Position papers of the Federated Conference on Computer Science and Information Systems, pp. 81–86
DOI: 10.15439/2017F470
ISSN 2300-5963 ACSIS, Vol. 12

On Pathological Fitness Landscapes for Constrained Combinatorial Optimization

Gary Greenfield
Mathematics and Computer Science
University of Richmond
Richmond, Virginia 23173
Email: ggreenfi@richmond.edu

Aldeida Aleti
Faculty of Information Technology
Monash University
Melbourne, Victoria, Australia
Email: aldeida.aleti@monash.edu

***Abstract*—Population-based search methods such as evolutionary algorithms follow gradients in the fitness landscape under the assumption that high quality solutions will lead to even better ones. Most real-world optimisation problems, however, have constraints which lead to infeasible solutions that may disrupt these gradients. As a result, high quality solutions may lie in regions that are often unreachable from regions in the fitness landscape where the preponderance of feasible solutions lie. In such cases, the make-up of the initial population as well as critical aspects of the search strategy become the crucial factors in determining whether or not high quality regions are ever reached. In this paper, we present examples of pathological landscapes that arise by considering the constrained component deployment optimisation problem for which standard evolutionary algorithms are almost certain to fail to reach the regions where high quality solutions lie. We indicate how some simple modifications can help alleviate this problem.**

I. Introduction

The typical oral presentation of an evolutionary algorithm paper might include a fitness landscape slide such as the one shown in Figure 1. This tends to lull the listener into thinking that standard exploitation and exploration computation techniques will successfully explore the landscape encountering some number of local minima and maxima and, hopefully, eventually a global minimum or maximum.

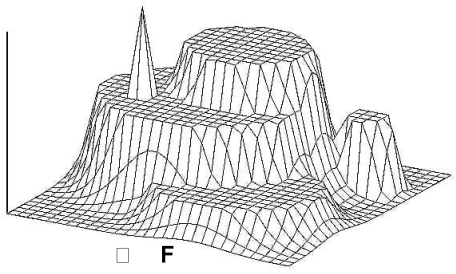

Fig. 1. A typical presentation slide for visualizing what the underlying fitness landscape for a combinatorial optimisation problem might look like.

Of course, this will not be true in general, but this trust becomes particulary misleading for problems where there are regions in the search space of the fitness function where fitness is undefined. The objective of this paper is to present examples inspired by a constrained combinatorial optimisation problem that highlight some of the pathologies that can arise in such situations and to suggest some simple "fixes" which might avoid these difficulties.

II. Background

A. Evolutionary Algorithms

In combinatorial optimisation *evolutionary algorithms* are iterative methods that evolve a population of solutions determined by *genomes* through the use of mutation, crossover, and selection operators. The optimisation process starts with a set of randomly generated genomes as an initial population. At every iteration, mutation and crossover operators are applied to some portion of the population.

The proportion of the population that is used for "reproduction" first undergoes crossover, which combines parts of two parent genomes to create two new genomes. A 1-point crossover splits both genomes at one point and combines the respective genes. It is possible to split solutions at more than one position, known as k-point crossover, and interleave the results. The points where the crossovers occur are selected at random. Next, the newly created solutions are mutated at a certain rate. There are various types of mutation operators: 1-point mutation mutates only one gene in the genome, uniform mutation mutates each gene with a certain probability, and transposition mutation operators swap two different genes in the same genome. Note that while 1-point and uniform mutation operators may alter existing genes, the transposition mutation operator can only change the position of genes. Hence, only 1-point and uniform mutation operators can introduce new genes not already in the population.

The selection operator decides which solutions will survive to the next iteration and adds new genomes as determined above in order to maintain the specified population size. Depending on the evolutionary approach, the selection can be based on elitism (only the best solutions survive), quality proportionate (the probability that a solution survives is based on its fitness), or random. We refer to mutation, crossover, and selection as search operators.

B. Fitness Landscapes

The suitability of an evolutionary method for solving an optimisation problem instance depends on the structure of the

fitness landscape of that instance. A fitness landscape in the context of constrained combinatorial optimisation problems is a setting comprising all of the following:

- a search space S of all possible genomes
- an ordered set V of fitness values
- a fitness function $F : S \to V$
- a set of constraints Ω
- a feasible set $S' \subset S$ satisfying all $\omega \in \Omega$,
- an infeasible set $I \subset S$ violating at least one $\omega \in \Omega$,
- a neighbourhood relation $N(s) \subset S$ for each $s \in S$

An example of a neighbourhood relation determined by 1-point mutation is the relation which assigns as neighbours to a genome all genomes that differ in one gene. A neighbourhood relation could also be specified by applying crossover with another genome, usually a gnome restricted to lie in some subset of S, followed by mutation.

C. The Software Deployment Problem

Aleti [1] considers a combinatorial optimisation problem that seeks to assign $n \geq 3$ software components $c_1, \ldots, c_n$ to $m \geq 3$ hardware devices $h_1, \ldots h_m$ subject to certain constraints in such a way that a fitness function that measures "reliability" is maximized once a *deployment* function $d : C \longrightarrow H$, where C is the set of components and H is the set of hardware units, is specified. Thus, subject to the constraints, once c_1 has been deployed to $d(c_1)$, c_2 to $d(c_2)$, and so forth fitness can be evaluated. But because of the constraints, not all candidate deployment functions $d : C \longrightarrow H$ are valid and thus the domain of feasible solutions for the fitness landscape has an unknown (and possibly unknowable) topology.

This context provided the inspiration for considering how difficult it might be to come up with a simple instance where "holes" in the domain might guarantee that absolute maxima would never be found using (standard) evolutionary search methods. In other words, we are looking for what would essentially be a minimal counterexample. Our attempts are described in the following sections.

III. A Minimal Counterexample

For a positive integer v, let Z_v denote the set $\{1, \ldots, v\}$. We modify the formulation of the component deployment optimisation problem slightly by writing the deployment function as $a : Z_n \longrightarrow Z_m$ so that c_1 gets assigned to $h_{a(1)}$, c_2 to $h_{a(2)}$, and so forth. In this way our fitness functions can be viewed as being defined on genomes that are vectors with n components *i.e.*, on n-tuples of the form $(a(1), \ldots, a(n))$, where $1 \leq a(i) \leq m$ for all i. This convention will facilitate counting in the sequel. Note that as n-tuples genotypes can be visualized as paths on the bounded region of the integer lattice given by $\{(x, y) \in Z \times Z | 1 \leq x \leq n, 1 \leq y \leq m\}$ by representing $(a(1), \ldots, a(n))$ as the path connecting the sequence of points $(1, a(1)), \ldots, (n, a(n))$.

Our constraints will be: c_1 cannot be deployed to h_m, or equivalently $a(1) < m$; c_n cannot be deployed to h_m, or equivalently $a(n) > 1$; and c_1 can be deployed to h_1 if and only if c_n is deployed to h_m, or equivalently $a(1) = 1$ if and only if $a(n) = m$. This last constraint is the critical constraint used to isolate a subset of assignment functions where maximal fitness solutions will lie. Our constraints are listed in Table I.

Constraints
$a(1) < m$
$a(n) > 1$
$a(1) = 1$ if and only if $a(n) = m$

TABLE I
CONSTRAINTS ON n-TUPLE GENOMES $(a(1), \ldots, a(n))$ WHERE $1 \leq a(i) \leq m$ FOR ALL i.

Our minimal counterexample takes $n = m = 3$, the first nontrivial case, so that of the twenty-seven possible 3-tuples that are potential candidates for a's only six satisfy the constraints, namely those of the form $(2, *, 2)$ or $(1, *, 3)$ where $*$ represents a "wild card" character that can assume any value chosen from the set $Z_3 = \{1, 2, 3\}$. Suppose the fitness function F satisfies $F((1, *, 3)) = 4$, $F((2, *, 2)) = 1$, and is undefined for any of the remaining twenty-one 3-tuples that do not satisfy the constraints. The six feasible solutions when represented as paths are shown in Figure 2.

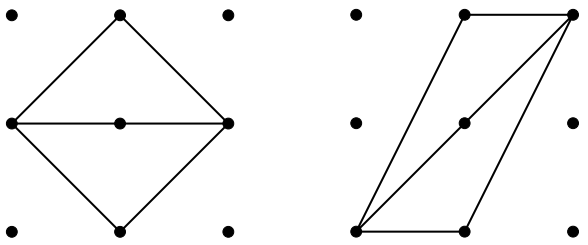

Fig. 2. The paths for our minimal counterexample.

Let our evolutionary method have population size $s = 3$, and suppose that none of the three 3-tuples that have maximal fitness make it into the initial population. That is, the initial population contains only genomes of the form $(2, *, 2)$. Then, regardless of whether one is using 1-point or 2-point crossover no genome produced by recombining two genomes in the current population will ever have maximal fitness. Further, if we use as a mutation operator single point mutation (*i.e.*, we change only one of the components) or we use a swap operator that interchanges two components, this is still the case. In fact, in a population of $(2, *, 2)$ genomes, crossover followed by either a one-point mutation or a swap will also fail to ever yield a maximal fitness genome of the form $(1, *, 3)$. In order for evolutionary computation to succeed for our toy problem when the initial population consists of only $(2, *, 2)$ genomes it must implement an operator that yields a swap combined with a one-point mutation or a mutation operator that perturbs two or more components *i.e.* a two-point mutation operator. It is also curious to note that if a global maximum is obtained by, say, a swap combined with a one-point mutation, then the genome must be $(1, 2, 3)$. Interestingly, the evolutionary algorithm used for the software deployment problem in Sabar

and Aleti [2] does implement a swap followed by a point mutation, however it checks the validity of the genome after the swap which would cause it to fail for our toy problem.

IV. Scaling The Counterexample

The reason our counterexample is so intriguing is because it generalizes and scales to a constrained combinatorial optimization problem with more plausible parameters. Using the same constraints, assume now that $m, n \geq 3$. Then the domain of candidate assignment functions consists of the m^n n-tuples with entries in $Z_m = \{1, \ldots, m\}$. We shall be fluid in referring to elements in this search space as n-tuples, candidates, solutions, genomes or points. We partition the set of candidates into three disjoint subsets: high quality feasible candidates Q, low quality feasible candidates B, and infeasible candidates I. Q consists of n-tuples of the form $(1, *, \ldots, *, m)$ of which there are m^{n-2}. B consists of n-tuples of the form $(X, *, \ldots, *, Y)$, where $1 < X, Y < m$ of which there are $(m-2)m^{n-2}(m-2)$. I consists of the remaining n-tuples. I also decomposes into disjoint sets. These disjoint sets together with their cardinalities are shown in Table II. We can check that this decomposition is correct by observing that we have accounted for all n-tuples as follows:

$$m^{n-2}[1 + (m-2)^2 + 4(m-2) + 3] = m^n.$$

Set	Cardinality
$(1, *, \ldots, *, Y)$	$(m-2)m^{(n-2)}$
$(X, *, \ldots, *, m)$	$(m-2)m^{(n-2)}$
$(m, *, \ldots, *, Y)$	$(m-2)m^{(n-2)}$
$(X, *, \ldots, *, 1)$	$(m-2)m^{(n-2)}$
$(m, *, \ldots, *, 1)$	$m^{(n-2)}$
$(1, *, \ldots, *, 1)$	$m^{(n-2)}$
$(m, *, \ldots, *, m)$	$m^{(n-2)}$

TABLE II
DECOMPOSITION OF SET OF I OF INFEASIBLE n-TUPLE GENOMES INTO DISJOINT SUBSETS. X AND Y ASSUME VALUES BETWEEN 2 AND $m-1$ WHILE $*$ IS A WILD CARD CHARACTER INDICATING ANY VALUE BETWEEN 1 AND m INCLUSIVE IS ALLOWED.

Our counting also tells us that when choosing an n-tuple at random, the probability of getting a feasible solution is $[(m-2)^2+1]/m^2$ and the probability of getting a candidate from B or I, a candidate that does *not* satisfy the condition $a(1) = 1$ if and only if $a(n) = m$, is $1 - (1/m^2)$. This makes it easy to determine the probability of randomly selecting genomes one at a time and winding up with an initial population of genomes lying exclusively in $B \cup I$ (see below).

It is more difficult to obtain a closed form expression for the probability that a an initial population where n-tuples are selected one by one, with infeasible solutions *discarded*, until a population (possibly with duplicates) of size s is obtained such that it contains only feasible solutions from B *i.e.*, only feasible solutions that don't have $a(1) = 1$ and $a(n) = m$. Let p_Q, p_B and p_I be the probabilities that a randomly chosen genome lies in Q, B and I respectively. We know

$$p_Q = 1/m^2,\ p_B = (1-2/m)^2,\ p_I = (4m-5)/m^2.$$

For fixed $j \geq s$, let p_j be the probability of getting a pool with s genomes from B after randomly choosing exactly j genomes. Then we know the last genome must have been from B and $s-1$ genomes from B must have shown up in the previous $j-1$ selections. Since there are $\binom{j-1}{s-1}$ ways for genomes from B to get selected, knowing the remaining $j-s$ choices all came from I, we have

$$p_k = [\binom{j-1}{s-1} p_B^{s-1} p_I^{j-s}] p_B,$$

whence the desired probability is:

$$\begin{aligned}
\sum_{j=s}^{\infty} p_k &= p_B^s \sum_{j=s}^{\infty} \binom{j-1}{s-1} p_I^{j-s} \\
&= p_B^s \sum_{j=s}^{\infty} \frac{s}{j} \binom{j}{s} p_I^{j-s} \\
&= p_B^s \sum_{k=0}^{\infty} \frac{s}{s+k} \binom{s+k}{s} p_I^k \\
&= (1-2/m)^{2s} \sum_{k=0}^{\infty} \frac{s}{s+k} \binom{s+k}{s} (\frac{4m-5}{m^2})^k.
\end{aligned}$$

If we are willing to accept infeasible solutions in the initial population, but require our fitness functions to assign positive values for feasible solutions and zero for infeasible solutions so that they will immediately be removed from the initial population, then we can assert that the probability of an initial population *not* having a feasible solution from Q, (*i.e.*, not having a genome of the form $(1, *, \ldots, *, m)$ is $(1-(1/m^2))^s$. Note that when $m = 10$ and $s = 100$ this probability already exceeds one-third. We assume an initial population of this type for the remainder of this paper. This assumption, coupled with more realistic parameter values, for example $s = 100$ and $m, n \geq 10$, allows us to formulate some additional pathological fitness landscape examples. We first digress to a discussion of search operators.

A. Search operators

For notation, we let X_i denote the 1-point crossover that occurs at position i where $1 < i < n$. Formally, for genomes a_1 and a_2, this means

$$\begin{aligned}
X_i((a_1(1), \ldots, a_1(n)), (a_2(1), \ldots, a_2(n))) = \\
((a_1(1), \ldots, a_1(i-1), a_2(i), \ldots, a_2(n)), \\
(a_2(1), \ldots, a_2(i-1), a_1(i), \ldots, a_1(n))).
\end{aligned}$$

For $1 \leq i \leq n$ and $1 \leq j \leq m$, we let $P_{i,j}$ denote the point mutation operator that assigns $a(i)$ to be j. For $1 \leq i < j \leq n$ we define $T_{i,j}$ to be the transposition operator that swaps $a(i)$ with $a(j)$.

B. Simple Scaling

Assume $m, n \geq 3$, the initial population contains solutions only from B and I, and the fitness function F is defined by setting $F((a(1), \ldots, a(n)))$ equal to 0, 1, and $m+1$ for

solutions from I, B, and Q respectively. Then, assuming that solutions from I are immediately removed from the population and only low quality solutions from B remain available for recombination and selection to form the next generation, remarks similar to the minimal counterexample apply. No single application of any $P_{i,j}$ or $T_{i,j}$ to a genome can produce a high quality solution from a low quality solution. This is still true even if they are applied to an intermediate genome arising from composition of a sequence of crossover operators.

However, such a "lifting" from B to Q will occur whenever $P_{1,1} \circ P_{m,m}$ is applied to a genome in B. If we posit a typical scenario where uniform point mutation is used, which is interpreted to mean that components are considered one by one so that, independently, each has probability p that a point mutation operator is applied, then for fixed i and j, the i-th component has probability p/m of having $P_{i,j}$ applied and the probability $P_{1,1}$ and $P_{n,m}$ are both applied is $(p/m)^2$. Note that for $p = 0.05$ and $m = 10$ this probability is 0.000025. Another possible way for such a lifting to occur is if a genome has $a(i) = m$ and $a(j) = 1$ where $1 < i, j < m$ — assume this occurs with probability ρ — and the composition of $T_{1,j}$ with $T_{i,m}$ (they are disjoint transpositions, so the order doesn't matter) is applied. Since there are $n(n-1)/2$ transposition operators, if there is some (small) probability ϵ of applying two swaps to a genome of the desired type, then the probability of a successful lifting is $4\rho\epsilon/(n^2 - n)$.

C. Biasing Low Quality Solutions

We can make it harder for liftings from B to Q to occur by arranging it so that the percentage of genomes with 1's and m's in interior components within a population consisting of genomes only from B decreases as the evolutionary algorithm progresses. That is, over time we can try to lower the value of ρ. Define the *target* t to be $\lfloor \frac{m+1}{2} \rfloor$. Note that $t = 2$ when $m = 3$, and that $1 < t < m$, for $m \geq 3$.

Assume the initial population consists of only candidates from B and I. As before, let genomes in I and Q have fitness 0 and $m + 1$, but now for $(a(1), \ldots, a(n)) \in B$ set

$$F((a(1), \ldots, a(n))) = 1 + \prod_{i=1}^{n} \frac{1}{|a(i) - t| + 2}.$$

Observe that these fitness values all lie strictly between 1 and 2. This biases the evolutionary algorithm such that as evolution proceeds an initial population consisting of genomes only from B and I converges to one consisting entirely of the unique local minimum solution $(t, \ldots, t)$ which has fitness $1 + 1/2^n$.

D. Biasing High Quality Solutions

Finally, we can immediately expel all but a select few high quality genomes that do creep into the population as a result of liftings from B to Q by lowering their fitness as follows.

Assume the initial population consists of only candidates from B and I. Let genomes in I and B have fitness 0 and 2 respectively. If $(1, a(2), \ldots, a(n-1), m) \in Q$, let u be minimal such that $a(1) = \ldots = a(u) = 1$ and $a(u+1) \neq 1$. Note that u is well defined and $1 \leq u < m$. Define $F(1, a(2), \ldots, a(n-1), m) = 1/(1+u)$ if $(1, a(2), \ldots, a(n-1), m) \neq (1, m, \ldots, m)$ and $m + 1$ otherwise. This fitness function immediately removes all genomes lifted from B to Q except for the global maximum $(1, m, \ldots, m)$ which has fitness $m + 1$ by ensuring they have fitness less than one.

V. Discussion — Part 1

It is of course possible to simplify the fitness functions in our examples. We decided to use more involved fitness terms, terms that without closer inspection might more easily pass for those one might expect to encounter in "real-world" applications, to try and promote plausibility.

The critical constraint we rely on may seem far fetched, but in highly constrained scheduling problems with large numbers of variables that are overseen by systems using evolutionary techniques, it can certainly be the case that a complex set of constraints winds up inducing simple or unusual constraints like ours without anyone consciously realizing it. Our best effort at formulating a problem instance where our constraints might make sense runs as follows. Assume mission critical or fail safe software processes $c_1, \ldots, c_n$ are currently running on hardware processors $h_2, \ldots, h_{m-1}$ in a real time system. Suppose that processors h_1 and h_m are now to be brought online while processes c_1 and c_n are upgraded such that for load balancing purposes c_1 should migrate to h_1 but cannot migrate to h_m and, similarly, c_n should migrate to h_m but cannot migrate to h_1.

If one does accept our thesis that fitness landscapes with pathological topologies can lead to situations where standard evolutionary algorithms are bound to fail, then as a byproduct of our examples we have an argument in favor of adopting a richer set of mutation operators, especially those that promote repeated applications of point mutation or transposition.

VI. Related Work

In this section we consider related work on the relationship between fitness landscapes and optimisation problems. Unfortunately, most of the work considers only *unconstrained* optimisation. For comprehensive surveys of empirical approaches to characterising fitness landscapes see Malan [3] and Pitzer [4].

A. Models

Several different models for fitness landscapes have been proposed in the literature including additive fitness landscapes [5], random fitness landscapes [6], the block model [7], and the NK model [8]. Additive fitness landscapes are single peaked. In contrast, in a random fitness landscape there is no correlation between the fitnesses of mutational neighbours, hence such landscapes are considered rugged and tend to have many peaks. In the block model, the genotype is composed of blocks of genes which independently contribute to the overall fitness *i.e.*, the fitness of the genotype depends on the contribution from each block. The NK model comprises genotypes with N genes. It depends on the parameter K, where $K \leq N - 1$, signalling that the fitness contribution of

each gene depends on its interactions with a block of K other genes. Thus K also serves as an indicator of the ruggedness of the fitness landscape.

B. Time to Convergence

Another related avenue of research is using Markov chain theory to analyse the behaviour of evolutionary algorithms and predict how long it will take for a Markov chain representing the different states reached in the search space (*e.g.*, the search space history) to achieve stationarity. Hernandez *et al.* [9] use coupling from the past to detect time to convergence, while Propp *et al.* [10] propose a sampling algorithm based on the idea of coupling. Since, in theory, reaching stationarity requires infinite time, Propp and Wilson provide an algorithm that can detect when stationarity has been reached in finite time. Their work was later extended by Hernandez [9].

C. Problem Hardness

Much of the theoretical work relating fitness landscapes to problem hardness has taken place within the context of biological or evolutionary landscapes. Organismal biologists seek to understand the physical, biochemical and physiological basis of genotype to phenotype mappings, while evolutionary biologists study evolutionary causes and consequences. In these situations what matters most is whether the landscape is rugged or smooth and the degree of epistasis (the interaction between genes that are not alleles [11]) occurring in genomes.

In combinatorial optimisation, features of the fitness landscape that may have an impact on problem hardness have been estimated empirically using fitness landscape characterisation metrics [12]. These features pertain to the existence of local optima, global optima, and plateaus. Assuming the optimization objective is maximisation, given a search space S and a neighbourhood relation N, a local optimum occurs at a point $s_l \in S$ if for any solution $s_n \in N(s_l)$, $F(s_l) \geq F(s_n)$. A global optimum occurs at a point $s_g \in S$ if for all $s \in S$, $F(s_g) \geq F(s)$. A plateau is defined as a set $P \subseteq S$ such that for all $s_p \in P, F(s_p) = k$, where k is a constant. (A technical condition for ensuring *connectedness* is also needed, but it will not concern us here.) A plateau indicates that the landscape is neutral, and the progress of a gradient-based search algorithm, such as an evolutionary algorithm, potentially stagnates. Counteracting such stagnation requires special measures (see, for example, Barnett [13]).

Landscape modality also figures into problem hardness. Modality is a feature of fitness landscapes that encompasses the number of local optima, the distribution of the points where they occur, and the nature of their respective basins of attraction [14]. In a search space S equipped with neighborhood relation, a local optimum s_l, the basin of attraction for s_l is defined as the set of all $s \in S$ such that there is a hill climb starting at s that ends at s_l. More precisely, a *path* in S is a finite sequence $_1, \ldots, s_k$ in S such that $s_{i+1} \in N(s_i)$ for $1 \leq i < k$ and a hill climb from s to s_l is a path such that $s_1 = s$, $s_k = s_l$ and $F(s_{i+1}) \geq F(s_i)$ for $1 \leq i < k$. The number of basins of attraction and their relative sizes in a multi-modal landscape have been found to determine how difficult it is for a gradient-based search algorithm to find a global optimum among all the local optima it encounters (see Horn [15]). While on one hand finding global optima in unimodal problems can be difficult if plateaus dominate the landscape, on the other hand in some highly multi-modal landscapes it can be easy to find global optima for both hill-climbing algorithms and evolutionary algorithms if, for example, the modes themselves "lean" towards a global optimum.

Ruggedness is another feature that has been found to affect the performance of gradient-following algorithms. An optimisation problem is considered easier to solve using either local search or an evolutionary algorithm if highly correlated parts of the landscape form easy-to-follow gradients to the optima [16]. As mentioned previously, in rugged landscapes neighbouring solutions have uncorrelated fitnesses which makes it harder for a search method to infer a search direction from previous solution quality. When the landscape is smoother and the correlation between the fitnesses of neighbouring solutions is high, there are persistent gradients (*i.e*, long paths) for the solver to follow. Because there is little correlation between neighbouring solutions, gradients in a rugged fitness landscape are not persistent which, in turn, suggests numerous local optima.

VII. Discussion — Part 2

The literature in the previous section on theoretical and empirical investigations of fitness landscapes and their relationship to optimisation problems is focused on problems without constraints. It provides a backdrop for providing further insight into our examples. Our search space S is a finite set of n-tuples and the neighborhood relation of interest is 1-point mutation, so $N(s) = \{P_{i,j}(s_n) | 1 \leq i \leq n, 1 \leq j \leq m\}$. This equips S with the edit distance metric, where two points are a distance k apart if they differ in exactly k positions or, in our notation, if one can be transformed into the other using a k-fold composition of 1-point mutations.

For our minimal counterexample the high quality solutions Q of the form $(1, *, 3)$ and the low quality solutions B of the form $(2, *, 2)$ are plateaus. Their genomes can be viewed as parallel lines in the $3 \times 3 \times 3$ lattice cube. These lines are edit distance two apart. The four parallel lines that are edit distance one from $(2, *, 2)$ (*viz.* $(2, *, 1), (2, *, 3), (1, *, 2), (3, *, 2)$) all lie in the infeasible region I. Figure 3 shows a schematic of this. More importantly, in *all* of our examples k-fold crossover is closed on both Q and B. That is every k-fold crossover operator takes $Q \times Q$ to itself and $B \times B$ to itself. Hence searching in any direction from B does not reach genomes in Q unless search operators such as 2-fold mutation operators (*e.g.*, 2-point mutation) or doubly transitive permutation operators (*e.g.*, 2-fold transposition or a transposition composed with a 1-point mutation) are introduced.

Our three scaled examples increase the size of B relative to Q and also do a better job of filling out the search space with feasible solutions, meaning as m and n increase the ratio of the

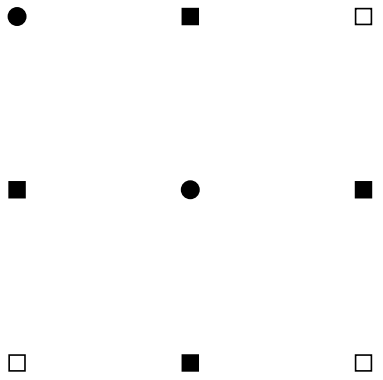

Fig. 3. A schematic showing the endpoints in the xz-plane of the nine parallel lines of the form $(x_0, *, z_0)$ for the $n = m = 3$ case. The filled circles are the two lines of feasible solutions (upper left is Q, center is B). The squares are the seven lines of infeasible solutions. The filled squares show the four lines that are edit distance one from B.

size of $Q \cup B$ to the size of I increases. Genomes in Q and B continue to remain at least edit distance two apart. In example IV-B, Q and B remain plateaus. In example IV-C, Q remains a plateau while B becomes a basin of attraction for the local minimum $(t, t, \ldots, t)$ where $t = \lfloor \frac{m+1}{2} \rfloor$. In example IV-D, B remains a plateau while Q has a global maximum occurring when the genome is $(1, m, \ldots, m)$. The situation in IV-D is a bit more complicated than that. There is a putative local maximum of $1/2$ at all genomes of the form $(1, Z, *, \ldots, *, m)$ except $(1, m, \ldots, m)$, where $2 \leq Z \leq m$, and a basin of attraction for one of them has been "punctured" in such a way that $(1, m, \ldots, m)$ yields the isolated global maximum. Since B is a plateau every genome in B yields a local maximum, so another way to phrase what is happening is to say, all the local maxima, putative or otherwise, of Q save one are smaller than all the local maxima of B.

Finally, if our quest was for a minimal counterexample, the reader may wonder why we didn't use just the constraint $a(1) = 1$ if and only if $a(3) = 3$ which, using our lines notation, would enlarge the pool of base solutions from $B = \{(2, *, 2)\}$ to $B = \{(2, *, 2), (2, *, 1), (3, *, 1), (3, *, 2)\}$. There are two reasons. First, this would admit the possibility of the transposition operator $T_{1,3}$ lifting a genomes of the form $(3, *, 1)$ from B to Q. Second, this would increase the number of "subspaces" invariant under crossover so that populations with genomes restricted to B, any one of the lines in B, or any pair of lines in B' that agree in one coordinate (*e.g.*, $\{(2, *, 2), (3, *, 2)\}$) would all be invariant under crossover.

VIII. Conclusion

We have considered how pathological fitness landscapes affect the success of evolutionary algorithms in finding global optima in constrained optimisation problems. We formulated examples to show how constraints can shape fitness landscapes in such a way that regions of high quality solutions become unreachable from regions of lower quality solutions when using the standard search operators. Our examples stem from the software deployment problem. We presented a minimal counterexample and generalized it to provide several examples with real-world parameters as well as a more plausible narrative for the problem instances. The unexpected byproduct is that our examples provide a compelling argument for including iterated transposition and iterated point mutation among the set of search operators when using evolutionary algorithms to find solutions to highly constrained optimization problems.

References

[1] A. Aleti, "Designing automotive embedded systems with adaptive genetic algorithms," *Automated Software Engineering*, vol. 22, no. 2, pp. 199–240, 2015. doi: 10.1007/s10515-014-0148-0

[2] N. R. Sabar and A. Aleti, "An adaptive memetic algorithm for the architecture optimisation problem," in *Australasian Conference on Artificial Life and Computational Intelligence*. Springer, 2017. doi: 10.1007/978-3-319-51691-2_22 pp. 254–265.

[3] K. M. Malan and A. P. Engelbrecht, "A survey of techniques for characterising fitness landscapes and some possible ways forward," *Information Sciences*, vol. 241, pp. 148–163, 2013. doi: 10.1016/j.ins.2013.04.015

[4] E. Pitzer and M. Affenzeller, "A comprehensive survey on fitness landscape analysis," *Studies in Computational Intelligence*, vol. 378, pp. 161–191, 2012. doi: 10.1007/978-3-642-23229-9_8

[5] R. Mani, R. P. S. Onge, J. L. Hartman, G. Giaever, and F. P. Roth, "Defining genetic interaction," *Proceedings of the National Academy of Sciences*, vol. 105, no. 9, pp. 3461–3466, 2008. doi: 10.1073/pnas.0712255105

[6] S.-C. Park and J. Krug, "Evolution in random fitness landscapes: the infinite sites model," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 04, p. P04014, 2008. doi: 10.1088/1742-5468/2008/04/P04014

[7] A. S. Perelson and C. A. Macken, "Protein evolution on partially correlated landscapes," *Proceedings of the National Academy of Sciences*, vol. 92, no. 21, pp. 9657–9661, 1995. doi: 10.1073/pnas.92.21.9657

[8] S. A. Kauffman and E. D. Weinberger, "The nk model of rugged fitness landscapes and its application to maturation of the immune response," *Journal of theoretical biology*, vol. 141, no. 2, pp. 211–245, 1989. doi: 10.1016/S0022-5193(89)80019-0

[9] G. Hernandez, K. Wilder, F. Nino, and J. Garcia, "Towards a self-stopping evolutionary algorithm using coupling from the past," in *Proceedings of the 2005 conference on Genetic and evolutionary computation*. ACM, 2005. doi: 10.1145/1068009.1068112 pp. 615–620.

[10] J. G. Propp and D. B. Wilson, "Exact sampling with coupled markov chains and applications to statistical mechanics," *Random structures and Algorithms*, vol. 9, no. 1-2, pp. 223–252, 1996. doi: 10.1002/(SICI)1098-2418(199608/09)9:1/2<223::AID-RSA14>3.0.CO;2-O

[11] R. A. Fisher, "The correlation between relatives on the supposition of mendelian inheritance." *Transactions of the royal society of Edinburgh*, vol. 52, no. 02, pp. 399–433, 1919. doi: 10.1017/S0080456800012163

[12] I. Moser, M. Gheorghita, and A. Aleti, "Identifying features of fitness landscapes and relating them to problem difficulty," *Evolutionary computation*, 2016. doi: 10.1162/EVCO_a_00177

[13] L. Barnett, "Netcrawling-optimal evolutionary search with neutral networks," in *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, 2001. doi: 10.1.1.32.9203 pp. 30–37.

[14] J. Garnier and L. Kallel, "Efficiency of local search with multiple local optima," *SIAM Journal on Discrete Mathematics*, vol. 15, pp. 122–141, 2002. doi: 10.1137/S0895480199355225

[15] J. Horn and D. Goldberg, "Genetic algorithm difficulty and the modality of fitness landscapes," in *Foundations of Genetic Algorithms*, 1994. doi: 10.1.1.31.3340 pp. 243–269.

[16] P. Stadler and W. Schnabl, "The landscape of the traveling salesman problem," *Physics Letters A*, vol. 161, pp. 337 – 344, 1992. doi: 10.1016/0375-9601(92)90557-3