Decision Space Coverage of Random Walks

Ryan Dieter Lang
Computer Science Division
Stellenbosch University
South Africa
Email: 19066171@sun.ac.za

Andries Petrus Engelbrecht
Department of Industrial Engineering, and
Computer Science Division
Stellenbosch University
South Africa
Email: engel@sun.ac.za

***Abstract*—Fitness landscape analysis is an approach used to mathematically characterize optimization problems. Random walk algorithms are used to sample fitness landscapes in order to perform fitness landscape analysis. Random walk algorithms have an advantage over random samples, in that random walk algorithms keep note of successive points in the walk, along with the relationships between them. It is important that the sample generated by a random walk algorithm is representative of the entire fitness landscape. A representative sample can be said to have good coverage of the decision space of the optimization problem. A new measure of the coverage of random walk algorithms, i.e. the Hausdorff distance, is proposed. The coverage of random walk algorithms found in the literature is investigated using the Hausdorff distance. This study shows that it is not sufficient to consider only the robustness of a random walk algorithm when performing fitness landscape analysis, but that the coverage of decision space should also be considered. This study shows that there is no significant difference in the coverage provided by the random walk algorithms investigated. However, the differences between the coverage of the random walk algorithms is more prominent when the length of the random walks is short, or the dimensionality of the optimization problem is increased.**

***Keywords*—random walk algorithms, fitness landscape analysis, decision space coverage, Hausdorff distance**

I. Introduction

Fitness landscape analysis (FLA) is an approach used to analyze the search landscape of an optimization problem. In order to do this, the value of the objective function, or fitness function, of the optimization problem can be calculated for each possible combination of values of the decision variables. In a continuous-valued decision space, there are infinitely many points, and it is therefore infeasible to calculate all of the fitness values for a continuous-valued optimization problem. Therefore, in order to perform analyses of the fitness landscape, a sample of points in the decision variable space is considered.

Random walk (RW) algorithms are used to sample fitness landscapes. RW algorithms are different to regular sampling algorithms, in that RW algorithms take into consideration the neighbourhood of the points in the sample. As described by Malan and Engelbrecht [9], in order to adapt RW algorithms for continuous-valued decision spaces, there are a few considerations that should be taken into account. Firstly, each consecutive point in the RW should be within the neighbourhood of the previous point in the RW. It is typical to use distance measures, such as Euclidean distance, to define these neighbourhoods for the points in the RW. Secondly, the RW should not make use of the fitness values to direct the random walk. Lastly, the RW should provide as large as possible coverage of the decision space given a certain computational budget. It is important to note that a RW algorithm should have significantly less computational cost than attempting to solve the optimization problem when using a trial-and-error process [9].

By nature, RW algorithms are stochastic. Therefore, it is likely that two different executions of a RW algorithm will cover different areas of the decision space. Even so, it is important that fitness landscape characteristics, which are calculated from these random walks, are reliable. Lang and Engelbrecht [8] performed a study of the reliability, or robustness, of FLA measures obtained by different RW algorithms. A RW is said to be robust if, over multiple independent runs of the RW algorithm, its resulting FLA measures do not fluctuate significantly. Note when calculating the robustness of the RW, the RW should have the same parameterization over the multiple independent runs. The results of the study in [8] indicated that the longer the RW, the more likely it is to provide robust measures. However, an interesting result from this study was that a number of short RWs also provided robust measures for certain fitness landscape measures. Since short RWs are likely to not cover large areas of the decision space, it implies that these short RWs have FLA measures that are biased to the small area of the decision space covered by the RW. It is, therefore, not sufficient to only consider the robustness of the FLA measures, but the coverage of the decision space should also be considered.

The objectives of this study are the following:

- To introduce a measure of the coverage of RWs in continuous-valued decision spaces, i.e. the Hausdorff distance.
- To illustrate that the short RWs that were found to provide robust FLA measures by Lang and Engelbrecht [8] do not provide sufficient coverage of the decision space.
- To determine which RW algorithm provides the best coverage of continuous-valued decision spaces. This study provides a survey of continuous RW algorithms and their coverage of decision spaces.

- To investigate the effect of the length of the RWs on the decision space coverage.

The remainder of this paper is structured as follows: Section II provides an overview of RW algorithms for continuous-valued decision spaces. Section III discusses the commonly used RW coverage measures, and introduces a new coverage measure. Section IV discusses the empirical procedure of the study. Section V presents and discusses the results. Section VI provides the conclusions of the study, and discusses potential future work.

II. Random Walks

This section describes the RW algorithms investigated in this study. The simple random walk was the first RW developed for continuous-valued decision spaces [9], however, it exhibits poor coverage of the decision space, as noted by Malan and Engelbrecht [9]. All the other RW algorithms discussed in this section were developed to improve the coverage of the decision space.

RW algorithms for continuous-valued decision spaces have two parameters, namely the number of steps and a bound on the step size. The upper bound on the length of a RW can therefore be calculated by multiplying the number of steps by the bound on the step size. Note that since the step size for a RW algorithm is stochastic, the exact length of a RW will be different over different iterations of the RW algorithm. For brevity sake, 'upper bound on the length of a RW' is abbreviated to 'length of a RW' for the remainder of this paper.

The simple random walk (SRW), introduced by Malan and Engelbrecht [9], is the most basic RW algorithm in continuous-valued decision spaces. Consider an optimization problem with m decision variables. The first point in a SRW is then a random point within the m-dimensional decision space defined by the domains of the decision variables. Each subsequent step in the walk is generated by drawing m random values in the range $[-stepSize, stepSize)$ from the uniform distribution, and then adding these values to the previous point in the RW. Note that $stepSize$ is the bound on the step size for the SRW. This is repeated until there are n steps in the walk. Note that the SRW algorithm continuously generates a vector of random values until the new point in the walk is within the bounds of the decision space, and that the neighbourhood of each point is defined by the bound on the step size for the RW. Detailed pseudocode for the SRW algorithm can be found in [9]. As noted by Malan and Engelbrecht [9], the SRW algorithm provides poor coverage of the decision space, and the poor coverage is more pronounced for small step sizes.

Malan and Engelbrecht [9] proposed the progressive random walk (PRW) algorithm in order to improve on the coverage provided by the SRW algorithm. The PRW divides an m-dimensional decision space into 2^m non-overlapping zones. The PRW begins by randomly selecting one of these zones to begin the walk, with the initial point in the walk on the edge of the decision space in this zone. As with the SRW, each subsequent step in the walk is generated by generating m random values. However, for the PRW the random values are drawn from a uniform distribution with the range $[0, stepSize)$. In order to improve the coverage of the PRW, the direction of the walk is biased towards the opposite side of the decision space to where the walk currently is. If the PRW reaches the boundary of the decision space, the bias is switched to the opposite direction [9]. Detailed pseudocode of the PRW algorithm can be found in [9].

Jana et al. [7] proposed that a chaotic pseudo-random number generator be used instead of the pseudo-random number (PRN) generator for the SRW. Jana et al. found that their chaotic RW algorithms provide better coverage of the decision space, when compared to the SRW and PRW. The chaotic maps are initialized with a random value from the default pseudo random number generator. A number of chaotic maps were introduced in [7], namely the Logistic map, the Tent map, the Chebyshev map, the Cubic map, and the iterative chaotic map with infinite collapse (ICMIC). Jana et al. chose to only investigate chaotic maps that generate numbers in the range $[-1, 1]$, since this value can then simply be multiplied with the bound on the step size in order to get the next step in the walk. Thus, the two chaotic maps investigated are the Chebyshev and ICMIC maps. The resulting walks are respectively referred to as Chebyshev chaotic random walk (CCRW), and the ICMIC chaotic random walk (ICRW). Detailed pseudocode of the CCRW and ICRW algorithms can be found in [7]. Note that these algorithms are modifications of the SRW algorithm, where the random numbers are drawn from chaotic maps rather than from a uniform distribution. Jana et al. found that the CCRW and ICRW algorithms provided better coverage of the decision space than the SRW and PRW algorithms. On average, the CCRW algorithm was found to provide the best coverage across all investigated problem dimensions, but the ICRW algorithm provided smaller standard deviations for the coverage.

Viktorin et al. [11] proposed a modified PRW (mPRW) that also makes use of a chaotic pseudo-random number generator. Viktorin et al. investigated a number of chaotic maps in their study, namely the Burgers, Dissipative, Lozi and Tinkerbell chaotic maps. Furthermore, a mechanism called direction switching, is proposed to change the direction bias of the PRW. The direction switching in the mPRW algorithm works as follows: a predetermined switching probability threshold is chosen. Then, for each dimension of the RW, there is a chance that the bias of that dimension changes. This is determined by generating a PRN from the uniform distribution, and if this value is less than the threshold, the bias changes. The direction switching also keeps track of the previous dimension that had a bias change, so that the bias of the same dimension does not change twice (which would result in the RW going in the same direction, and therefore coverage is not improved). Only the coverage of the Lozi chaotic map mPRW (LPRW) and the Dissipative chaotic map mPRW (DPRW) algorithms were investigated with both of the above mentioned modifications. The pseudocode of the mPRW algorithm modifications can be found in [11]. Viktorin et al. found that both the LPRW and the DPRW algorithms provided better coverage of the decision space than the PRW.

The coverage provided by the RW algorithms introduced by Jana et al. and Viktorin et al. have yet to be compared with one another.

III. Coverage Measures

This section discusses existing approaches used to quantify the coverage of RW algorithms in continuous-valued decision spaces. Thereafter, a new coverage measure is proposed.

Previous research measured coverage of RW algorithms by binning the points of the RW into a predetermined number of bins, B, of equal size [7], [9], [11]. This process is performed for $m \geq 1$ dimensions in the decision space. Suppose a RW has P points in the walk. The points from the RW are then placed into the appropriate bins, depending on each point's position in the decision space. The number, or frequency, of points in each bin is then calculated. Thereafter, the average frequency of points per bin, across all decision variables, can be calculated. In the literature, this is termed the mean frequency of points in a bin. This process can be seen as a generalization of using histograms to determine the distribution of the points in higher dimensional spaces. A RW algorithm is then said to exhibit good coverage if its mean frequency of points in a bin is similar to $\frac{P}{B}$, which is the ideal mean frequency of points in a bin.

However, there are a number of issues that arise from estimating coverage by using a binning approach. If the number of bins is too small, then the granularity of the point distribution will be lost, and therefore the measure will not give a true representation of the distribution of points in the decision space. If the number of bins is too large, then the standard deviation of the mean frequency of points will be large, because there are likely to be many bins that end up with very few points in them. Therefore, the choice of the number of bins can be seen as an additional parameter for the problem of analyzing the coverage of a RW.

This paper proposes the use of the Hausdorff distance as a metric to measure the coverage of a RW algorithm. It will be noted that the Hausdorff distance metric does not introduce additional control parameters. The Hausdorff distance, d_H [5], measures the distance between two finite subsets of a metric space. Two finite sets are close in the Hausdorff distance if every point of either set is close to some point of the other set. Note that the Hausdorff distance has been used in a number of applications, most notably in multiobjective optimization [1], [6], [10]. In the case of multiobjective optimization, it is used to compare the solution set obtained by a multiobjective evolutionary algorithm (MOEA) with the true Pareto front. The solution set from a MOEA, or the approximation set, is denoted by $A = \{\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_{|A|}\}$, and the true Pareto front, or the reference set, is denoted by $R = \{\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_{|R|}\}$.

The Hausdorff distance is composed of the generational distance, GD, and the inter-generational distance, IGD. The GD of a solution set is the average distance from each solution point to the nearest reference point in the decision space, and the IGD of a solution set is the average distance from each reference point to the nearest solution point in the decision space [6].

The GD between the reference and approximation sets is denoted as $GD(A, R)$, and is defined as follows:

$$GD(A,R) = \frac{1}{|A|}\left(\sum_{a \in A} \min_{r \in R} d(\boldsymbol{a}, \boldsymbol{r})^p\right)^{\frac{1}{p}} \tag{1}$$

where $d(\boldsymbol{a}, \boldsymbol{r})$ can be any distance measure, but it is typically the Euclidean distance. Note that a value of $p = 2$ was initially used in the calculation of GD; however, a value of $p = 1$ is typically used to simplify interpretability and computation [1].

The IGD between the reference and approximation sets is denoted as $IGD(A, R)$, and is defined as follows:

$$IGD(A,R) = GD(R,A) \tag{2}$$

The Hausdorff distance is then defined as follows:

$$d_H = \max\{GD(A,R), IGD(A,R)\} \tag{3}$$

When the Hausdorff distance is applied to quantify the coverage of a RW, the metric space is the decision space, D, of the optimization problem, and the two subsets are:

- $R \subset D$, which is the set of randomly sampled points of D, and
- $W \subset D$, which is the set of points in D which are visited by a RW algorithm.

In order for a RW to provide a representative sample of the decision space, the points in R should be drawn from a uniform distribution. Therefore, R can be seen as the reference set, and W can be seen as the approximation set.

When $d_H = 0$, the reference and the approximation sets are equal. However, note that when $d_H = 0$, it does not imply perfect coverage of the decision space, but rather that the RW has provided uniform coverage of the decision space. It is infeasible for a sample produced by a RW algorithm to have perfect coverage of a continuous-valued decision space. Since the reference set in this application of the Hausdorff distance is drawn from a uniform distribution, the points in the reference set will uniformly cover the entire decision space for a sufficiently large sample. Therefore, the smaller the value of d_H, the better a RW algorithm covers the decision space.

In initial attempts at finding a new technique to quantify decision space coverage of a RW, the authors investigated statistical techniques such as Kolmogrov-Smirnov and Kullback-Leibler divergence. However, these statistical techniques often require a number of assumptions such as normality of the distribution of the points, and that the points in a sample are identically and independently distributed (i.i.d.). The Hausdorff distance requires no assumptions beyond the need of two finite sets of points, and therefore has the benefit of no assumptions about the distribution of the points in the RW, nor does it require the assumption that the points in the RW to be i.i.d.

TABLE I
LIST OF THE RW ALGORITHMS INVESTIGATED IN THIS STUDY, ALONG WITH THEIR ABBREVIATIONS

Random Walk Algorithm	Abbreviation
Simple random walk	SRW
Progressive random walk	PRW
Chebyshev chaotic simple random walk	CCRW
ICMIC chaotic simple random walk	ICRW
Lozi chaotic progressive random walk	LPRW
Dissipative chaotic progress random walk	DPRW

IV. Empirical Procedure

This section discusses the empirical procedure followed in this study to evaluate the applicability of the Hausdorff distance metric as a RW coverage measure. Section IV.A discusses the RW algorithms evaluated in this study. Section IV.B discusses the experimental procedure followed in this study.

A. Random Walk Algorithms

Table I lists the RW algorithms that are investigated in this study, along with their respective abbreviations.

For the LPRW and DPRW, the implementation used in this study makes use of both the chaotic pseudo-random number generator and the direction switching modifications. In this study, a switching probability of 5% is used, as in [11].

B. Experimental Procedure

This section describes the experimental procedure for each of the experiments in this paper. The experiments are as follows:

- To determine which RW algorithm provides the best coverage of the decision space, that is which RW algorithm has the smallest Hausdorff distance.
- To illustrate that the short RWs that were found to produce robust FLA measures in [8] do not have sufficient coverage of the search space.
- To investigate the effect of the length of the RWs on the decision space coverage.

In order to determine the applicability of the Hausdorff distance as a coverage measure, and to determine which RW algorithm provides the best coverage of the decision space, the empirical process of Malan and Engelbrecht [9] is followed: the decision space is the hypercube with bounds $[-100, 100]$ in all dimensions. The step size of the RW algorithms is set to 10% of the domain, which implies that the upper bound on the step size is 20. The number of steps in each of the RW algorithms is $10^4 \times m$, where m is the number of dimensions in the decision space. The RW algorithms are investigated in 2, 3, 4, 6 and 10 dimensions. This procedure is repeated for 30 independent runs. The coverage of the RWs in one dimension is not investigated, since the direction switching modification for the RW algorithms in [11] has no effect in a single dimension. It is important to note that this study investigates the coverage of a single RW in the decision space.

To find the RW algorithm that provides the best coverage from the above-mentioned procedure, a statistical analysis is performed as suggested by Derrac et al. [3]. Note that the smaller the Hausdorff distance of a RW, the better the RW covers the decision space of the optimization problem. Since the distribution of the sample of points generated by RW algorithms cannot be assumed to follow a normal distribution, non-parametric statistical tests are required. To compare performance of the RW algorithms, the Friedman, Friedman Aligned Ranks, and Quade statistical tests are applied. The Friedman test is a commonly used non-parametric statistical test, used to compare multiple paired groups. However, the Friedman test only allows for intra-dataset comparisons, and not for inter-dataset comparisons [4]. The Friedman Aligned test was created to address this issue. The Quade test assigns more weight to datasets which are deemed more difficult [4], which emphasises differences among the investigated algorithms. In the case of this experiment, the higher the dimensionality of the decision space, the more difficult it is considered. For a more in depth discussion of these tests, the reader is directed to [3]. The null hypothesis for these statistical tests is that there is no difference in the Hausdorff distance resulting from each of the investigated RW algorithms. A significance level of 0.05 is chosen a priori.

For the Hausdorff distance metric, the approximation set is the set of points in the decision space generated by the RW algorithm. The reference set is the set of points that are drawn from a uniform distribution. The complementary-multiply-with-carry (CMWC) pseudo-random number generator [2] is used to generate the reference set. In each experiment, the number of points in the reference set is equivalent to the number of steps in the RW. For example, with the above-mentioned experiment, there is $10^4 \times m$, points in both the reference and approximation sets, where m is the number of dimensions in the decision space. In order to simplify the calculations and the computations of the Hausdorff distance, the value of p in the GD and IGD calculations is set to 1.

In order to investigate the effect of increasing RW lengths on the coverage of the decision space, the Hausdorff distance is calculated for increasing number of steps in the RWs. The empirical procedure for this experiment is set up the same as the experiment used to determine which RW algorithm provides the best coverage of the decision space, which is described above. However, the number of steps in the RW algorithms are chosen as 10%, 20%, ..., 100% of $10^4 \times m$ steps, where m is the number of dimensions investigated.

In the study on the robustness of RW algorithms [8], Lang and Engelbrecht found that a number of relatively short RWs produced robust FLA measures. To illustrate the poor coverage of these RWs, the Hausdorff distance is calculated for the SRW in 10 dimensions, with a domain of $[-100, 100]$. As in [8], the number of steps in the RW is between 250 and 3050, and the bound on the step size is between 0.1 and 3.1. Each of these combinations of parameters of the SRW is used, so long as the upper bound on the length of the RW is below 2000 units. This experiment is repeated for 30 independent runs.

The results are plotted on a parallel coordinate plot, as in [8].

V. Results and Discussion

This section contains the results of the study, along with a discussion of the results. Section V.A contains the results for the experiment to determine which RW provides the best coverage. Section V.B contains the results relating to the coverage of short robust random walks. Section V.C contains the results relating to the experiment of the effect of increasing walk lengths and increasing dimensionality of the decision space.

A. Coverage of Random Walk Algorithms

Table II summarizes the results of the experiments conducted in order to determine the applicability of the Hausdorff distance as a coverage measure, and to determine which RW algorithm provides the best coverage. Since the experimental procedure is the same as in [7], [9], [11], the coverage provided by the Hausdorff distance metric can be compared to the coverage used by the binning method. Note that the results in Table II are averaged over 30 independent runs.

Jana et al. [7] investigated the coverage of the SRW, PRW, CCRW and ICRW algorithms. In order to compare the Hausdorff distance with the results in [7], the aforementioned RW algorithms are analyzed in isolation. From Table II, the PRW algorithm provides the best coverage in 2, 3 and 4 dimensions. However, in 6 and 10 dimensions, the CCRW provides the best coverage. Similarly, as the dimensionality of the decision space increases, the performance of the ICRW improves. The SRW provides a lower Hausdorff distance in 2 and 3 dimensions than both the CCRW and the ICRW. This is counter-intuitive, since the CCRW and ICRW are modifications of the SRW algorithm.

Viktorin et al. [11] investigated the coverage of the PRW, LPRW and DPRW algorithms. In order to compare the Hausdorff distance with the results in [11], the aforementioned RW algorithms are analyzed in isolation. From Table II, the LPRW provides the best coverage of the decision space in all dimensions except for the 2 dimensional decision space. Note that Viktorin et al. found that the DPRW provided the best coverage using the binning technique. However, the results in Table II indicate that the LPRW provides the best coverage using the Hausdorff distance. Furthermore, as the dimensionality of the decision space increases, DPRW actually provides worse coverage than the PRW.

When taking all of the RW algorithms into consideration, the RW algorithm that provided the lowest average Hausdorff distance, over 30 independent runs, in each dimension in Table II is boldfaced. Thus, the LPRW provides the best coverage in 2, 4 and 6 dimensions, the DPRW provides the best coverage in 3 dimensions, and the CCRW provides the best coverage in 10 dimensions. The RW algorithm that provides the best coverage is therefore problem dependent.

The average Hausdorff distance from all of the RW algorithms are quite similar for each of the dimensions investigated. Therefore, in order to determine if there is a statistically significant difference in the coverage obtain by the RW algorithms, a statistical analysis is performed as discussed in Section IV. Table III presents the rankings of the RW algorithms, along with the test statistic, and the p-value of the statistical test for the Friedman, Friedman Aligned and Quade tests.

Using a significance level of 0.05, and given the p-values for each of the statistical tests in Table III, the null hypothesis can not be rejected for any of the statistical tests. The null hypothesis states that there is no significant difference between the average Hausdorff distance for each of the RW algorithms investigated. Therefore, it cannot be concluded that any RW algorithm provides significantly better coverage of the decision space than any other RW investigated.

Despite rejecting the null hypothesis, the performances of the RW algorithms can still be ranked, as is indicated in Table III. The RW algorithm that has the highest ranking in each of the statistical tests is boldfaced in the table. From this table, it can be observed that, as expected, the SRW consistently provides the worst coverage, and the LPRW provides the best coverage according to each of the statistical tests.

The RW algorithms introduced in [7] and [11] both make use of the idea of replacing the PRNG with a PRN generated from a chaotic map. The RW algorithms in [7] modify the SRW, and the RW algorithms in [11] modify the PRW. Since the SRW has been shown to provide worse coverage than the PRW both here, and in [9], it is expected that the RWs introduced in [11] will provide better coverage than the RWs in [7]. However, this is not the case. From Table III, the CCRW is ranked higher than the DPRW by each of the statistical tests, while the ICRW ranks lower than the DPRW only in the Friedman test.

B. Coverage of short robust random walks

This section analyzes the coverage provided by the short RWs, that were found to produce robust FLA measures in in [8].

Figure 1 plots the average Hausdorff distance, over 30 independent runs. This experiment contains all of the combinations of parameter values of a SRW that result in a RW length of less than 2000 units. This experiment is conducted in 10 dimensions so that the coverage of the walks may be compared with both the results in Table II and the results found in [8]. Note that the boundary of the decision space is $[-100, 100]$ in all dimensions.

Figure 1 shows that increasing the length of the SRW results in lower Hausdorff distances. This is expected, because as the length of the RW increases, the more likely the RW is to explore more of the decision space. In this plot, the average Hausdorff distance ranges from 215.83 to 262.32. Therefore, the Hausdorff distances obtained for these short RWs are significantly higher than the Hausdorff distances in Table II.

In Figure 2, the path of a SRW that has 1000 steps and a step size of 1.0 can be visualized in its 2 dimensional decision space. Note that the x and y axes represent individual decision variables. This particular combination of parameters for a RW

TABLE II
AVERAGE HAUSDORFF DISTANCE FOR MAXIMUM NUMBER OF STEPS, OVER 30 RUNS. (STANDARD DEVIATION IN PARENTHESES.)
ENTRIES IN BOLD INDICATE THE RW ALGORITHM WHICH PROVIDES THE SMALLEST AVERAGE HAUSDORFF DISTANCE PER DIMENSION

		Average Hausdorff distance					
Dimension	Number of points in walk	SRW	PRW	CCRW	ICRW	LPRW	DPRW
2	20000	0.72983	0.71113	0.73739	0.74368	**0.71042**	0.7135
		(± 2.59e-05)	(± 7.92e-06)	(± 4.81e-05)	(± 9.59e-05)	(± 1.07e-05)	(± 1.91e-05)
3	30000	3.71466	3.62583	3.72901	3.74031	3.60962	**3.60860**
		(± 2.53e-04)	(± 1.59e-04)	(± 2.56e-04)	(± 5.17e-04)	(± 6.09e-05)	(± 1.05e-04)
4	40000	9.14666	8.89653	9.06195	9.10765	**8.82804**	8.93189
		(± 9.98e-04)	(± 5.16e-04)	(± 5.58e-04)	(± 9.47e-04)	(± 2.38e-04)	(± 5.75e-04)
6	60000	25.31897	24.70655	24.56895	24.67359	**24.42923**	24.83061
		(± 0.00234)	(± 0.0025)	(± 0.00243)	(± 0.00304)	(± 0.00144)	(± 0.00115)
10	100000	65.26578	64.50012	**63.16687**	63.28287	64.19945	66.19231
		(± 0.06565)	(± 0.05951)	(± 0.05289)	(± 0.0594)	(± 0.07265)	(± 0.07075)

TABLE III
RANKS ACHIEVED BY THE FRIEDMAN, FRIEDMAN ALIGNED, AND QUADE TESTS FOR THE RW ALGORITHMS IN TABLE II.
ENTRIES IN BOLD INDICATE THE HIGHEST RANKING RW ALGORITHM

Algorithm	Friedman	Friedman Aligned	Quade
SRW	5	24.2	5.267
PRW	3	13.8	3.34
CCRW	3.4	13.6	2.67
ICRW	4.4	15.8	3.67
LPRW	**1.6**	**7.6**	**1.8**
DPRW	3.6	18	4.267
Statistic	9.9143	8.5638	2.0937
p-value	0.0777	0.1278	0.1086

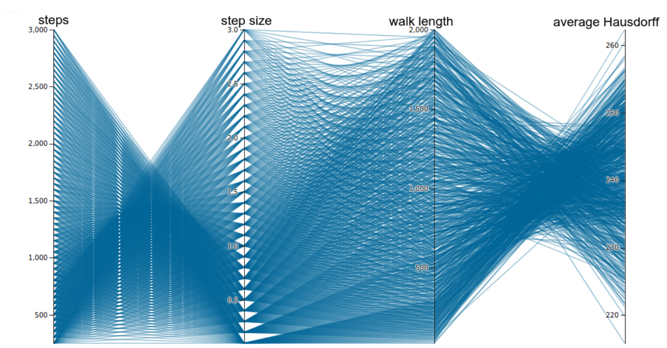

Fig. 1. Parallel coordinates plot of average Hausdorff distance for SRW in 10 dimensions, over 30 runs

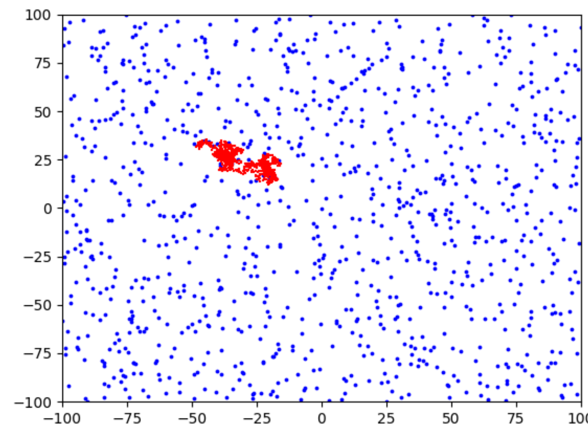

Fig. 2. Example of SRW with 1000 steps and step size 1.0 in 2 dimensions, resulting in a Hausdorff distance of 70.383. The points in the reference set are blue, and the points in the SRW are red.

is observed in [8] to produce FLA measures that are robust in 10, 20 and 50 dimensions. The SRW in Figure 2 resulted in a Hausdorff distance of 70.383, which is significantly lower than any of the Hausdorff distances found in Figure 1. Despite this lower Hausdorff distance, Figure 2 illustrates that this SRW provides extremely poor coverage of the decision space. This implies that when a RW algorithm provides a Hausdorff distance of 70, the coverage of the decision space is insufficient. Since the Hausdorff distances found in Figure 1 are much larger than 70, it is clear that the coverage of the decision space provided by these short RWs is poor, despite the fact that the resulting FLA measures for these short walks have been shown to be robust. Thus, these short RWs that provided robust FLA measures are not representative of the entire fitness landscape.

C. Effect of dimensionality and increasing random walk lengths

This section discusses the results of the experiments performed to investigate the effects of increasing RW lengths and increasing dimensionality of the decision space on the Hausdorff distance metric.

The Hausdorff distance for each of the RWs listed in Table I are plotted for various dimensions. In order to investigate increasing RW lengths, the bound on the step size is 10% of the decision space boundaries as in [9], while the number of steps in the walk is increased. The maximum number of steps for the RW is the same as in Table II. In each dimension, the boundaries of the decision space is $[-100, 100]$. Figures 3 to 7 contain the plots for the average Hausdorff distance in 2, 3, 4, 6 and 10 dimensions respectively.

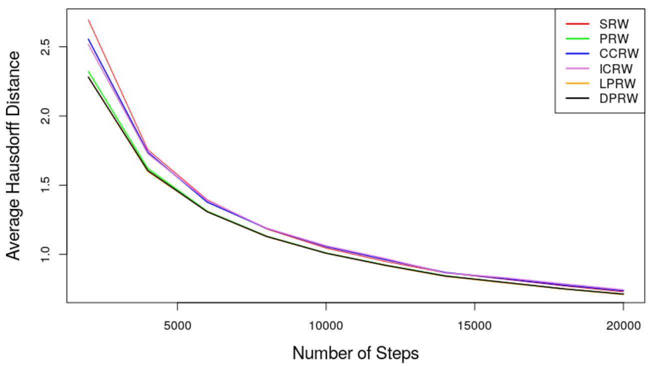

Fig. 3. Average Hausdorff distance for increasing RW lengths in 2 dimensions, based on 30 runs

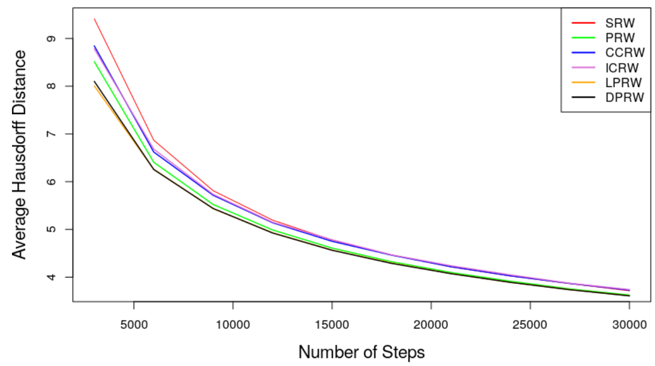

Fig. 4. Average Hausdorff distance for increasing RW lengths in 3 dimensions, based on 30 runs

The average Hausdorff distance observed for the different RW algorithms in Figures 3 to 7 are similar in each dimension. This can be seen by noting how close the plots of the average Hausdorff distance from the different RWs are in each dimension. This gives support to the finding in Section V.A that no RW algorithm provides significantly better coverage of the decision space that the other RW algorithms.

Figure 3 indicates that the average Hausdorff distance decreases exponentially for each of the RW algorithms as the length of the RW increases. The same pattern is observed in the remaining figures. This result is in line with intuition: as the length of the RW increases, the more likely the walk is to cover more of the decision space.

Figure 3 indicates that the differences between the average Hausdorff distance for the RW algorithms is much more prominent for short RW lengths. Likewise, this occurs in the remaining figures for the different dimensions. This result suggests that the choice of RW algorithm is more important for shorter RW lengths than for longer RW lengths.

The plots in Figures 3 and 4 are quite similar. However, it is important to note the increase in the scale of the y-axis (the average Hausdorff distance) from Figure 3 to Figure 4. Similarly, each increase in dimension shows significant increases in the average Hausdorff distance. This is likely due to the curse of dimensionality, indicating that the coverage of a RW becomes increasingly worse as the dimensionality of the problem increases.

The poor performance of the SRW becomes apparent in Figure 5, indicated by the gap between the SRW plot and the remaining RW algorithms' plots. Likewise, the good performance of the LPRW is observed in Figure 5, particularly with shorter RW lengths. The PRW begins to provide lower average Hausdorff distances in 4 dimensions than the DPRW.

In Figure 6, the SRW shows higher average Hausdorff distances than in Figure 5. Further, the ranking of the ICRW improves for 10 dimensions compared to previous dimensions.

Figure 7 shows more significant gaps between the plots of the various RW algorithms. This indicates that the choice

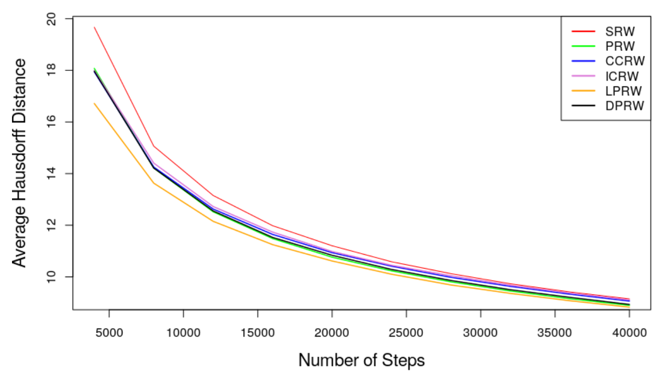

Fig. 5. Average Hausdorff distance for increasing RW lengths in 4 dimensions, based on 30 runs

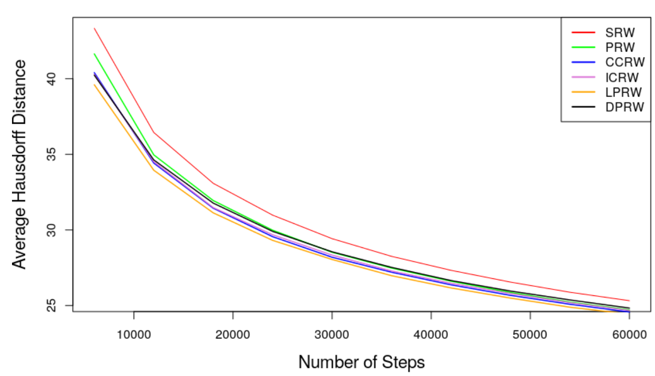

Fig. 6. Average Hausdorff distance for increasing RW lengths in 6 dimensions, based on 30 runs

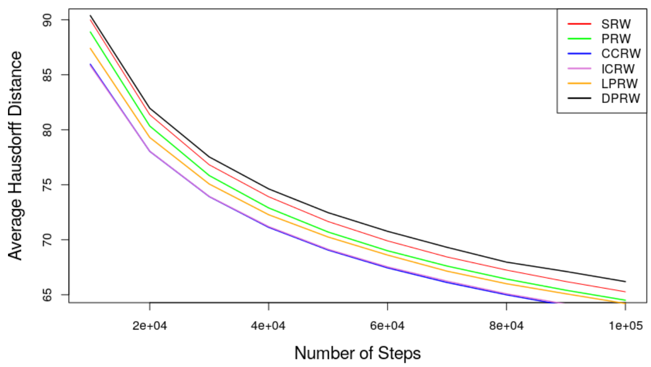

Fig. 7. Average Hausdorff distance for increasing RW lengths in 10 dimensions, based on 30 runs

of which RW algorithm to use becomes more important for higher dimensional decision spaces. The DPRW becomes the worst performing RW algorithm in 10 dimensions, even worse than the SRW. In 10 dimensions, the CCRW and the ICRW become the best performing RW algorithms, as discussed in Section V.A.

VI. Conclusions and Future Work

Random walk (RW) algorithms are used to sample points of a fitness landscape in order to perform fitness landscape analysis (FLA) on an optimization problem. A number of RW algorithms have been introduced for optimization problems with continuous-valued decision spaces. This paper introduced a new measure of decision space coverage of a RW algorithm, i.e. the Hausdorff distance. The coverage of a number of RW algorithms were investigated. The effects of increasing RW lengths and dimensionality of the decision space were also investigated.

The results of this study indicate that considering the robustness of a RW [8] in isolation is not sufficient, and that it is important to also investigate the coverage of the decision space that the RW provides. It is necessary that the sample that a RW provides is representative of the entire fitness landscape. If FLA is based on a sample generated by a RW with poor coverage, the results will be biased towards the properties of the portion of the fitness landscape that the RW sampled.

The results of this study indicate that there is no significant difference between the coverage provided by the RW algorithms investigated in this study. Despite this, the Lozi map based progressive RW (LPRW) was shown to provide the lowest Hausdorff distances, and thus provides the best coverage.

This study also indicates that the shorter the length of the RW, the more important the choice of RW algorithm becomes. This is because the differences between the coverage provided from different RW algorithms for short RWs is more significant than for long RWs. Increases to the dimensionality of the decision space also emphasizes the differences in coverage provided from different RWs. Therefore, the results suggest that there is a link between the coverage provided by a RW and the hypervolume of the decision space.

Possible future work can include the following:

- Investigate the coverage of alternative RW algorithms
- Investigate the coverage of RWs in more complicated decision spaces
- Investigate the effect of the size of the decision space on the coverage of the RW algorithms.
- Compare each of the RW samples against multiple random samples in order to reduce the bias of a single random sample on the coverage measures
- Analyze the coverage of RWs in higher dimensions
- Investigate the effect of step sizes and number of steps on the Hausdorff distance
- Investigate alternative reference point generators for the Hausdorff distance
- Up to now, all analysis of the coverage of a random walk has been on the decision space of a problem. The analysis can be expanded to coverage of the function space

Acknowledgment

The authors acknowledge the Centre for High Performance Computing (CHPC), South Africa, for providing computational resources to this research project.

References

[1] L. C. Bezerra, M. López-Ibáñez, and T. Stützle, "An empirical assessment of the properties of inverted generational distance on multi-and many-objective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2017, pp. 31–45.

[2] R. Couture and P. L'Ecuyer, "Distribution properties of multiply-with-carry random number generators," *Mathematics of Computation of the American Mathematical Society*, vol. 66, no. 218, pp. 591–607, 1997.

[3] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[4] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.

[5] J. Heinonen, *Lectures on analysis on metric spaces*. Springer Science & Business Media, 2012.

[6] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "Reference point specification in inverted generational distance for triangular linear Pareto front," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 6, pp. 961–975, 2018.

[7] N. D. Jana, J. Sil, and S. Das, "Continuous fitness landscape analysis using a chaos-based random walk algorithm," *Soft Computing*, vol. 22, no. 3, pp. 921–948, Feb 2018.

[8] R. Lang and A. Engelbrecht, "On the Robustness of Random walks for Fitness Landscape Analysis," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019, pp. 1898–1906.

[9] K. M. Malan and A. P. Engelbrecht, "A progressive random walk algorithm for sampling continuous fitness landscapes," in *Proceedings of the IEEE Congress on Evolutionary Computation*, July 2014, pp. 2507–2514.

[10] O. Schutze, X. Esquivel, A. Lara, and C. A. C. Coello, "Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 4, pp. 504–522, 2012.

[11] A. Viktorin, R. Senkerik, M. Pluhacek, and T. Kadavy, "Modified progressive random walk with chaotic PRNG," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 33, no. 5, pp. 450–459, 2018.