# Robustness and Evolvability of Recombination in Linear Genetic Programming

Ting Hu[1], Wolfgang Banzhaf[2], and Jason H. Moore[1]

[1] Computational Genetics Laboratory, Geisel School of Medicine, Dartmouth College, Lebanon, NH 03756, USA
{ting.hu,jason.h.moore}@dartmouth.edu
[2] Department of Computer Science, Memorial University, St. John's, NL, A1B 3X5, Canada
banzhaf@mun.ca

**Abstract.** The effect of neutrality on evolutionary search is known to be crucially dependent on the distribution of genotypes over phenotypes. Quantitatively characterizing robustness and evolvability in genotype and phenotype spaces greatly helps to understand the influence of neutrality on Genetic Programming. Most existing robustness and evolvability studies focus on mutations with a lack of investigation of recombinational operations. Here, we extend a previously proposed quantitative approach of measuring mutational robustness and evolvability in Linear GP. By considering a simple LGP system that has a compact representation and enumerable genotype and phenotype spaces, we quantitatively characterize the robustness and evolvability of recombination at the phenotypic level. In this simple yet representative LGP system, we show that recombinational properties are correlated with mutational properties. Utilizing a population evolution experiment, we demonstrate that recombination significantly accelerates the evolutionary search process and particularly promotes robust phenotypes that innovative phenotypic explorations.

**Keywords:** Robustness, Evolvability, Accessibility, Neutrality, Recombination, Genetic Programming.

## 1 Introduction

In natural systems, the term *evolvability* is usually put forward to describe the capacity of a population to produce heritable and beneficial phenotypic variations [16,25,31]. Although the mechanisms and origins of evolvability are still largely under debate, another pervasive property of natural systems, *robustness*, is often discussed in connection with evolvability and is assigned explanatory power for some of the high evolvability of living systems [18,21]. Despite the fact that most random mutations to genetic material are deleterious, random mutations are the fundamental fuel of long-term evolutionary innovation and adaptation. Robustness enables living systems to remain intact in the face of constant genetic perturbations through allowing genetic variants to expand in

K. Krawiec et al. (Eds.): EuroGP 2013, LNCS 7831, pp. 97–108, 2013.

neutral spaces. These neutral spaces are genotypic regions in which mutations do not change phenotype or fitness and are the consequence of a redundant genotype-to-phenotype mapping [32]. Such *neutrality* augments evolvability, by accumulating genetic variations that might be non-neutral under changes of the environmental context [7,8,17,22,35].

A redundant mapping from genotype to phenotype is also pervasive in Genetic Programming (GP), where multiple genotypes encode identical phenotypes. A genetic change to a genotype, either mutation or crossover, is considered as neutral if it does not alter the phenotype or fitness. Extensive investigations and discussions have been carried through on how to characterize and utilize such neutrality in GP [1,3,4,11,28]. It has been recognized that neutrality enables phenotypes to be robust to genetic perturbations [29,36] and, more importantly, that it promotes the evolvability of phenotypes by expanding genotypes in neutral genotypic space without subjecting them to selection pressure [9,13].

In addition to extensive studies on mutational robustness and evolvability, it has been proposed recently in the context of gene regulatory circuits that recombination can create novel phenotypes more efficiently with a much less disruptive effect than mutation [20,33]. It is argued that recombination reorganizes genes and gene circuits and thus has greater phenotypic consequences than point mutation. Meanwhile it is less deleterious since it reuses existing genetic materials. In terms of expanding neutral spaces, recombination is also considered to be able to promote evolvability better than mutation. Neutral genetic variations by mutations are also called cryptic genetic variations that possess potential for creating novel phenotypes [21]. Such mutational robustness provides the quantitative staging ground for long-term adaptation and innovation. Recombination has a powerful effect augmenting those cryptic genetic variations to make qualitative changes [2,23,24].

Recombination has long been the center of the discussion on effective genetic operations in GP [5,19,27,30,34]. Similar to observations in gene regulatory circuits, it is well accepted in GP community that recombination is less destructive and has a larger phenotypic effect compared to point mutation. However, most robustness and evolvability studies in GP only consider mutations, and little has been done on quantifying recombinational robustness and evolvability and investigating the correlation between mutational and recombinational properties.

In a previous study [14], a quantitative characterization of mutational robustness and evolvability was performed in a simple Linear GP (LGP) system, where the entire genotype and phenotype spaces are finite and enumerable. In the current study, we adopt the same LGP system to utilize its compact properties and extend the quantitative metrics to recombination. In particular, we are interested to see whether different phenotypes have varying resilience or innovation potential under recombination. That is, if crossover is applied to a genotype from a given phenotype, is the individual more likely to stay in the same phenotype or to reach a novel phenotype? Are the probabilities to reach different novel phenotypes evenly distributed? Which phenotypes are more accessible from recombining genotypes of other phenotypes? Can recombination promote robust

phenotypes to generate high evolvability? We answer these questions by characterizing phenotypic recombinational properties in genotype and phenotype space under a special recombination operator as well as utilizing a population evolution scheme to look into the interplay between mutation and crossover in evolutionary dynamics.

## 2 Methods

### 2.1 Linear Genetic Programming on Boolean Search

We consider a simple Linear Genetic Programming system as in the previous study [14]. In the LGP representation, an individual (or computer program) consists of a set of $L$ instructions, which are structurally similar to those found in register machine languages. Each instruction has an operator, a set of operands, and a return value. To further restrict the search space, we use the LGP system on a Boolean search problem where each instruction consists of an operator drawn from the Boolean function set {`AND`, `OR`, `NAND`, `NOR`}, two Boolean operands, and one Boolean return value. The inputs, operands, and return values are stored in registers with varying read/write permissions. Specifically, $\texttt{R}_0$ and $\texttt{R}_1$ are calculation registers that can be read and written, whereas $\texttt{R}_2$ and $\texttt{R}_3$ are input registers that are read-only. Thus, a calculation register can serve in an instruction as an operand or a return, but an input register can only be used as an operand. An example program of length $L = 4$ is given here:

$$\begin{aligned}
\texttt{R}_1 &= \texttt{R}_2 \ \texttt{AND} \ \texttt{R}_3 \\
\texttt{R}_0 &= \texttt{R}_2 \ \ \texttt{OR} \ \ \texttt{R}_1 \\
\texttt{R}_1 &= \texttt{R}_1 \ \texttt{NOR} \ \texttt{R}_2 \\
\texttt{R}_0 &= \texttt{R}_3 \ \texttt{NAND} \ \texttt{R}_0
\end{aligned}$$

Instructions are executed sequentially from top to bottom. Prior to program execution, the values of $\texttt{R}_0$ and $\texttt{R}_1$ are initialized to `FALSE`. Registers $\texttt{R}_2$ and $\texttt{R}_3$ read two Boolean input values. After program execution, the final value in $\texttt{R}_0$ is returned as output.

### 2.2 Genotype and Phenotype Space

We consider each LGP program as a *genotype* and the binary Boolean function $f : \mathbf{B}^2 \to \mathbf{B}$, where $\mathbf{B} = \{\texttt{TRUE}, \texttt{FALSE}\}$, represented by the program as its *phenotype*. As described in the previous section, we allow two calculation registers, $\texttt{R}_0$ and $\texttt{R}_1$, two input registers, $\texttt{R}_2$ and $\texttt{R}_3$, and four possible Boolean operators, `AND`, `OR`, `NAND`, `NOR`. For the four loci on each instruction, only the two calculation registers can serve as the return (first locus), but all four registers can serve as operands (second and fourth locus), and all four Boolean functions can serve as the operator (third locus), which means there are $2 \times 4 \times 4 \times 4 = 2^7$ possible instructions and thus $2^{28}$ possible programs of length $L = 4$. These $2^{28}$ programs

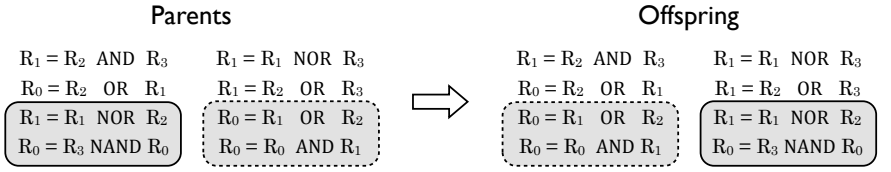


**Fig. 1.** Symmetric recombination. The crossover point is chosen at half length of a LGP program. Two parent programs (left) swap their third and forth instructions with each other to form two new offspring (right). A offspring is called phenotypically neutral with its parents if it does not map to a novel phenotype different from its parents.

define the finite genotype space mapping to the 16 possible binary Boolean functions $f : \mathbf{B}^2 \rightarrow \mathbf{B}$ as phenotypes. Such a highly redundant genotype-to-phenotype mapping suggests great robustness in the system.

We can expect that the distribution of genotypes among different phenotypes is highly heterogeneous. We use $s_i$ to denote the size of a phenotype $i$, i.e. the total number of genotypes that map to the same phenotype $i$. $s_i$ ranges from a minimum of 24,832 genotypes (for phenotype `EQUAL` and `NOTEQUAL`) to a maximum of 60,393,728 genotypes (for `FALSE`), occupying between $\ll 1\%$ and 23% of the genotype space, respectively. As examined previously [14], for this particular Boolean LGP system, all phenotypes are connected to each other in the mutational genotypic space. That is, for any given phenotype, there exists a genotype that belongs to this phenotype and can transform to another genotype in any other phenotypes through a point mutation.

### 2.3 Symmetric Recombination

In the current study, we only consider a single-point recombination [10,12,26] and always choose half of program length as the crossover point (Fig. 1). This allows all offspring resulting from crossover to have the same length as their parents and thus to limit recombination dynamics to within the finite genotype space we have defined. We restrict the crossover point in the current study in order to reduce the computational load of monitoring all possible recombination events. Recombinations are allowed for genotypes within a phenotype and across different phenotypes. Two parent programs generate two offspring through a recombination event.

We investigate crossover events for genotypes from each of the $\binom{16}{2} + 16 = 136$ different unordered phenotype pairs, denoted $\langle i, j \rangle$, where $i, j \in \{1, 2, 3, ..., 16\}$. A phenotype pair $\langle i, j \rangle$ has a finite number $r_{i,j}$ of possible recombination events, i.e. $r_{i,i} = s_i \times (s_i - 1)/2$ if $i = j$ and $r_{i,j} = s_i \times s_j$ otherwise, where $s_i$ is the size of phenotype $i$ as defined previously. Although the genotype space is finite, enumerating all possible recombination events for all pairs of phenotypes would be computationally prohibitive. For instance, there are more than 308 million possible recombination events for choosing two genotypes from even the smallest phenotype `EQUAL`. Therefore, we sample $S = 1,000,000$ crossover events

(without replacement) for genotypes from each phenotype pair $\langle i, j \rangle$. Among all the offspring generated by recombining a parent from phenotype $i$ and a parent from phenotype $j$, we use $x_{(i,j),k}$ to denote the number of offspring that belong to phenotype $k$. Since we sample the same number $S$ of crossover events across all possible phenotype pairs, we normalize $x_{(i,j),k}$ by adjusting it for different phenotype pairs, i.e. $x'_{(i,j),k} = \frac{r_{i,j}}{S} \times x_{(i,j),k}$.

### 2.4 Metrics on Recombinational Properties of Phenotypes

Intuitively, a phenotype is more robust under recombination if its crossover offspring are less likely to be phenotypically different from their parents. We define *recombinational robustness* $R$ of phenotype $i$ as the average fraction of phenotypically neutral offspring over all offspring,

$$R_i = \frac{1}{16} \times \sum_{j=1}^{16} \frac{\sum_{k=i,j} x'_{(i,j),k}}{\sum_{k=1}^{16} x'_{(i,j),k}}. \tag{1}$$

Similar to mutational metrics [6,14,15], we capture recombinational evolvability as the potential to change from one phenotype to another (different) phenotype. Let

$$f_{(i,j),k} = \begin{cases} \frac{x'_{(i,j),k}}{\sum_{l \neq i, l \neq j} x'_{(i,j),l}}, & \text{if } k \neq i \text{ and } k \neq j \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

denote the fraction of offspring that result in genotypes of phenotype $k$ by recombining genotypes from phenotypes $i$ and $j$. We define *recombinational evolvability* $E$ of a phenotype $i$ as

$$E_i = 1 - \sum_{j,k} \left( \frac{f_{(i,j),k}}{\sum_{l,m} f_{(i,l),m}} \right)^2. \tag{3}$$

Since $\sum_{j,k} \frac{f_{(i,j),k}}{\sum_{l,m} f_{(i,l),m}} = 1$ for each $i$, Eq.(3) describes the diversity of the connections from phenotype $i$ to other phenotypes via recombination. In other words, $E_i$ captures the probability that randomly chosen genotypes from phenotype $i$ generate recombination offspring with distinct phenotypes. This evolvability measure takes on a higher value if a phenotype has a more evenly distributed potential to reach other phenotypes through recombinations.

In addition to measuring the propensity to leave a phenotype, we also use *recombinational accessibility* $A_k$ to describe how easily a phenotype $k$ can be reached via recombination events from other phenotypes, formally defined as,

$$A_k = \sum_{i,j} f_{(i,j),k}. \tag{4}$$

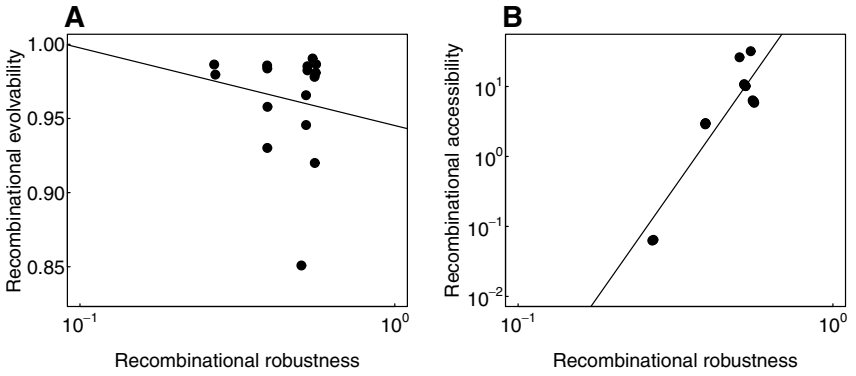


**Fig. 2. A**) recombinational evolvability and **B**) recombinational accessibility relative to recombinational robustness. Each data point represents a phenotype. Linear-log scale is chosen for **A**) and log-log scale is chosen for **B**) based on their best fitting relationship. The lines show the best fitting curves and provide a guide for the eye.

### 2.5 Population Evolution

In addition to sampling recombination events in the static genotype and phenotype spaces, we also perform population evolution experiments to investigate the interplay between mutation and recombination in a population under evolution. We choose the least represented phenotype `EQUAL` as the target phenotype to allow evolution to proceed over a longer time.

A non-overlapping generational evolution model with a fixed population size $|P|$ is adopted in this study. After population initialization, a new generation of offspring is produced sequentially. We randomly choose an individual with replacement, mutate according to a certain rate, and place it into the next generation. This is repeated $|P|$ times until the next generation of the population is filled. When both mutation and recombination are applied, for each generation, we randomly choose two individuals with replacement, cross them over at a given rate, mutate their crossover offspring at a given rate, and place both offspring into the next generation. This is repeated $\frac{|P|}{2}$ times until the next generation of the population is filled. The evolution process is terminated when the target phenotype is reached, and the required number of generations is recorded for each run.

## 3 Results

### 3.1 Recombinational Robustness, Evolvability, and Accessibility

Through the extensive sampling, this LGP problem instance is found having complete recombinational connections. That is, for any given phenotype pair, there exist pairs of their genotypes that can generate recombinational offspring of any other phenotypes. Fig. 2 shows the correlations among the recombinational

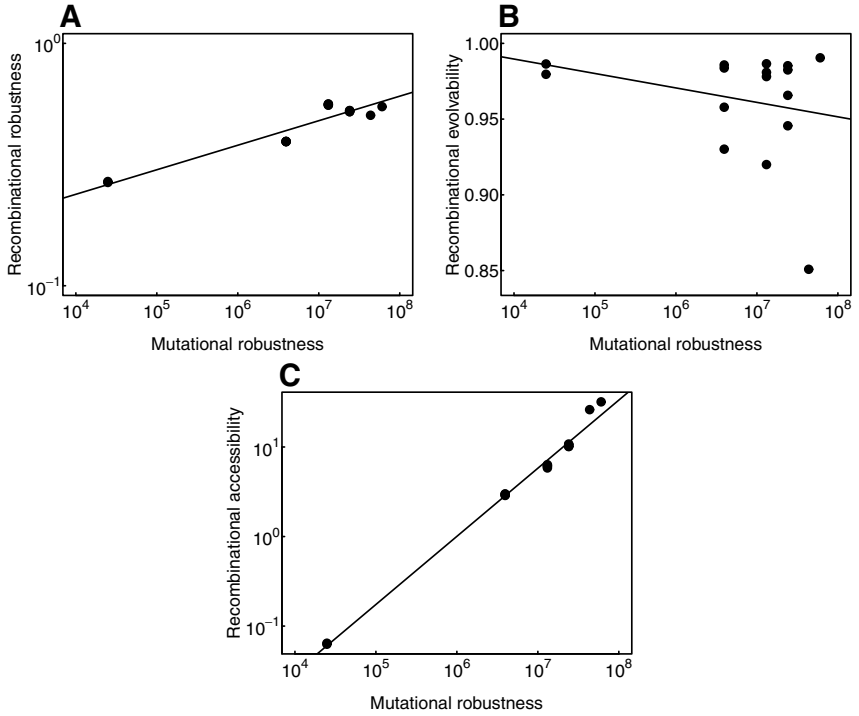


**Fig. 3.** **A**) recombinational robustness, **B**) recombinational evolvability, and **C**) recombinational accessibility relative to mutational robustness. Log-log, linear-log, and log-log scales are chosen accordingly based on the best-fitting relationships. Each data point represents a phenotype and the lines depict the best fitting curves.

metrics. Recombinational evolvability is weakly and negatively correlated with recombinational robustness with linear-log fitting $r^2 = 0.02395$, $p = 0.5671$ (Fig. 2-**A**). Phenotypes that have low recombinational robustness are highly evolvable, and robust phenotypes can have either high or low recombinational evolvability. In contrast, recombinational accessibility is strongly and positively correlated with recombinational robustness with a log-log fitting $r^2 = 0.8262$, $p = 1.09 \times 10^{-6}$ (Fig. 2-**B**). This suggests that phenotypes that are resilient to recombination are also very accessible from recombining genotypes of other phenotypes.

### 3.2 Comparisons of Recombinational and Mutational Measures

We now compare the recombinational measures to the previously investigated mutational measures [14,15]. Fig.3 shows recombinational robustness, evolvability, and accessibility relative to mutational robustness. Recall that phenotypic mutational robustness is defined as the size of a phenotype, i.e. its total number of underlying genotypes.

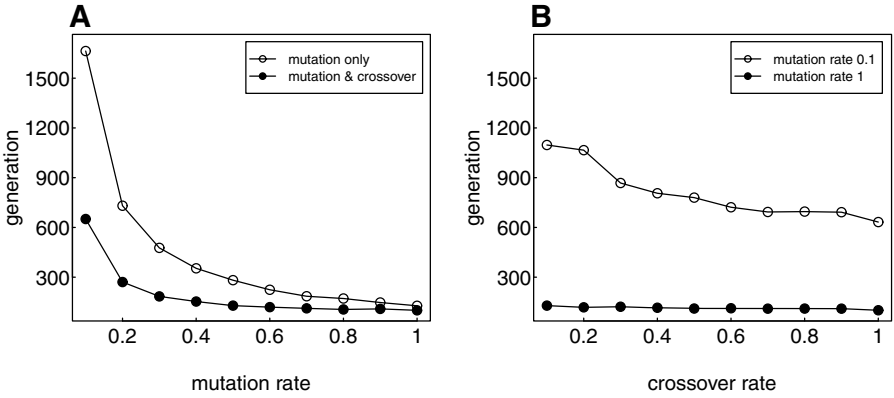


**Fig. 4.** The generations required to reach the target as a function of **A**) mutation rate and **B**) crossover rate. **A**) Mutation rate varies from 0.1 to 1 and crossover rate is fixed to 1. Two sets of experimental results are included, population evolution with mutation only (circles) and population evolution with both mutation and crossover (solid points). **B**) When both mutation and crossover are applied, we fix mutation rate to 0.1 and 1 and vary crossover rate from 0.1 to 1.

Recombinational robustness is positively correlated with mutational robustness (Fig. 3-**A**), which suggests that mutationally robust phenotypes are also resilient to recombinations ($r^2 = 0.8732$, $p = 1.172 \times 10^{-7}$). Similar to the weak relationship between recombinational evolvability and recombinational robustness, recombinational evolvability is weakly and negatively correlated with mutational robustness (Fig. 3-**B** with $r^2 = 0.06805$, $p = 0.3291$). As seen in the upper-left corner of the figure, less mutationally robust phenotypes that have fewer underlying genotypes are highly evolvable through recombination. Among mutationally robust phenotypes, some are also highly evolvable through recombination, but some only have very biased recombinational connections to other phenotypes. Interestingly, recombinational accessibility has a very strong positive correlation with mutational robustness (Fig. 3-**C**). In addition to the previously found strong positive relationship between mutational robustness and mutational accessibility [14], this very strong correlation ($r^2 = 0.9915$, $p = 7.021 \times 10^{-16}$) suggests that phenotypes with a large number of underlying genotypes are highly accessible from other phenotypes by both mutation and by recombination.

### 3.3 Population Dynamics Results

We compare two evolution scenarios with mutation only and with both mutation and crossover. Population size is set to 100 for both cases. Fig. 4 shows the population evolution results. Each data point is an averaged value of 100 runs for each configuration. We first vary mutation rate from 0.1 to 1 and fix crossover rate to 1. In general, increasing mutation rate accelerates the search process, and applying both mutation and crossover allows to reach the target faster than

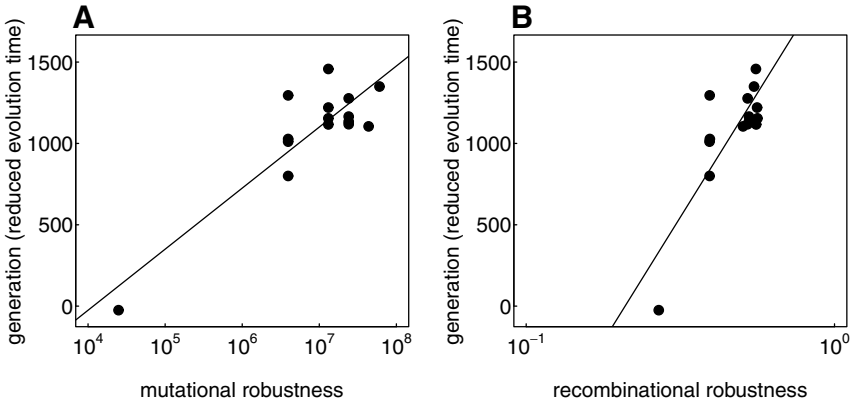


**Fig. 5.** The reduced evolution time, obtained by comparing mutation-only evolution and mutation-and-crossover evolution, relative to **A**) mutational robustness and **B**) recombinational robustness of the starting phenotypes. Data points represent different starting phenotypes, the line depicts the best linear-log fitting curve.

applying mutation alone. We then fix mutation rate and vary crossover rate (Fig. 4-**B**). Mutation rate is set to 0.1 and 1, and for each fixed mutation rate, crossover rate varies from 0.1 to 1. As seen in the figure, increasing crossover rate also accelerates the evolution process. The trend is more significant for mutation rate 0.1 than for mutation rate 1 in the figure, but it is clearly observable for both cases when one takes a closer look at appropriate scales.

Since it is shown that combining mutation and crossover significantly accelerates evolution, next we are interested to see if this improvement is correlated with the robustness of the starting phenotype. We choose a representative setting with mutation rate 0.1 and crossover rate 1, and obtain the reduced evolution time by taking the difference between mutation-only evolution time and mutation-and-crossover evolution time. As shown in Fig. 5, the reduced evolution time is positively correlated with the starting phenotypic mutational robustness ($r^2 = 0.7756$, $p = 1.513 \times 10^{-5}$) and recombinational robustness ($r^2 = 0.6772$, $p = 1.644 \times 10^{-4}$). This suggests that recombination improves the evolvability of phenotypes, and this improvement is more significant for more robust phenotypes. In other words, recombination promotes robust phenotypes to be more evolvable.

## 4 Discussion

This study examines the phenotypic robustness and evolvability subject to recombination. Utilizing a simple LGP system that has compact and finite genotype and phenotype spaces allows us to quantitatively characterize robustness

and evolvability at the phenotypic level. We also investigate the interplay between mutation and recombination in evolution dynamics by performing a generational population evolution experiment.

The phenotypes of our LGP system have varying recombinational robustness. Some of them are more tolerant to recombinations but some of them are not. Recombination-sensitive phenotypes are found highly evolvable by possessing a relatively evenly distributed potential to reach other phenotypes via recombination. Recombination-robust phenotypes are very accessible from recombining genotypes of other phenotypes. Recombinational robustness is positively correlated with mutational robustness, which suggests that over-represented phenotypes that have a great number of underlying genotypes are robust to both mutation and recombination. These over-represented phenotypes are also very accessible via both mutation and crossover. Through investigating population dynamics, recombination is found to be able to significantly accelerate evolutionary search if added to mutation. This acceleration is more significant when a population is initialized from a more robust phenotype.

Our results agree with findings from biological systems and also provide insights into our own computational systems. The ease of finding a target phenotype considerably depends on whether this target phenotype is over-represented by many genotypes. Less-represented phenotypes might be hard to reach, but they could serve as important bridges accessing other novel phenotypes. Robust phenotypes enhance the innovative power of recombination as they provide rich cryptic genetic variations for phenotypic exploration.

Future work will consider applying different recombinational operations such as a crossover point different from the mid-point adopted in this study or a non-symmetric crossover operation. We want to test if the current observations still hold in other scenarios and if our quantitative measures are sensitive to the choice of recombinational operation. It is also important to extend our quantitative measures to larger and more realistic systems. The results obtained here using a simple LGP system showcase the effectiveness of the quantitative approach and also generate hypotheses on how real and large-scale computational systems could behave. It would be very beneficial to test the scalability of our approach on more complex problem instances. An advantage of using larger-scale problem instances is that evolution will have a longer trajectory, and thus we could make observations on the detailed evolution dynamics at the individual level and see whether crossover leads to the prevalence of robust genotypes/phenotypes. We also would like to include fitness selection in our next step in particular with larger-scale problem instances. Finally, a varying selection pressure may have an impact on the evolution towards high robustness.

**Acknowledgments.** This work was supported by National Institute of Health (USA) grants R01-LM009012, R01-LM010098, and R01-AI59694 to J.H.M. W.B. acknowledges support from NSERC Discovery Grants, under RGPIN 283304-2012.

## References


1. Altenberg, L.: The evolution of evolvability in genetic programming. In: Advances in Genetic Programming, pp. 47–74. MIT Press, Cambridge (1994)
2. Azevedo, R.B., Lohaus, R., Srinivasan, S., Dang, K.K., Burch, C.L.: Sexual reproduction selects for robustness and negative epistasis in artificial gene networks. Nature 440(2), 87–90 (2006)
3. Banzhaf, W.: Genotype-phenotype mapping and neutral variation - a case study in genetic programming. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) PPSN 1994. LNCS, vol. 866, pp. 322–332. Springer, Heidelberg (1994)
4. Banzhaf, W., Leier, A.: Evolution on neutral networks in genetic programming. In: Yu, T., Riolo, R., Worzel, B. (eds.) Genetic Programming Theory and Practice III, ch. 14, pp. 207–221. Springer (2006)
5. Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D.: Genetic Programming: An Introduction. Morgan Kaufmann (1998)
6. Cowperthwaite, M.C., Economo, E.P., Harcombe, W.R., Miller, E.L., Meyers, L.A.: The ascent of the abundant: How mutational networks constrain evolution. PLoS Computational Biology 4(7), e1000110 (2008)
7. De Visser, J.A.G.M., Hermission, J., Wagner, G.P., Meyers, L.A., Bagheri-Chaichian, H., et al.: Evolution and detection of genetic robustness. Evolution 57(9), 1959–1972 (2003)
8. Draghi, J.A., Parsons, T.L., Wagner, G.P., Plotkin, J.B.: Mutational robustness can facilitate adaptation. Nature 463, 353–355 (2010)
9. Ebner, M., Shackleton, M., Shipman, R.: How neutral networks influence evolvability. Complexity 7(2), 19–33 (2002)
10. Francone, F.D., Conrads, M., Banzhaf, W., Nordin, P.: Homologous crossover in genetic programming. In: Banzhaf, W., Daida, J.M., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M.J., Smith, R.E. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1021–1026 (1999)
11. Galvan-Lopez, E., Poli, R.: An empirical investigation of how and why neutrality affects evolutionary search. In: Cattolico, M. (ed.) Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1149–1156 (2006)
12. Hansen, J.V.: Genetic programming experiments with standard and homologous crossover methods. Genetic Programming and Evolvable Machines 4, 53–66 (2003)
13. Hu, T., Banzhaf, W.: Neutrality and variability: two sides of evolvability in linear genetic programming. In: Rothlauf, F. (ed.) Proceedings of the Genetic and Evolutionary Computation Conference, pp. 963–970 (2009)
14. Hu, T., Payne, J.L., Banzhaf, W., Moore, J.H.: Robustness, Evolvability, and Accessibility in Linear Genetic Programming. In: Silva, S., Foster, J.A., Nicolau, M., Machado, P., Giacobini, M. (eds.) EuroGP 2011. LNCS, vol. 6621, pp. 13–24. Springer, Heidelberg (2011)
15. Hu, T., Payne, J.L., Banzhaf, W., Moore, J.H.: Evolutionary dynamics on multiple scales: a quantitative analysis of the interplay between genotype, phenotype, and fitness in linear genetic programming. Genetic Programming and Evolvable Machines 13, 305–337 (2012)
16. Kirschner, M., Gerhart, J.: Evolvability. Proceedings of the National Academy of Sciences 95, 8420–8427 (1998)
17. Landry, C.R., Lemos, B., Rifkin, S.A., Dickinson, W.J., Hartl, D.L.: Genetic properties influcing the evolvability of gene expression. Science 317, 118–121 (2007)

18. Lenski, R.E., Barrick, J.E., Ofria, C.: Balancing robustness and evolvability. PLoS Biology 4(12), e428 (2006)
19. Luke, S., Spector, L.: A comparison of crossover and mutation in genetic programming. In: Koza, J.R., Deb, K., Dorigo, M., Fogel, D.B., Garzon, M., Iba, H., Riolo, R.L. (eds.) Proceedings of the Annual Conference on Genetic Programming, pp. 240–248 (1997)
20. Martin, O.C., Wagner, A.: Effects of recombination on complex regulatory circuits. Genetics 183, 673–684 (2009)
21. Masel, J., Trotter, M.V.: Robustness and evolvability. Trends in Genetics 26, 406–414 (2010)
22. McBride, R.C., Ogbunugafor, C.B., Turner, P.E.: Robustness promotes evolvability of thermotolerance in an RNA virus. BMC Evolutionary Biology 8, 231 (2008)
23. Neher, R.A., Shraiman, B.I., Fisher, D.S.: Rate of adaptation in large sexual populations. Genetics 184, 467–481 (2010)
24. Otto, S.P.: The evolutionary enigma of sex. The American Naturalist 174(s1), s1–s14 (2009)
25. Pigliucci, M.: Is evolvability evolvable? Nature Review Genetics 9, 75–82 (2008)
26. Platel, M.D., Clergue, M., Collard, P.: Maximum Homologous Crossover for Linear Genetic Programming. In: Ryan, C., Soule, T., Keijzer, M., Tsang, E.P.K., Poli, R., Costa, E. (eds.) EuroGP 2003. LNCS, vol. 2610, pp. 194–203. Springer, Heidelberg (2003)
27. Poli, R., Langdon, W.B.: On the search properties of different crossover operators in genetic programming. In: Koza, J.R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D.B., Garzon, M.H., Goldberg, D.E., Iba, H., Riolo, R.L. (eds.) Proceedings of the Annual Conference on Genetic Programming, pp. 293–301 (1998)
28. Rothlauf, F., Goldberg, D.E.: Redundant representations in evolutionary computation. Evolutionary Computation 11(4), 381–415 (2003)
29. Soule, T.: Resilient individuals improve evolutionary search. Artificial Life 12, 17–34 (2006)
30. Soule, T., Heckendorn, R.B.: An analysis of the causes of code growth in genetic programming. Genetic Programming and Evolvable Machines 3, 283–309 (2002)
31. Wagner, A.: Robustness, evolvability, and neutrality. Federation of European Biochemical Societies Letters 579(8), 1772–1778 (2005)
32. Wagner, A.: Robustness and evolvability: A paradox resolved. Proceedings of The Royal Society B 275(1630), 91–100 (2008)
33. Wagner, A.: The low cost of recombination in creating novel phenotypes. BioEssays 33(8), 636–646 (2011)
34. White, D.R., Poulding, S.: A Rigorous Evaluation of Crossover and Mutation in Genetic Programming. In: Vanneschi, L., Gustafson, S., Moraglio, A., De Falco, I., Ebner, M. (eds.) EuroGP 2009. LNCS, vol. 5481, pp. 220–231. Springer, Heidelberg (2009)
35. Wilke, C.O.: Adaptive evolution on neutral networks. Bulletin of Mathematical Biology 63, 715–730 (2001)
36. Yu, T., Miller, J.F.: Through the interaction of neutral and adaptive mutations, evolutionary search finds a way. Artificial Life 12, 525–551 (2006)