



LCS Based Diversity Maintenance in Adaptive Genetic Algorithms

Ryoma Ohira^(✉), Md. Saiful Islam, Jun Jo, and Bela Stantic

School of Information and Communication Technology, Griffith University,
Southport, QLD 4215, Australia

{r.ohira,saiful.islam,j.jo,b.stantic}@griffith.edu.au

Abstract. A genetic algorithm (GA) experiences premature convergence when the diversity is lost in the population. Adaptive GAs aim to maintain diversity in the population by trading off a balance between exploring the problem space and exploiting known solutions. Existing metrics for population diversity measures only examine the similarity between individuals on a genetic level. However, similarities in the order of genes in individuals in ordered problems, such as the travelling salesman problem (TSP) can play an important role in effective diversity measures. By examining the similarities of individuals by the order of their genes, this paper proposes longest common subsequence (LCS) based metrics for measuring population diversity and its application in adaptive GAs for solving TSP. Extensive experimental results demonstrate the superiority of our proposal to existing approaches.

Keywords: Adaptive GA · LCS · Diversity maintenance · TSP

1 Introduction

Genetic Algorithms (GAs) find and develop solutions through exploring the global search space to look for good solutions and then exploiting found solutions to discover optimal solutions. By maintaining a balance between exploration and exploitation, a GA attempts to search and converge on the global optima. However, a simple GA has a tendency to converge to a local optima without sufficiently exploring the search space. This premature convergence is often attributed to a loss of diversity through selective pressure, mutation rates and crossover between similar individuals. Identifying optimal settings for these parameters for maintaining population diversity is a long-standing challenge for GAs [1]. While studies [2, 3] have shown that a GA can have a significant performance increase when parameter controls are applied in an adaptive and on-line manner, the success of adaptive controls are reliant on performance and health measures. With GAs, the no free lunch (NFL) theorem is presented in how a given set of parameter control mechanisms may not be optimal, or even applicable, to another problem [1]. One sub-set of optimisation problems are ordered problems where the order of nodes in a solution is critical to the quality of the

solution. An example of this is the travelling salesman problem (TSP) where a salesman must travel to n number of cities and the objective is to find the shortest route possible with the salesman visiting each city once.

1.1 Related Works

Early diversity metrics simply calculate the Hamming distance between individuals [4, 5]. Ursem et al. [6] extend [5] and propose a Hamming distance based diversity metric considering the size of the population and problem. Mc Ginley et al. [7] propose a novel healthy population diversity (HPD) metric that examines the diversity of a population from a fitness perspective for their ACROMUSE GA. ACROMUSE utilises both a standard population diversity (SPD) measure similar to earlier works and the proposed HPD to demonstrate how the relationship between a healthy, or highly fit, population and a diverse population can contribute to maintaining a diverse population. By introducing the HPD, ACROMUSE outperforms other adaptive GAs that use diversity based adaptive parameter controls in benchmark tests. However, ACROMUSE and its competitors were all based on diversity metrics that only measured the gene-wise similarity between individuals and tested with general optimisation problems. Experiments demonstrate lower performance gains when ACROMUSE and its gene-wise approach to measuring population diversity is applied to the TSP.

Many works make novel use of diversity metrics based on the works [8–10], however, they do not offer new metrics for measuring population diversity. While general diversity measures and adaptive GAs are an improvement over traditional GAs [4–11] for ordered problems such as the TSP, they do not take into consideration the relationship between nodes in the problem space. As such, existing population diversity metrics for general optimisation problems are not efficient for the ordered problems, including the TSP.

1.2 Our Contributions

This paper proposes new metrics for measuring the SPD and HPD of a population for problems with ordered lists by considering the longest common subsequence (LCS) between the fittest individual and the population (Sect. 2). Furthermore, the application of these new SPD and HPD metrics are applied to an adaptive GA framework that applies crossover and mutation rates in an on-line manner and dynamically manages selective pressure on exploration and exploitation sub-populations (Sect. 3). An experimental evaluation is also presented to demonstrate the effectiveness of the proposed approach (Sect. 4).

2 Proposed LCS Based Diversity Metrics

In this section, we propose a metric for measuring population diversity with a sequence-wise approach. To motivate our approach consider different tours for Berlin52 (an instance of TSP problem) as illustrated in Fig. 1, where Fig. 1a is the optimal tour. A gene-wise approach to measuring population diversity will

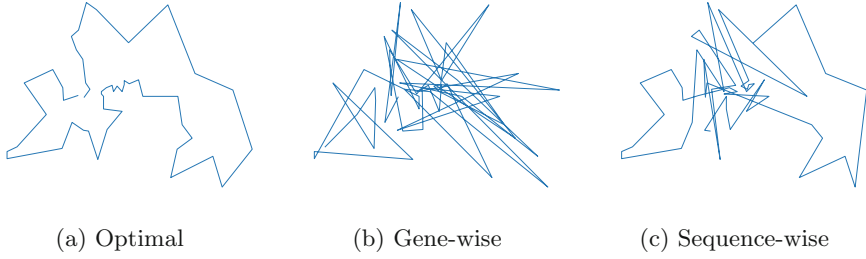


Fig. 1. Comparison of sequence-wise similarity and gene-wise similarity to the optimal tour for Berlin52

find that Fig. 1b has a 50% genetic similarity to the optimal solution and the tour represented in Fig. 1c has no similarity to the optimal solution. However, a sequence based diversity metric will find a 50% similarity between Fig. 1a and c. With the TSP, a tour may start from a different city but if the sequence in which the cities are visited are the same, the total distance travelled will also be the same. The solution in Fig. 1b has a travel distance of 24,986 while the solution in Fig. 1c is 13,786. This example demonstrates the importance of considering the sequence of genes in an ordered problem and the need for it to be taken into consideration when measuring the diversity of a GA's population.

To develop sequence-based diversity metrics, we can adopt either a contiguous sequence based approach or a non-contiguous sequence based approach. A contiguous sequence is where each node in the sub-sequence must be sharing a border with the next node. Individuals with a high level of gene-wise similarity will have a high level of non-contiguous, sequence-wise similarity. However, they may not necessarily share a high level of contiguous sequence-wise similarity. Therefore, this paper adopts non-contiguous sequence based approach for determining the similarity between two individuals and exploits *longest common subsequence* (LCS) for developing diversity measures. The LCS identifies the longest common non-contiguous subsequence between two individuals [12]. The dynamic programming formulation of LCS is given in Eq. 1, where X and Y are the two sequences that are being compared.

$$LCS(X_l, Y_k) = \begin{cases} 0 & \text{if } l = 0 \text{ or } k = 0 \\ LCS(X_{l-1}, Y_{k-1}) + 1 & \text{if } X_l = Y_k \\ \max(LCS(X_l, Y_{k-1}), LCS(X_{l-1}, Y_k)) & \text{if } X_l \neq Y_k \end{cases} \quad (1)$$

Standard Population Diversity (SPD). SPD describes the variation of individuals of a population regardless of its fitness [7]. It is generally used to control adaptive mutation and crossover parameters in order to maintain population diversity.

In order to calculate the SPD, the average LCS length for the generation is calculated as G^μ . This can be expressed in Eq. 2 as the average LCS length between the fittest individual, F , and the remaining individuals in population P . While population

$$G^\mu = \frac{1}{P} \sum_{i=1}^P LCS(F, i) \quad (2)$$

diversity could be defined as a simple average, different problems and populations may result in large variances in SPD. To prevent this, we normalise the standard deviation against the mean. Equation 3 calculates the standard deviation of the LCS lengths between individuals of the population.

$$\sigma(G^\mu) = \sqrt{\frac{1}{P} \sum_{i=1}^P (LCS(F, i) - G^\mu)^2} \quad (3)$$

The standard deviation $\sigma(G^\mu)$ is then used to calculate the coefficient of variance in Eq. 4. Finally, the coefficient of variance is then used as the SPD to measure the variation of individuals in a population.

$$SPD = C_v(G^\mu) = \frac{\sigma(G^\mu)}{G^\mu} \quad (4)$$

Healthy Population Diversity (HPD). HPD differs itself from SPD by considering the fitness of the individuals. A healthy population refers to a population with a high degree of fitness and diversity. The HPD is used as a control mechanism for applying selective pressure [7]. Equation 5 calculates the individual i 's contribution to the overall fitness of population P . This is expressed as its fitness in proportion to the total fitness of the population. By applying the weights to LCS length of each individual, the weighted average is calculated as G_w^μ . This enables the HDP to take into consideration the sequence-wise contribution of each individual's fitness to the overall diversity and health of the population.

Similar to the SPD measure, the HPD measure also needs to be normalised against the mean in order to prevent variances that may occur. Therefore, the standard deviation of the weighted LCS is calculated using the weighted average genotype, G_w^μ , which is given in Eq. 7.

$$\sigma(G_w^\mu) = \sqrt{\sum_{i=1}^P w_i (LCS(F, i) - G_w^\mu)^2} \quad (7)$$

Finally, the HPD is calculated as being the coefficient of variance as in Eq. 8 and is considered as a fitness-based measure of population diversity.

$$HPD = C_v(G_w^\mu) = \left(\frac{\sigma(G_w^\mu)}{G_w^\mu} \right) \quad (8)$$

3 LCS-Based Adaptive Genetic Algorithm

This section describes the details of *LCS based adaptive genetic algorithm* (LCSB-AGA) and its implementation of the SPD and the HPD. The LCSB-AGA uses a T_{size} tournament selection method [13] with a modified crossover operator

(MOX) [14] and partial shuffled mutation (PSM) [15] as the mutation operator. MOX and PSM were selected for their performance and ability to retain integrity constraints for ordered lists. The framework for LCSB-AGA is presented in Fig. 2 where components with adaptive mechanisms have been highlighted.

LCSB-AGA uses an adaptive crossover probability (P_c) to determine whether an offspring will be generated through crossover. P_c is determined by the SPD of the previous generation and is used to create an exploration and exploitation sub-populations. When an offspring is determined to be generated through crossover, two individuals from the previous generation are selected through tournament selection. The HPD is used to determine the T_{size} for the tournament in order to adaptively apply selective pressure according to the health and diversity of the population. Crossover and mutation are applied with the P_m rate being 0.01. This encourages the exploitation sub-population to refine existing solutions.

When an offspring is designated to be generated without crossover, it becomes part of the exploration sub-population. A single individual is selected from a tournament where the T_{size} is determined by the HPD. Adaptive mutation is applied where P_m is determined by the SPD. This enables the GA to explore new solutions where a highly converged population will result in a high P_m .

By creating the two sub-populations, LCSB-AGA is able to process each with different strategies, designed for either exploration or exploitation. This enables LCSB-AGA to dynamically balance exploration and exploitation, which is similar to the idea presented in ACROMUSE [7]. Furthermore, as the size of these sub-populations are determined by P_c , which is in turn determined by SPD, the sizes of these sub-populations adapt to a population's diversity. Thus the LCSB-AGA is able to adjust the bias between exploration and exploitation according to the degree to which it has converged. This enables LCSB-AGA to maintain a healthy population diversity.

Along with the adaptive mechanisms outlined in Fig. 2, LCSB-AGA implements elitism by selecting an elite individual from the population. This is done through a tournament selection where $T_{size} = P$. This ensures that the best solution for each generation is not lost and is carried to the new generation.

3.1 LCSB-AGA Adaptive Crossover

Probability of crossover (P_c) is determined by the degree of diversity of the population and determines whether crossover occurs for the offspring. This is expressed as in Eq. 9.

$$P_c = \left[\left(\frac{SPD}{SPD_{max}} \times (K_2 - K_1) \right) + K_1 \right] \quad (9)$$

SPD_{max} is set to 4 as the coefficient of variance does not exceed 4 in practice. Two constants are used to manage limits with the P_c to ensure that a minimum proportion exists for the exploitation and exploration sub-populations. K_1 defines the minimum proportion of individuals that are protected from exploitation. With K_1 given a value of 0.4, in the situation where there is total convergence on a solution, 40% of the population is dedicated to exploration. K_2

defines the proportion of a population that is to be reserved for exploitation in a population with a maximum diversity attainable. K_2 is set to 0.2 to allow for 20% of the population to continue exploring with the majority focusing on exploitation.

3.2 LCSB-AGA Adaptive Selection

Adaptive selection is implemented through applying selective pressure according to the diversity of the population. This is done through adjusting the tournament size, T_{size} , for the tournament selection method as shown in Eq. 10. T_{max} is given a value of $\frac{P}{6}$ as results demonstrate that a tournament size larger than $\frac{1}{6}$ of the population $T_{size} = T_{max} \frac{HPD}{HPD_{max}}$ (10) size results in an instant loss of population diversity. This value ensures that a balance between selection pressure and diversity is maintained. HPD_{max} is the maximum attainable HPD value. Experiment results have shown that, in practice, the HPD does not exceed a value of 3.

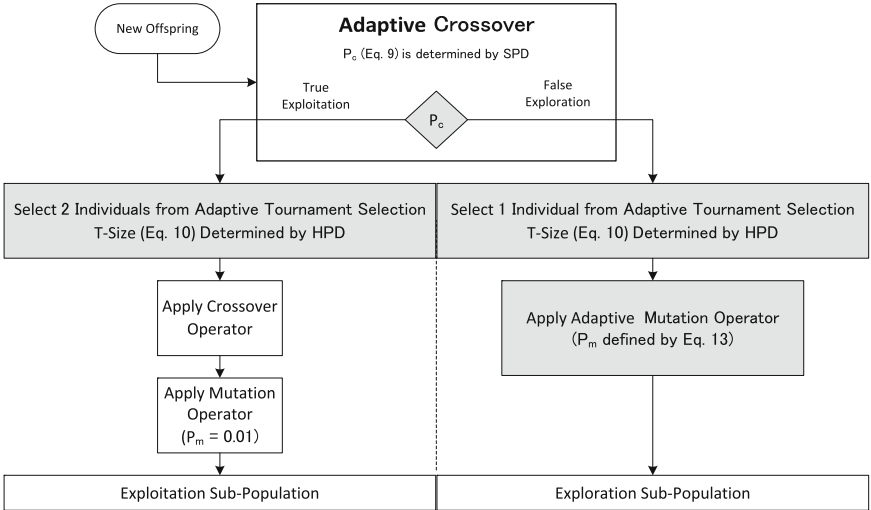


Fig. 2. Schematic diagram of our proposed LCSB-AGA framework

3.3 LCSB-AGA Adaptive Mutation

The probability for mutation (P_m) is the average between the diversity ratio of the population and the fitness ratio of the individual. This enables LCSB-AGA to adapt the mutation control parameters according to the population diversity and

the relative fitness of individuals.

Equation 11 shows the contribution of the SPD to the P_m where K is the upper bound of P_m and is set to 0.5 to ensure

$$P_m^{Diversity} = K \frac{SPD_{max} - SPD}{SPD_{max}} \quad (11)$$

only a maximum mutation probability of 50%. SPD_{max} is the maximum attainable SPD value as described in Eq. 9. Equation 12 describes the contribution of an individual's fitness as a ratio of the overall fitness. As an individual's fitness is yet to be calculated at this point, the fittest parent's

$$P_m^{Fitness} = K \left(\frac{f_{max} - f}{f_{max} - f_{min}} \right) \quad (12)$$

f value is used. f_{max} and f_{min} are respectively the best and worst fitness values from the previous population. K represents the upper limit and is given a value of 0.5. With both the $P_m^{Fitness}$ and $P_m^{Diversity}$ values, the average can be calculated as the P_m as expressed in Eq. 13. The K value for both Eqs. 11 and 12 ensures that a population that has the maximum diversity can not have a P_m value that exceeds the K limit. A P_m value of 0.5 will result in a very significant change in an individual's genotype.

4 Experiments

This section demonstrates the superiority of our LCSB-AGA in comparison to the best known static GA and ACROMUSE.

$$P_m = \frac{P_m^{Fitness} + P_m^{Diversity}}{2} \quad (13)$$

4.1 Setup

The experiments are conducted on an Intel Core i7-3770 CPU with 32 GB main memory. The algorithms are implemented in Python 2.7 and executed using the PyPy interpreter with Numpy. All three GAs use tournament selection [13], MOX [14] and PSM [15] as operators. The static GA has a static tournament size of 5 with ACROMUSE and LCSB-AGA implementing adaptive tournament sizes. The static GA has a p_c of 0.95 and p_m of 0.015. The p_c and p_m values for ACROMUSE and LCSB-AGA are dependent on their adaptive mechanisms. All GAs were run with a population size of 100. The convergence criteria is outlined in Eq. 14 where n is the problem size. When there are no improvements after c generations, the algorithm will complete. Instances from the Travelling Salesman Problem Library (TSPLIB) were selected according to the diversity in n size. Each test instance is run a total of 50 iterations for each GA. A two sample Z-Test is used to determine the significance of the difference in performance. The z value is calculated as shown in Eq. 15. The degree of freedom is 98 as $d_f = n_1 + n_2 - 2$. Both the d_f and t values are used in the z-distribution to determine the significance (p) in the difference between mean performance. $p \leq 0.05$ demonstrates a significant difference in performance where a $p \leq 0.01$ signifies a very significant difference. The Z-Test columns indicate the significance of the LCSB-AGA's performance improvements over the GA and ACROMUSE. A * indicates no significant improvement while a + indicates a significant improvement.

$$c = n + \sum_{k=1}^n k \quad (14)$$

$$z = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \quad (15)$$

4.2 Results

The experimental results are outlined in Table 1. These results demonstrate that there are no significant gains for either ACROMUSE or LCSB-AGA in problems where $n < 100$ with all GAs achieving average distances close to the optimal distance for each test. However, with problems where $100 \leq n < 200$, the LCSB-AGA manages to significantly outperform both the GA and ACROMUSE on a majority of test instances. While LCSB-AGA has a minor improvement in Eil101, Lin105 and Pr144 test instances, problems where $n \geq 200$, the results demonstrate that LCSB-AGA is able to achieve more robust performance gains as the problem size, n , increases.

Table 1. Results of the best GA, the ACROMUSE and the proposed LCSB-AGA

Test	Optima	GA			ACROMUSE			LCSB-AGA			Z-Test	
		Best.	Avg	S. Dev	Best.	Avg	S. Dev	Best.	Avg	S. Dev	GA	ACR.
wi29	27603	27603	27883	306.3	27603	27915	388.7	27603	27933	377.4	*	*
dj38	6656	6659	6716	116.7	6659	6765	163.6	6659	6691	118.3	*	+
eil51	426	438	451	7.9	433	454	9.2	434	453	8.8	*	*
berlin52	7542	7544	8089	261.6	7544	8036	223.6	7544	7834	254	+	+
st70	675	694	722	14.9	691	722	21.3	688	717	18.6	*	*
eil76	538	566	593	12.6	562	592	13.8	564	587	10.8	+	+
pr76	108159	109258	115261	3069.9	110195	114771	2953.8	109456	114018	3242.2	+	*
rat99	1211	1298	1349	25.4	1270	1351	33.5	1274	1336	33.6	+	+
rd100	7910	8282	8664	201.7	8191	8673	225.2	8127	8569	263.9	+	+
eil101	629	667	700	12.9	667	696	12.4	678	694	11.4	+	*
lin105	14379	14682	15499	460.3	14713	15635	416.5	14565	15162	410.1	+	+
pr107	44303	44762	47431	1302.8	44939	47211	1373.1	44685	46825	1047.3	+	*
pr124	59030	59736	62363	1789.3	60322	62592	1561.8	59409	61652	1796.3	+	+
ch130	6110	6394	6696	133.3	6376	6685	178.1	6345	6616	170.8	+	+
pr136	96772	101478	105843	2514.6	100574	106173	2394.4	101289	104868	2272.6	+	+
pr144	58537	59034	61627	1831.8	59115	62613	2413.2	58831	61459	2144.1	*	+
ch150	6528	7035	7294	156.7	6899	7275	184.6	6714	7203	172.9	+	+
pr152	73682	75618	78098	1489.5	75514	78308	1776.3	75278	77130	1377.4	+	+
rat195	2323	2510	2630	44.1	2541	2641	52.5	2496	2512	55.9	+	+
d198	16169	16169	16660	197	16169	16708	193.4	16169	16516	191.6	+	+
kroB200	29437	31720	32747	538.5	31157	32775	680.9	30689	32124	754.7	+	+
pr226	80369	81711	85995	2574.6	81788	85992	3372.8	81824	84144	3548.8	+	+
pr264	49135	53562	55756	1289.8	53177	55843	1400.1	52409	54923	1316.4	+	+
pr299	48191	51669	54201	1172.6	51120	54210	1120.5	5215	53366	1170.7	+	+
lin318	42029	45589	46904	727.8	44762	47017	965.1	44571	46204	861.4	+	+
pr439	107217	116090	120522	2618.7	114446	121546	2584.1	117840	118917	2876.7	+	+

Fitness Analysis. By examining four instances from Table 1, Fig. 3 illustrates some of the characteristics of LCSB-AGA, ACROMUSE and GA as they solve given problems. In Fig. 3a, all three algorithms can be seen as having a similar rate of improvement in fitness. However, ACROMUSE can be seen as converging on a good solution the fastest with LCSB-AGA and the traditional GA following afterwards. If looking at this data alone, the traditional GA could either be the least

efficient at exploiting known solutions or generating the most diverse populations. This can be clarified by examining the SPD and HPD later. However, with the small problem size, where $n = 100$, the final solutions found by the GAs are similar with no significant differences in performance. Figure 3b shows the performances of the algorithms in a slightly larger problem set where $n = 150$. Here, the LCSB-AGA can be seen as having a faster rate of convergence at the beginning and continues to explore longer than ACROMUSE. This suggests that the sequence-wise approach to maintaining diversity is capable of exploring and then exploiting solutions with greater efficacy than a gene-wise approach. The traditional GA can be seen as having a longer time than ACROMUSE to converge but also having lower efficacy than both ACROMUSE and LCSB-AGA. As the problem sizes get larger, the LCSB-AGA can be seen as having a greater and more significant advantage over ACROMUSE and the GA. In Fig. 3c LCSB-AGA maintains its population diversity longer than the other algorithms but also has a significant improvement in its average fitness. This is even more pronounced in Fig. 3d.

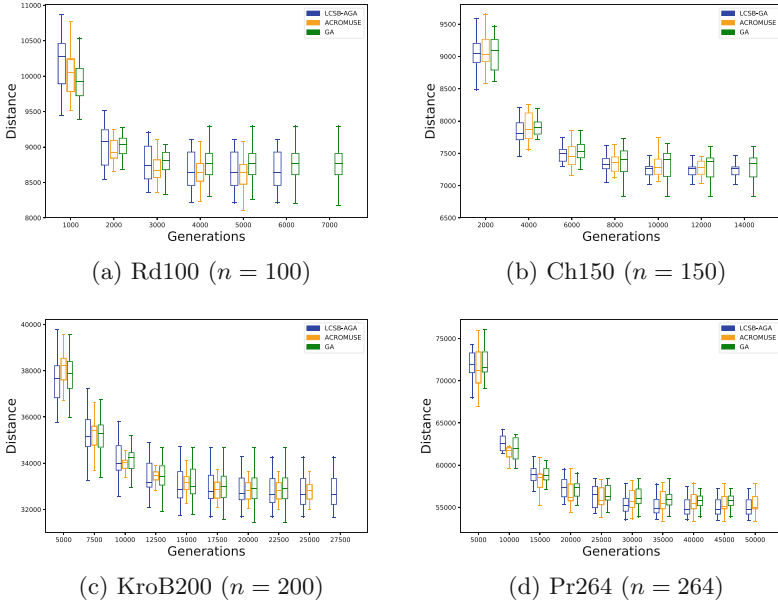


Fig. 3. Comparison of LCS, ACROMUSE and GA for best of fitness

Standard Population Diversity Analysis. SPD measures the population diversity without taking the population's overall fitness into consideration. This metric provides insight into the overall diversity of a population. Figure 4 illustrates the SPD trend for each GA when applied to the Rd100, Ch150, KroB200 and Pr264 TSPLIB problem instances. These instances were selected to demonstrate the characteristics of our sequence-wise population diversity metrics as the

problem size increases. While ACROMUSE and LCSB-AGA both share similar strategies for maintaining diversity, it can be seen that LCSB-AGA manages to maintain a higher level of sequence-wise diversity throughout the generations. The results in Table 1 show the sequence-wise approach of the LCSB-AGA has significantly improved performance over the gene-wise approach of ACROMUSE, we can see here that LCSB-AGA is also able to achieve and maintain a higher level of population diversity. In fact, Fig. 4 demonstrates that there is little difference between ACROMUSE and the GA with sequence-wise population diversity.

One common characteristic between all methods is how the SPD appears to fluctuate prior to converging. The traditional GA experiences larger fluctuations while the LCSB-AGA can be seen as attempting to alleviate the lack of diversity. ACROMUSE appears to also respond to the converging population by activating its own diversity mechanisms. In comparison with the GA, both the LCSB-AGA and ACROMUSE can be seen as fluctuating less severely and less frequently. This is likely due to the algorithms' attempts at maintaining population diversity. ACROMUSE experiences a sudden loss in SPD in Fig. 4d and both GA and LCSB-AGA in Figs. 4a and c. All five cases demonstrate how a sudden loss in SPD results in an early convergence. When the adaptive GA is able to maintain diversity as the population converges, it is able to continue improving its solution. This can be seen in Fig. 4d where LCSB-AGA begins to lose diversity at generation 30,000 but continues until converging around generation 50,000.

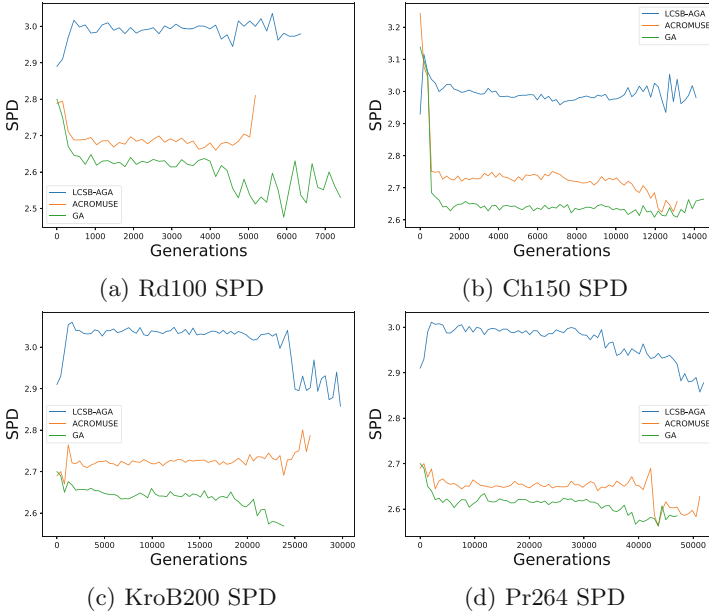


Fig. 4. SPD: Comparison between LCSB-AGA, ACROMUSE and GA

Another common characteristic from Fig. 4 is how a sudden and significant loss in diversity is soon followed by a premature convergence. This is a logical conclusion given the definition of convergence. In Fig. 4c, a number LCSB-AGA test runs where the test had run for a longer number of generations appear to have experienced lower diversity. LCSB-AGA can be seen as trying to recover its lost diversity by the fluctuations in its SPD. This can be attributed to LCSB-AGA attempting to re-balance its exploration and exploitation sub-populations. This loss in diversity lead to P_m values reaching 0.32 which would have resulted in very disruptive changes to many individuals.

Healthy Population Diversity Analysis. HPD is a fitness-weighted measure of the population diversity. A highly fit and diverse population is considered to be a healthy population. Figure 5 demonstrates the HPD trend for each GA when applied to the Rd100, Ch150, KroB200 and Pr264 TSPLIB problem instances. The graphs presents the average HPD value over 50 runs at regular intervals. A higher value represents a healthier population. Across the four instances, the HPD can be seen as starting at a low value and increasing in a manner similar to the inverse of the fitness data from Fig. 3 as should be expected. A number of interesting characteristics of the HPD are presented in the figures. As the HPD takes fitness and diversity into consideration, the difference between LCSB-AGA and the other two algorithms is more significant. In smaller problems, such as Rd100 in Fig. 5a, there is little difference between the GA and ACROMUSE

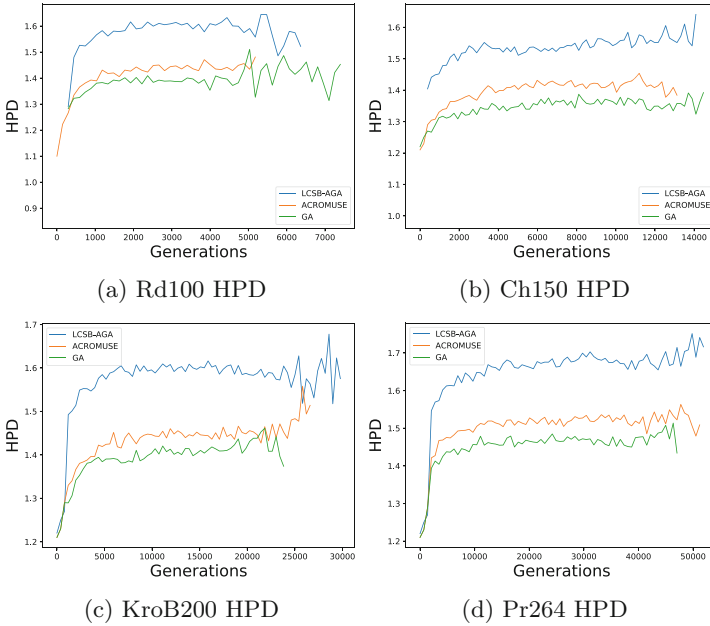


Fig. 5. HPD: comparison between LCS, ACROMUSE and GA

with both catching up to the HPD performance of LCSB-AGA towards the end. This is expected as all three methods performed strongly in the smaller problems with no significant difference in fitness between the three. However, the higher HPD of LCSB-AGA indicates that the greater population diversity contributed greatly. It can be seen that as the problem size increases, so do the differences between LCSB-AGA, ACROMUSE and GA.

One interesting observation is the KroB200 instance when comparing Fig. 4c and Fig. 5c. As stated previously, some runs for the KroB200 instance experienced lower diversity which resulted in longer run times and LCSB-AGA can be seen as trying to recover its diversity by introducing disruptive levels of mutation, as demonstrated in the fluctuations in SPD. However, when viewed in context with the HPD data, it can be seen that LCSB-AGA was successful in maintaining a reasonable level of healthy population diversity as the general trend maintained an upward projection.

5 Conclusion and Future Work

The TSP is a well known ordered optimisation problem with much theoretical and practical implications for which many variations of GAs have been proposed. While existing adaptive GAs have performed very well for general optimisation problems, they have not demonstrated the same success for ordered problems. In this paper, we present LCSB-AGA as an adaptive GA for ordered optimisation problems such as the TSP. A novel sequence-wise approach is also proposed for measuring the population diversity. The performance of the LCSB-AGA in terms of solution quality is demonstrated in the extensive experiments presented in this paper. In particular, the Z-Tests highlight the significance in the improvements as the problem size gets larger. Furthermore, by inspecting the SPD and HPD data, the sequence-wise approach is found to be having a greater degree of diversity and being better equipped to maintain it. The future work of this paper includes further investigation of the proposed sequence-based diversity metrics for other ordered optimisation problems such as the vehicle routing problem.

References

1. Eiben, Á.E., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. *IEEE Trans. Evol. Comput.* **3**(2), 124–141 (1999)
2. Chaiyaratana, N., Piroonratana, T., Sangkawelert, N.: Effects of diversity control in single-objective and multi-objective genetic algorithms. *J. Heuristics* **13**(1), 1–34 (2007)
3. Pan, Q.K., Suganthan, P.N., Wang, L., Gao, L., Mallipeddi, R.: A differential evolution algorithm with self-adapting strategy and control parameters. *Comput. Oper. Res.* **38**(1), 394–408 (2011)
4. Zhu, K.Q.: A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time windows. In: *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, pp. 176–183. IEEE (2003)

5. Shimodaira, H.: A diversity-control-oriented genetic algorithm (DCGA): performance in function optimization. In: *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 44–51. IEEE (2001)
6. Ursem, R.K.: Diversity-guided evolutionary algorithms. In: Guervós, J.J.M., Adamidis, P., Beyer, H.-G., Schwefel, H.-P., Fernández-Villacañás, J.-L. (eds.) *PPSN 2002. LNCS*, vol. 2439, pp. 462–471. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45712-7_45
7. Mc Ginley, B., Maher, J., O’Riordan, C., Morgan, F.: Maintaining healthy population diversity using adaptive crossover, mutation, and selection. *IEEE Trans. Evol. Comput.* **15**(5), 692–714 (2011)
8. Vidal, T., Crainic, T.G., Gendreau, M., Prins, C.: A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Comput. Oper. Res.* **40**, 475–489 (2013)
9. Segura, C., Hernandez, A., Luna, F., Alba, E.: Improving diversity in evolutionary algorithms: new best solutions for frequency assignment. *IEEE Trans. Evol. Comput.* **21**(4), 539–553 (2017)
10. Cruz-Salinas, A.F., Perdomo, J.G.: Self-adaptation of genetic operators through genetic programming techniques. In: *GECCO*, pp. 913–920. ACM (2017)
11. Adra, S.F., Fleming, P.J.: Diversity management in evolutionary many-objective optimization. *IEEE Trans. Evol. Comput.* **15**, 183–195 (2011)
12. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: *Introduction to Algorithms*. MIT Press, Cambridge (2009)
13. Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. *Found. Genet. Algorithms* **1**, 69–93 (1991)
14. Abdoun, O., Abouchabaka, J.: A comparative study of adaptive crossover operators for genetic algorithms to resolve the traveling salesman problem. *Int. J. Comput. Appl.* **31**(11), 49–57 (2011)
15. Abdoun, O., Abouchabaka, J., Tajani, C.: Analyzing the performance of mutation operators to solve the travelling salesman problem. *Int. J. Emerg. Sci.* **2**, 61–77 (2012)