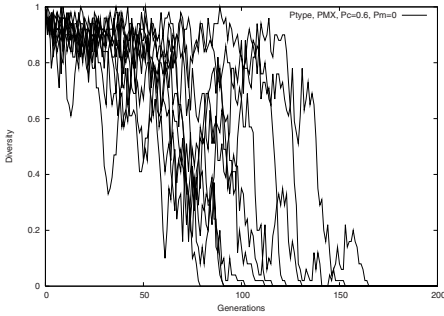


Fig. 1. Evolution of Ptype over generations

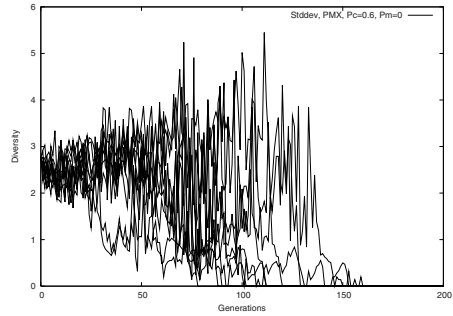


Fig. 2. Evolution of Stdev over generations

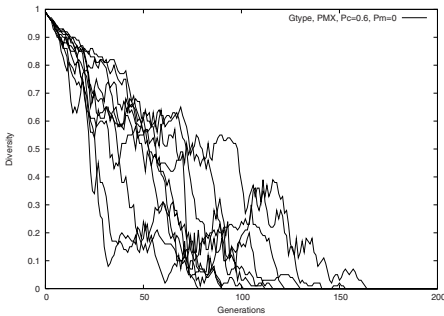


Fig. 3. Evolution of Gtype over generations

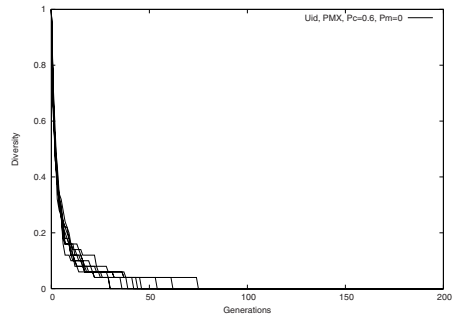


Fig. 4. Evolution of Uid over generations

Both phenotype measure and standard deviation measure displays a rather steep phase transition at about the 100th generation and quickly converges to zero, except stdev is more volatile. These two measures are similar because both are based on

fitness values. Standard deviation measure is more sensitive to the variance in the population, therefore the large fluctuation is observed. Genotype decreases more gradually but the descent happens right from the beginning, unlike ptype and stddev. This can be explained because recombination causes convergence at the genotype level first, before such change is reflected in the fitness values. The gradual descent of gtype measure also suggests this measure can be more useful in early prediction and diversity control. The rapid convergence of uid is a certainty because, with no mutation, the number of ids will monotonically decrease. The selection mechanism (in which the fitter individuals are preferred in mating) accelerates the decrease in unique ids.

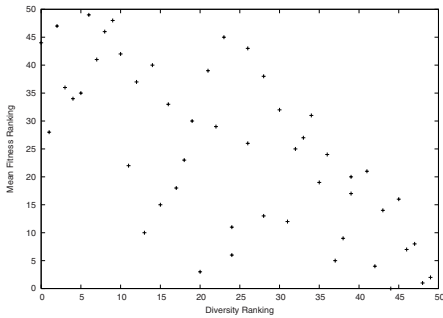


Fig. 5. Ptype rankings vs mean fitness rankings

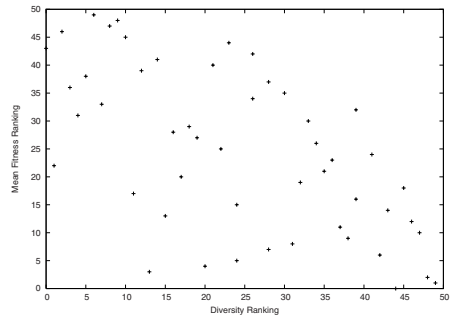


Fig. 6. Stddev rankings vs mean fitness rankings

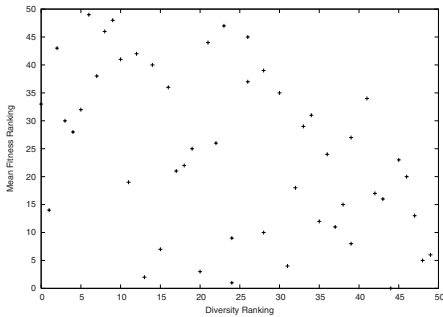


Fig. 7. Gtype rankings vs mean fitness rankings

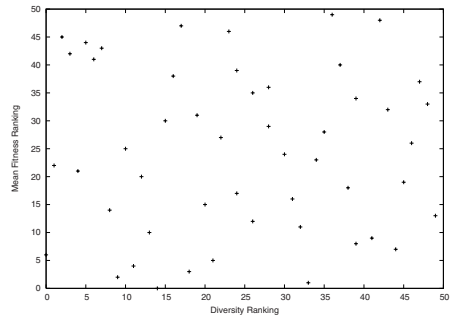


Fig. 8. Uid rankings vs mean fitness rankings

We ran the algorithm to 201 generations for 50 times, and plot the rankings of accumulated diversity over 201 generations against the rankings of the mean fitness at the 201st generation, for every diversity measure in Fig. 5 through 8. Notice in the case of VRPTW, the fitness value is taken to be the total cost of a solution which is a combination of the number of routes and the total distance. In these figures, both diversity and fitness are ranking in ascending order.

Some moderate, “negative” correlation can be seen from all plots except that of uid. In Fig. 5 through Fig. 7, dots are clustered near the upper-left to lower-right diagonal line. In other words, the general trend is: the higher the diversity, the smaller the fitness

value, or the better the search quality. The correlation is particularly evident in *ptype* as it is defined by fitness values only. The randomness displayed in Fig. 8 is again the result of early convergence of *uid* measure (Fig. 4).

The consistency of the four diversity measures can be estimated by taking the standard deviation of the average diversity over the 50 runs. The standard deviations are $\sigma_{ptype} = 0.1004$, $\sigma_{gtype} = 0.0557$, $\sigma_{stddev} = 0.334$, and $\sigma_{uid} = 0.0068$. *Stddev* measure values were normalized to 1 before the calculation. *Gtype* and *uid* measures appear more consistent than others, because of their low deviation. The very small deviation of *uid* is due to its very premature convergence. The volatility of the *stddev* measure shown in the graph contributes to its high variance and it is the least reliable measure of the four.

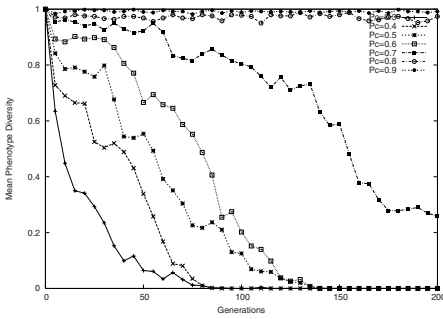


Fig. 9. Effect of PMX on Ptype

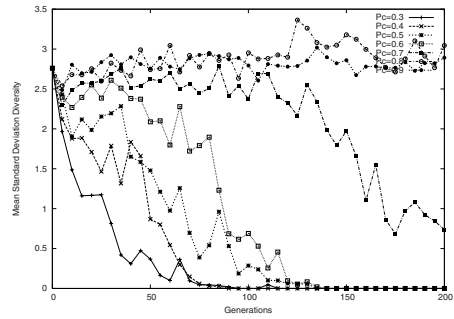


Fig. 10. Effect of PMX on Stddev

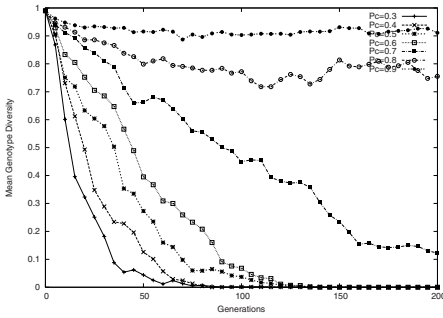


Fig. 11. Effect of PMX on Gtype

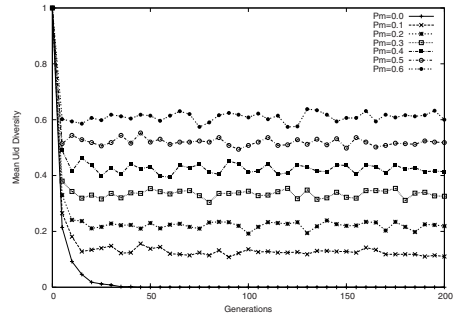


Fig. 12. Effect of mutation on Uid

4 Effects of Genetic Operators on Diversity Measures

Methods of maintaining population diversity generally come in two categories: methods based on the selection process and those based on genetic operators. In this paper, the second approach is adopted. We hope to control the diversity through the three common genetic operators defined in Section 2. Three sets of experiments with the following

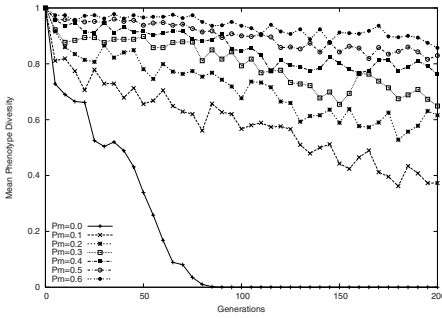


Fig. 13. Effect of mutation on Ptype

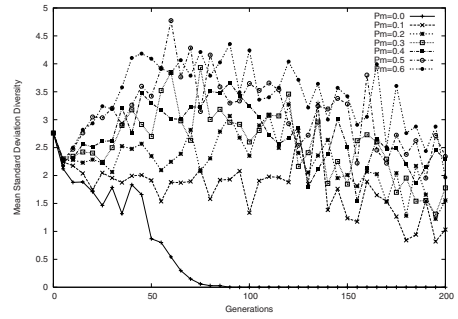


Fig. 14. Effect of mutation on Stdev

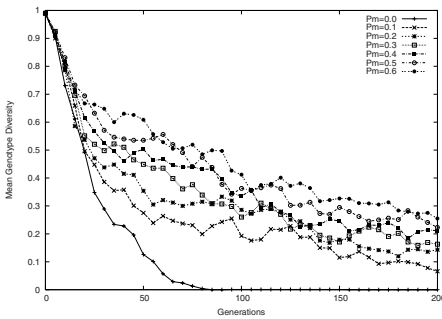


Fig. 15. Effect of mutation on Gtype

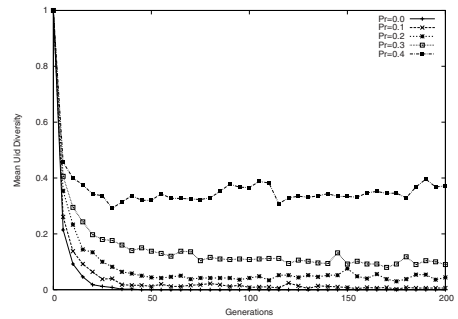


Fig. 16. Effect of random immigrants on Uid

parameters were first conducted to demonstrate the independent effect of crossover, mutation and random immigrants on diversity.

Crossover: $p_c = 0.3 \dots 0.9$ (PMX), $p_m = 0$, $p_r = 0$ at steps of 0.1.

Mutation: $p_c = 0.4$ (PMX), $p_m = 0 \dots 0.60$, $p_r = 0$ at steps of 0.1.

Random immigrants: $p_c = 0.4$ (PMX), $p_m = 0$, $p_r = 0.1 \dots 0.4$ at steps of 0.1.

Each set of parameters were tested 10 times and the mean diversity at each of the 201 generations was recorded by all four measures. We then plot the convergence graph of the mean diversity for each diversity measure under different parameter settings, and these are included in Fig. 9 through Fig. 19. Uid diversity is only affected by mutation and random immigrants as only these two operators contribute new ids into the system, therefore a plot of crossover effect on uid is not included.

One can observe from these plots that all three operators promote diversity by all measures. With increasing crossover rate p_c (Fig. 9, 10 and 11), The diversity curves are spread out almost evenly, which suggests p_c is a good tool in controlling the diversity. For lower crossover rates (up to 0.8), population eventual converges; for higher p_c , our experiments show that population remains highly diversified even over many more generations. It is noted, however, when diversity drops below certain threshold, crossover alone is not able to reverse the converging trend. This is especially evident in the ptype

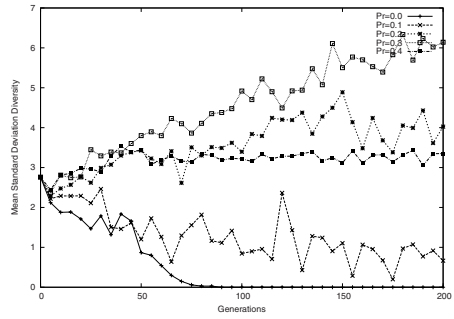
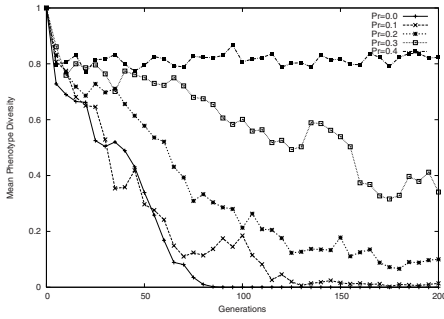


Fig. 17. Effect of random immigrants on Ptype **Fig. 18.** Effect of random immigrants on Stddev

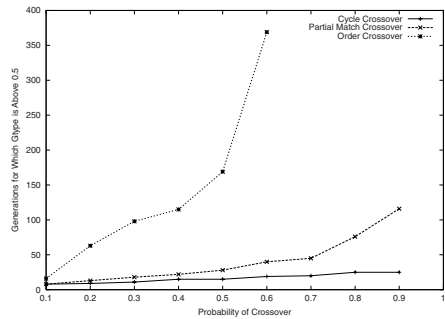
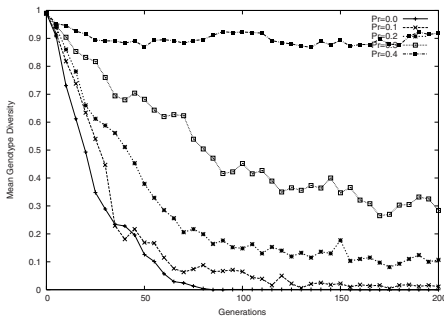


Fig. 19. Effect of random immigrants on Gtype **Fig. 20.** Comparison of OX, PMX and CX

and stddev measures, where the diversity curve is generally flat in the beginning, and turns down sharply after some transition point.

The effect of various mutation rates on the diversity is visibly different from the crossover. Fig. 12 shows that varying the mutation rate effectively sets the uid diversity at corresponding levels, which is easy to understand, given the definition of uid. Fig. 9 to Fig. 11 show a big jump in the diversity levels (in all three measures) from $p_m = 0$ to $p_m = 0.1$, but the differences become less significant as p_m increases. This is due to the fact that mutation happens in genotype level and the changes are always localized, therefore the effect is limited. Stronger diversifying forces can be seen in Fig. 16 through Fig. 19, except excessive random immigrants can drag down the overall fitnesses of the population uniformly, which causes some anomaly at $p_r = 0.4$ in Fig. 18. It is easy to prove that the population does not converge to uniformity when p_m or p_r is non-zero. We can classify the use of random immigrants, crossover and mutation as diversification at the population, phenotype and genotype levels, respectively. Therefore, a combination of all three operations is expected to give a more balanced and consistent result in promoting diversity.

Fig. 20 shows the comparison among the three crossover operators. OX, PMX and CX are applied to random population at p_c from 0.1 to 0.9, and the number of generations when the gtype diversity reaches 0.5 is recorded for each setting. Apparently, OX is the most effective in boosting diversity while CX is the least effective.

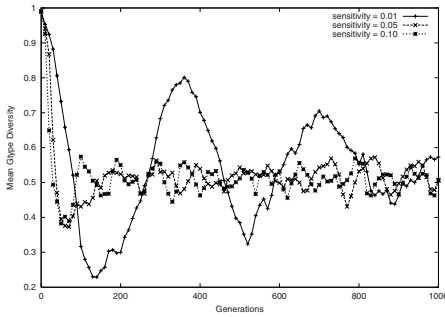


Fig. 21. Oscillation at various sensitivities ($d_t = 0.5$)

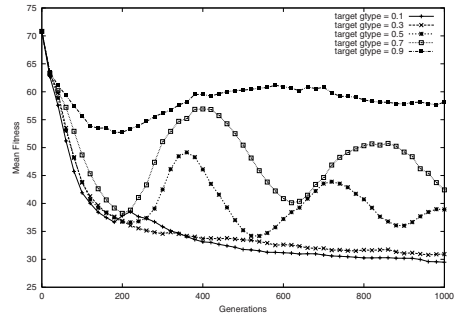


Fig. 22. Mean fitness progression under target diversities 0.1 - 0.9 at step 0.2 (OX)

5 Adaptive Control

From the previous sections, the positive correlation of diversity and search quality, along with the fact that genetic operations promote diversity, motivate us to maintain diversity at some healthy level so that more promising regimes in the search space can be explored. One interesting way to control the diversity is to adaptively calibrate the crossover/mutation rates against the changing population diversity. To illustrate the benefits of adaptive control, we apply the following simple adaptive function on the rates of crossover, mutation or random immigrants to maintain diversity at a target level.

$$p' = \max(p_{\min}, \min(p_{\max}, p(1 + \frac{\xi(d_t - d)}{d}))), \quad (5)$$

where p is the current rate of genetic operations, p' is the new rate in the next generation, d is the diversity of current population, d_t is the target diversity, ξ is the control sensitivity, and p_{\min} , p_{\max} are the lower and upper bounds of the rate.

In every generation of the algorithm, the crossover and mutation as well as random immigrants rates are recomputed by (5), and the new rates will be used in the recombination phase of that generation. The choice of (5) is based on the belief that the population is further diversified when genetic operations are applied at higher probability. So when d falls below d_t , the new rate p' is somewhat increased in (5). Contrary to that, if d becomes larger than d_t , probability will be decreased until d comes in line with d_t . A small ξ means gradual change in the rate, and that also translates into slower fluctuation of the diversity, also known as the oscillation of the feedback control (Fig. 21). Larger ξ causes the population diversity to follow more closely to the target diversity. The oscillation, instead of monotonic convergence or divergence, allows the population to concentrate on one region to optimize for a while, and then move on to other regions. While (5) may appear simplistic and arbitrary, our experiments below already demonstrate the clear advantage of the adaptive control. The search for better adaptive functions remains an open area of research.

Setting the target diversity d_t at 0.1 to 0.9, and using OX crossover alone, we ran the basic algorithm five times to examine the significance of d_t . The progression of the mean fitness at various target gtype levels is plotted in Fig. 22. When diversity is maintained at low levels, the population converges rather monotonically. When target

diversity rises above 0.5, the population mean fitness starts to fluctuate, although such fluctuation appears to subside with time. Because the mean fitness curve approximately portrays the search horizon, a hovering curve essentially represents the traversal across a terrain of peaks and valleys in the solution space. In other words, the population is sufficiently diversified to explore a number of localities.

However, we also observe that the higher diversity levels generally result in higher fitness values or worse search result. This is because our canonical GA has only limited capability of local optimization, as the algorithm traverses across many domains. Thus we propose to use some local search techniques such as greedy descent to exploit the “good areas” further, as a complement to the global exploration provided by diversity adaptive control. Fig. 22 also indicates that the oscillation is diminishing under overly high diversity settings, e.g. $d_t \geq 0.9$. This is undesirable because the algorithm does not have the chance to zoom in to some of the prospective good neighborhood. The optimal value for d_t is problem-specific and needs to be tuned for a class of problems.

In the last experiment, we compare the solution quality of adaptive GA and a fixed-parameter GA which is based on our canonical algorithm. The benchmark problems are the full set of Solomon’s VRPTW 100-node problems. Now, a mixed initial population of both random and good solutions is used. The good individuals are obtained from Push Forward Insertion Heuristic (PFIH) [16] and its *2-neighborhood*. We will focus on the gtype diversity as it behaves most consistently and is easy to control. OX is used as the only crossover operator. The initial crossover and mutation rates are 0.77 and 0.1, respectively. These initial rates are obtained from some VRPTW literature and are known to be standard.

We test-ran both the fixed parameter GA and the adaptive GA up to 500 generations, with target diversity $d_t = 0.5$ and $\xi = 0.01$. The reason for selecting $d_t = 0.5$ is to take advantage of the oscillation. In the fixed-parameter GA, $p_c = 0.77$ and $p_m = 0.1$. Our preliminary tests indicate that these settings appear to be the best for fixed-parameter GA.

The average results over 10 runs are compared in Table 1. Columns marked “Fixed” and “Adaptive” record the solutions on the number of routes and total distance traveled, in the two GA schemes tested. The smaller these numbers are, the better the solution. Clearly, with the target diversity at 0.5, our adaptive GA consistently outperforms fixed-parameter GA in terms of the quality of solutions, in all categories of VRPTW. The average execution time, however, is somewhat longer than the fixed algorithm. This is expected as the adaptive algorithm normally traverses more regions in the search space.

Table 1. Fixed parameter vs. adaptive algorithm

Category	p_c/p_m	Fixed	Time	Category	Adaptive	Time
C1	0.88/0.49	10/835.6	72	C1	10/828.9	80
C2	0.89/0.36	3/610.9	804	C2	3/589.9	735
R1	0.85/0.56	13.3/1263.4	131	R1	12.8/1242.7	305
R2	0.84/0.48	3.2/1021.4	1288	R2	3/1016.4	1308
RC1	0.85/0.57	13.1/1437.2	177	RC1	13/1412.0	239
RC2	0.84/0.49	3.9/1249.7	765	RC2	3.7/1201.2	883

References

1. Ágoston E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141, 1999.
2. A. L. Barker and W. N. Martin. Dynamics of a distance-based population diversity measure. In *Proceedings of the 2000 Congress on Evolutionary Computation*, pages 1002–1009.
3. M. A. Bedau, M. Zwick, and A. Bahm. Variance and uncertainty measures of population diversity dynamics. *Advances in Systems Science and Applications Special Issue*, 1:7–12, 1995.
4. L. Bodin, A. Assad, and M. Ball. *Routing and Scheduling of Vehicles and Crews - the State of the Art*. Pergamon Press, 1983.
5. E. Burke, S. Gustafson, and G. Kendall. A survey and analysis of diversity measures in genetic programming. In *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 716–723.
6. H. G. Cobb and J. J. Grefenstette. Genetic algorithms for tracking changing environments. In *Proceedings of International Conference on Genetic Algorithms*, pages 523–530, 1993.
7. C. Cotta and J. M. Troya. Using dynastic exploring recombination to promote diversity in genetic search. In *Proceedings of 6th International Conference on Parallel Problem Solving from Nature - PPSN VI*, pages 325–334. Springer, 2000.
8. G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large-scale traveling-salesman problem. *Operations Research*, 2, 393–410.
9. M. Glickman and K. Sycara. Reasons for premature convergence of self-adapting mutation rates. In *Proc. of the 2000 Congress on Evolutionary Computation*, pages 62–69.
10. D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Pub. Co., 1989.
11. D. E. Goldberg and M. Rudnick. Genetic algorithm and the variance of fitness. *Complex Systems*, 5(3):265–278, 1991.
12. F. Herrera and M. Lozano. Adaptation of genetic algorithm parameters based on fuzzy logic controllers. *Genetic Algorithms and Soft Computing*, pages 95–125, 1996.
13. S. W. Mahfoud. Crowding and preselection revisited. In *Parallel problem solving from nature 2*, pages 27–36, 1992. North-Holland.
14. N. F. McPhee and N. J. Hopper. Analysis of genetic diversity through population history. In *GECCO99: Proceedings of the Genetic and Evolutionary Computation Conference*, 1999.
15. R. W. Morrison and K. A. De Jong. Measurement of population diversity. In *5th International Conference, Evolution Artificielle, EA 2001*, pages 31–41. Springer, 2001.
16. M. M. Solomon. Algorithms for vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2), 1987.
17. K. Q. Zhu. A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time windows. In *Proceedings of International Conference on Tools with Artificial Intelligence (ICTAI 2003)*, pages 176–183.