Applying Elementary Landscape Analysis to Search-Based Software Engineering

Guanzhou Lu, Rami Bahsoon, Xin Yao
School of Computer Science, University of Birmingham
Birmingham B15 2TT, UK
{*G.Lu, R.Bahsoon, X.Yao*}*@cs.bham.ac.uk*

***Abstract*—Recent research in search-based software engineering (SBSE) has demonstrated that a number of software engineering problems can be reformulated as a search problem, hence search algorithms can be applied to tackle it. However, most of the existing work has been of empirical nature and the techniques are predominately experimental. Therefore in-depth studies into characteritics of SE problems and appropriate algorithms to solve them are necessary. In this paper, we propose a novel method to gain insight knowledge on a variant of the next release problem (NRP) using elementary landscape analysis, which could be used to guide the design of more efficient algorithms. Preliminary experimental results are obtained to indicate the effectiveness of the proposed method.**

I. INTRODUCTION

In the current state of Search-based software engineering [1], a wide range of search-based optimisation techniques have been successfully developed and applied to a number of software engineering activities, right across the life-cycle from requirements engineering to software testing [2], including local search, simulated annealing, genetic algorithms, etc. As a new flourishing research area it is being natural that the existing methodology in SBSE has been predominately experimental and lacks in-depth theoretical work except for a few cases [3]. However, to allow the future development of the field requires a deeper understanding of problem and algorithm characteristics.

In order to acheive this goal, fitness landscape characterisation shows a lot of promise. Since no matter what algorithm is employed and what problem is investigated, it is the fitness landscape that captures the nature of the relationship between these two. So far little attention in the literature has concerned the analysis of the fitness landscapes arising from software engineering problems.

There is a special class of fitness landscapes termed "elementary landscapes" [4]. Elementary landscapes possess a unique characteristic where the objective function is an eigenfunction of the graph Laplacian induced by the search operator. This has resulted in several properties, which could be applied implictly or explicitly to design a novel more successful algorithm for a particular problem.

In particular, Whitley et al. [5] observed an interesting consequence that in most practical elementary landscapes studied, the objective function for a particular candidate solution can be written as a linear combination of a subset of a collection of components. A good example is TSP, in which a fitness function is a linear combination of edge weights. As a landscape is not necessarily an elementary landscape, this property enables the construction of elementary landscapes, where both the objective function and the search operator should be appropriately designed as well.

There are a number of software engineering problems sharing the property that the fitness function is linearly decomposable, e.g. a variant of the Next Release Problem (NRP) [6]. In this paper, we propose a methodology to apply the elementary landscape analysis to gain insight knowledge on a variant of NRP. The insight knowledge gained could be used to construct more suited algorithms for this class of problems. We choose to analyze the Sampling Hill Climbing (SHC) algorithm [7], since only a single operator is involved in this algorithm that makes it simple and clear for analysis.

The main contributions of this paper include:

- We develop a method to construct elementary landscapes and carry out the elementary landscape analysis to gain insight knowledge.
- We show how the insight knowledge gained could be applied to design a novel better algorithm.

The remainder of the paper is organised as follows. Section 2 is a brief introduction to elementary landscapes. The problem formulation that we choose to analyze follows in Section 3. In Section 4 we propose the method in detail. Section 5 presents the results from the experiments. Finally, we conclude this paper in Section 6.

II. ELEMENTARY LANDSCAPES

Elementary landscapes [4] are a special class of fitness landscapes, which possess certain properties that could be applied to design novel more successful algorithms. We could characterise software engineering problems that obey certain constraints using elementary landscape analysis, which would give insight knowledge on both selection and design of the algorithms.

A. Fitness Landscapes

The notion of fitness landscapes [8] has been studied extensively in both evolutionary biology and evolutionary optimisation. It has proved to be very powerful in evolutionary optimisation theory, particularly in understanding the

978-0-7695-4195-2/10 $26.00 © 2010 IEEE
DOI 10.1109/SSBSE.2010.10

behaviour of search algorithms for combinatiorial optimisation problems and in predicting their performance.

Formally, the fitness landscape of a problem instance for a combinatorial problem is defined by a triple (X, N, f), where X is a set of candidate solutions, the objective function $f : X \longmapsto \Re$ asigns a real-valued fitness to each point in X and the neighbourhood operator $N : x \longmapsto N(x)$ imposes a neighbourhood structure among X. Given a candidate solution $x \in X, N(x)$ is the neighbourhood set that can be reached by one application of the operator.

B. Elementary Landscapes and Properties

Grover [9] first observed that the landscapes of certain combinatorial optimisation problems such as Travelling Salesman Problem (TSP), could be characterised by a wave equation. Stadler [4] gave a definition to this kind of landscape and named it "Elementary Landscape". A landscape is elementary when the objective function is an eigenfunction of the laplacian of the graph induced by the neighbourhood operator [4].

From a simpler perspective, Whitley et al. [5] provided a wave equation in terms of the expected value of the neighbours, which more concretely expressed the properties of elementary landscapes. Suppose x is some fixed but arbitrary element of X, y is an element drawn uniformly at random from the neighbourhood set $N(x)$ of x and $\overline{f}$ is the mean value over all solutions in X. On an elementary landscape, the following wave equation holds.

$$E[f(y)] = f(x) + \frac{k}{d}(\overline{f} - f(x))$$

for some k which is fixed for the entire landscape. Since y is drawn uniformly at random, the expected value of the fitness value of a neighbour y is always equal to the average fitness value over all solutions in the neighbourhood [5].

Barnes et al. [10] classifed the elementary landscapes in smooth and rugged. The wave equation holds for smooth elementary landscapes has resulted in several properties, which include relative smoothness, constraints on certain plateaus structures and local optima, as well as allowing for predictions about the fitness values of partial or full neighbourhoods during search, etc. In addition, arbitrary fitness landscapes can be decomposed into a superposition of elementary landscapes.

Additionally, landscapes with this property tend to be relatively smooth when contrasted to other combinatorial optimisation problems with well-studied local move operators, which could be considered to be an advantage for local search algorithms [5].

The wave equation also imposes constraints on the structure of local optima and precludes the existence of certain plateaus structure. One of the following observations by Codenotti and Margara [11] is true.

- if $f(x) = \overline{f} \quad f(x) = E[f(y)] = \overline{f}$
- if $f(x) < \overline{f} \quad f(x) < E[f(y)] < \overline{f}$
- if $f(x) > \overline{f} \quad f(x) > E[f(y)] > \overline{f}$

Grover [9] observed similar consequences. Let Z_{min} and Z_{max} be a local minimum and a local maximum, respectively. Then

$$Z_{min} < \overline{f} < Z_{max}$$

In other words, all local minima lie below the average function value of the search space.

Whitley et al. [5] also proved that for a plateau P on a (non-flat) elementary landscape, if $x \in P$ has only equal and disimproving neighbours, then there cannot exist a solution $z \in P$ with only equal and improving neighbors. A plateau is a set P of candidate solutions in X such that for all $a, b \in P$, $f(a) = f(b)$ and there is a path $(a = x1, x2, ..., xk = b)$ such that $x_{i+1} \in N(x_i)$. Plateaus (also known as neutral networks) are structural features that arise in many combinatorial problems [12]. Plateau structure is a challenge for local search that can cause the algorithm to cease progress.

We have seen that the expected fitness value of the full neighbourhood can be predicted by the wave equation. Moreover, we could expand a partial neighbourhood during search, and make predictions for the remaining neighbourhood. This property gives significant insight knowledge to the search algorithm that could be explicitly applied in designing algorithms.

In addition, arbitrary fitness landscapes can be decomposed into a superposition of "elementary landscapes" via a Fourier series expansion. A series expansion $f(x) = \sum_{i=1}^{N} \alpha_i \varphi_i(x)$, where φ_i forms a complete and orthonormal system of eigenfunctions of the graph laplacian, is termed a Fourier series expansion of the objective function. This decomposition is helpful in a sense that some statistical properties of the landscape could be computed and the decomposed elementary landscapes can be studied individually. The information about the effective hardness of an elementary landscape is contained in the relative ordering of the associated eigenvalues [13].

C. Component-based Model

Whitley et al. [5] observed several interesting consequences arise from the expected value equation. In most practical elementary landscapes studied, the objective function for a particular candidate solution can be written as a linear combination of a subset of a collection of components. A good example is TSP, in which a fitness function is a linear combination of edge weights. Let C be a set of real valued components and there exists $C_x \subset C$ such that $f(x) = \sum_{c \in C_x} c$. The set C_x is referred to as the intracomponents of a solution x and the set $C - C_x$ as the intercomponents of x. When a local search move has been made from an incumbent solution x to a neighbouring solution, an exchange of components is made. In particular, a subset of the intracomponents is removed and a subset of the intercomponents is added.

On this basis, Whitley et al. [5] constructed a component-based model that can be used to characterise a neighbourhood structure. In this model, the neighbourhood size is regular and denoted by d. The model consists of the following equations.

$$\overline{f} = p3 \sum_{c \in C} c$$

$$E\{f(y)\} = f(x) - p1 f(x) + p2(\sum_{c \in C} c - f(x))$$

$$= f(x) - p1 f(x) + p2(\frac{1}{p3}\overline{f} - f(x))$$

where $0 < p1 < 1$ is the proportion of the intracomponents that that are removed from the solution in one move, $0 < p2 < 1$ is the proportion of the intercomponents that are added to the solution in a move. Finally, $0 < p3 < 1$ is the proportion of the total components in C that contribute to the cost function for any randomly chosen solution, which is independent of the neighborhood size. Whitley et al. [14] proposed a component theorem:

Theorem 1: If $p1, p2$ and $p3$ (must be constants) can be defined for any regular landscape such that the evaluation function can be decomposed into components where p1 = α/d and p2 = β/d and

$$\overline{f} = p3 \sum_{c \in C} c = \frac{\beta}{\alpha + \beta} \sum_{c \in C} c$$

then the landscape is elementary.

III. THE NEXT RELEASE PROBLEM (NRP)

The Next Release Problem (NRP) was originally formulated by Bagnall et al. [6]. The variant of the NRP studied in this paper is a representative of a class of software engineering problems where the fitness functions could be linearly decomposed. The problem is formulated as follows.

Given a software product, let R denote a set of candidate requirements to be considered to implement for the next release of the software, each $r \in R$ has an associated cost(r) which is a measure of the resouce consumption to implement it. and a weight w_i which reflects the requirement's importance. Also there is a budget for the total cost of the implemented requirements.

Associated with R, there is a directed acyclic graph G = (R,E) where $(r_i, r_j) \in E$ iff r_i is a prerequisite of r_j, G is also transitive since $(r_i, r_j) \in E \wedge (r_j, r_k) \in E \Rightarrow (r_i, r_k) \in E$. If the company decides to satisy requirement r_i, it must satisfy the prerequisites of r_i. In a special case where no requirement has any prerequisite $E = \emptyset$, we say the problem is basic.

Assuming there are n requirements, the problem faced is to find a subset S of R, the cardinality of S is fixed and is k, such that

$$\sum_{r_i \in S} w_i \text{ is maximised,}$$

$$\sum_{r_i \in S} cost(r_i) \text{ is minimised.}$$

Different search algorithms have been applied to NRP [6], but they were all experimental work. There is no analysis of whether the obtained results are good and whether they could be improved. There is no analysis either what characteristics the NRP has and whether the search algorithms used are appropriate.

IV. PROPOSED METHOD FOR ANALYSING SE LANDSCAPES

A. Overview of the Proposed Method

The elementary properties possessed by certain fitness landscapes are promising and could be applied to improve the performance of certain algorithms on particular problems. Initially, a fitness landscape is not necessarily elementary, and thus we will need to modify either the fitness function or the neighbourhood operator to construct an elementary landscape. In addition, to the best of our knowledge, the fitness function should be linearly decomposable in order to enable the construction of an elementary landscape.

In this section, we give a detailed description of our proposed method, with a case study on how the elementary landscape analysis could be applied to the Sampling Hill Climbing (SHC) algorithm on a variant of the Next Release Problem (NRP).

We categorize the elementary properties into two classes. One is implicit, which are inherent given the landscape is elementary and do not affect the design of the algorithm, e.g. relative smoothness. The other one is explicit, which can be explicitly applied while designing algorithms e.g. allow prediction for partial neighbourhoods. Here is an overview of the proposed method.

a. For a given software engineering problem where its fitness function is linearly decomposable, pick up a search algorithm and develop a fitness function and a neighbourhood operator for the selected algorithm.
b. Carry out the elementary landscape analysis.
c. Apply the insight knowledge to the initial algorithm and develop the improved algorithm.
d. Evaluate the performance of the improved algorithm.

B. Initialisation

As a simple but effective local search algorithm, and involving only a single move operator, Sampling Hill Climbing (SHC) is selected as the initial search algorithm. This algorithm works by moving from an inital solution to a local optima providing the move is improving. In each iteration, it simply samples, randomly, a number of solutions from the neighbourhood and take the best of them. The algorithm might get stuck when each of the samples is worse than the current solution, we chose to continue the search by accepting the best move irrespective of whether or not it is improving. The algorithm will terminate after a fixed number of iterations.

In each local search algorithm, there is an objective function to guide the search. According to the problem formulation of a variant of the Next release problem (NRP), the objective function can be defined as

$$f(S) = \sum_{R_i \in S} w_i + Budget - \sum_{R_i \in S} cost(R_i).$$

With respect to the local search operator, the initial choice is 2-exchange, which will randomly exchange two requirements in S and $R - S$.

C. Elementary Landscape Analysis

First of all, we apply the component-based model to determine whether the fitness landscape generated by the initial algorithm is elementary or not.

The objective function defined above is a combination of weights and costs, which is similar to Whitley's observation of components. Let the set $w_i - cost(R_i)$ make up the set of components C, where $|C| = |R|$, we could apply the component-based model and component theorem to prove whether the induced landscape is elementary or not.

We first compute $p3$ and $\overline{f}$, since k components have been picked up to contribute to the objective function.

$$p3 = \frac{k}{|R|}, \quad \overline{f} = p3 \sum_{c \in C} c = \frac{k}{|R|}(\sum_{R_i \in S} w_i - \sum_{R_i \in S} cost(R_i) + |R| * Budget).$$

To compute $p1$ note there are k components in any solution, and two-exchange changes exactly 2 components. Therefore $p1 = 2/k$.

To compute $p2$ note there are $|R| - k$ components with the components in $f(x)$ removed and 2 new components are picked from these.Therefore $p2 = \dfrac{2}{|R| - k}$.

Adding the terms to the component-based model yields:

$$Avg\{f(y)\} = f(x) - p1 f(x) + p2(\frac{1}{p3}\overline{f} - f(x))$$

$$= f(x) - \frac{2}{k} f(x) + \frac{2}{|R| - k}(\frac{|R|}{k}\overline{f} - f(x))$$

$$= f(x) + \frac{2|R|}{(|R| - k)k}(\overline{f} - f(x))$$

Hence the fitness landscape induced is elementary.

D. Apply the Insight Knowledge

Given the fact that the landscape is elementary, certain explicit elementary properties could be applied in a form of heuristics to replace certain components of the initial algorithm. In the initial Sampling Hill Climbing (SHC) algorithm, it randomly samples N solutions from the neighbourhood. The size of samples has a large impact on the search behaviour - expanding the size is more likely to find an improving move. Since one of the following observations is true for elementary landscapes

- if $f(x) = \overline{f} \quad f(x) = E[f(y)] = \overline{f}$
- if $f(x) < \overline{f} \quad f(x) < E[f(y)] < \overline{f}$
- if $f(x) > \overline{f} \quad f(x) > E[f(y)] > \overline{f}$

When $f(x) < \overline{f} \quad f(x) < E[f(y)]$, one can be sure that a neighbourhood includes an improving move (Assume maximisation). A significantly smaller sample size could have identified an improving move under this circumstance.

When $f(x) > \overline{f} \quad f(x) > E[f(y)]$, one cannot be sure that a neighbourhood includes an improving move, however, elementary properties allow for expanding a partial neighbourhood and predict for the remaining neighbourhood. This prediction is expected to guide the search to a more promising direction and reduce the time wasted in less promising expansions and evaluations. Here is a sketch of the algorithm.

Algorithm 1 Elementary Sampling Hill Climbing.

```
Randomly generate an initial solution x.
for K iterations do
    if f(x) <= f̄ then
        Sample A solutions in the neighbourhood and take
        the best move among them, A << N;
    else
        Expand a partial neighbourhood of size B, compute
        the expected value of the remaining neighbour-
        hood.
        if Exp[Remaining neighbourhood] > f(x) then
            Sample C solutions in the remaining neigh-
            bourhood and take the best move among them,
            (B + C) << N;
        else
            Take the best move in the partial neighbourhood;
        end if
    end if
end for
return local optima found;
```

V. COMPUTATIONAL STUDIES

To evaluate the performance of the algorithm incorporated with the insight knowledge obtained by the elementary

Table I
ALGORITHM PARAMETERS

PARAMETERS	VALUES
Number of iterations K	100
Initial neighbourhood size N	10^{-3} * full neighbourhood size
Neighbourhood size A	0.1 * N
Partial neighbourhood size B	0.6 * A
Neighbourhood size C	0.4 * A

landscape analysis, both Elementary Sampling Hill Climbing (ESHC) and Sampling Hill Climbing (SHC) have been implemented in MATLAB. We have carried out an empirical study to pick up the value of different neighbourhood sizes that can find the solution of best quality. Table I presents the algorithm parameters obtained.

To rate the effectiveness of ESHC requires comparison with SHC on different problem instances, in terms of the quality of the best solution found and the time consumed. By varying the size of candidate requirements set $|R|$ and the selected requirements set $|S|$, we have studied 16 problem instances. These synthetic problem instances are randomly generated according to the data sets generator described in [15]. For each problem instance (PI), 50 algorithm runs are performed. The experimental results are listed in Table II, in the first column, the value after slash is the size of candidate requirements set R and the ratio before slash specifies the proportion of R to be selected.

In order to measure the performance of ESHC, we have performed a statistical analysis on the experimental data above using T-test with confidence level at 95%. From which we could show that the algorithm running time of ESHC is significantly less than that of SHC, while there is no significant difference between the quality of the best solution found by both algorithms, on each problem instance studied.

The experimental results show that the Sampling Hill Climbing algorithm incorporated with elementary properties outperforms the initial SHC with far less running time while being able to find roughly the same optimal solution. As described in Section IV, the insight knowledge gained from the elementary landscape analysis can lead the search to focus on promising moves that prevent wasting time in non-promising exploitations. Hence we suppose this is where the performance improvement in time comes from. The experimental results presented in this work are still preliminary, since only one elementary property has been used, which is the predictions for the fitness values of the partial or full neighbourhood. However, if more elementary properties could be appropriately applied to the algorithms, it shows some promise that the elementary landscape analysis can be useful to construct a more suited algorithm for a particular problem, or a class problems sharing certain similarities.

Table II
PERFORMANCE AVERAGES AND STANDARD DEVIATIONS OF 50 RUNS OF ESHC AND SHC ON 16 PROBLEM INSTANCES

	BEST SOLUTION		TIME	
PROBLEM INSTANCE	ESHC	SHC	ESHC	SHC
PI-1 (10%/50)	31.2±0.88	25.8±2.17	0.03±0.0007	0.08±0.003
PI-2 (20%/50)	59.2±2.4	61.9±1.9	0.15±0.0036	0.26±0.0027
PI-3 (50%/50)	117±1.5	118.6±1.56	0.36±0.0077	0.66±0.0015
PI-4 (80%/50)	146.2±2	146.1±2.74	0.15±0.0035	0.26±0.0009
PI-5 (10%/100)	74.76±0.43	72±1.02	0.6±0.0026	1.42±0.0028
PI-6 (20%/100)	137.1±1.1	137.9±0.81	2.43±0.05	4.72±0.025
PI-7 (50%/100)	269.4±0.85	270.3±0.79	6.06±0.15	11.9±0.12
PI-8 (80%/100)	326.1±1.3	327.1±1.1	2.5±0.07	4.78±0.04
PI-9 (10%/200)	149.9±0.76	150.5±0.71	14.1±0.33	28.4±0.12
PI-10 (20%/200)	241.1±0.57	241.7±0.5	43.8±1.03	92.9±0.37
PI-11 (50%/200)	536.6±0.6	536.8±0.5	108.9±2.4	231.3±0.4
PI-12 (80%/200)	370.4±0.55	371±0.71	64.3±0.1	91.7±0.16
PI-13 (10%/500)	357.7±0.48	358.5±0.71	81.65±3.11	178.22±3.70
PI-14 (20%/500)	675.2±0.84	675.4±5.37	241.4±5.37	551.57±1.66
PI-15 (50%/500)	1301.2±1.30	1302±0.45	593.93±12.4	1376.75±17
PI-16 (80%/500)	1601.8±1.30	1604±0.00	249.17±0.23	557.54±1.1

VI. CONCLUSION

In this work we have developed a fitness landscape analysis method to gain insight knowledge on certain software engineering problems. The main goal is to characterise a class of fitness landscapes sharing certain similarities, which would give more insights that could be applied to construct more suited algorithms for particular problems. So far the proposed method is applicable for problems where the objective functions could be linearly decomposed into components.

We carried out a case study to analyze the effectiveness of the proposed method, which is Sampling Hill Climbing (SHC) on a variant of the Next Release Problem (NRP). We found out that if the objective function of a software engineering problem is linearly decomposable, it is possible to construct an elementary landscape and apply elementary properties to design a better algorithm for this problem. The experimental results show that SHC incorporated with elementary properties outperforms the initial algorithm. Therefore we could assume that the performance of algorithms for a particular problem could be improved by the applicaiton of the elementary landscape analysis.

The future work will include exploiting other elementary properties that could be applied to the algorithms, and extending this method to more software engineering problems.

Acknowledgement

This work was partially supported by an EPSRC grant (No. EP/D052785/1) on "SEBASE: Software Engineering By Automated SEarch."

References

[1] M. Harman and B. F. Jones. Search-based software engineering. *Information and Software Technology*, 43:833–839, 2001.

[2] M Harman. The current state and future of search based software engineering. In *FOSE '07: 2007 Future of Software Engineering*, pages 342–357, Washington, DC, USA, 2007. IEEE Computer Society.

[3] P. K. Lehre and X. Yao. Runtime analysis of the (1+1) ea on computing unique input output sequences. *Information Sciences*, 2010.

[4] P. F. Stadler. Towards a theory of landscapes. In R. Lopéz-Peña, R. Capovilla, R. García-Pelayo, H. Waelbroeck, and F. Zertuche, editors, *Complex Systems and Binary Networks*, volume 461, pages 77–163, Berlin, New York, 1995. Springer Verlag.

[5] D. Whitley, A. M. Sutton, and A. E. Howe. Understanding elementary landscapes. In *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 585–592, New York, NY, USA, 2008. ACM.

[6] A. J. Bagnall, V. J. Rayward-Smith, and I. M. Whittley. The next release problem. *Information and Software Technology*, 43(14):883 – 890, 2001.

[7] L.D. Davis. Bit-climbing, representational bias, and test suite design. In *The Fourth International Conference on Genetic Algorithms*, pages 18–23, San Mateo, CA, USA, 1991.

[8] S.Wright. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In *Proc. 6th Congr. Genetics*, volume 1, page 365, 1932.

[9] Lk. Grover. Local search and the local structure of NP-complete problems. *Operations Research Letters*, 12(4):235–243, OCT 1992.

[10] J. W. Barnes, B. Dimova, S. P. Dokov, and A. Solomon. The theory of elementary landscapes. *Applied Mathematical Letters*, 16:337–343, 2002.

[11] B. Codenotti and L. Margara. Local properties of some np-complete problems. Technical report, ICST, 1992.

[12] J Frank, P Cheeseman, and J Stutz. When gravity fails: local search topology. *J. Artif. Int. Res.*, 7(1):249–281, 1997.

[13] P. F. Stadler and G. P. Wagner. Algebraic theory of recombination spaces. *Evol. Comput.*, 5(3):241–275, 1997.

[14] D. Whitley and A. M. Sutton. Partial neighborhoods of elementary landscapes. In *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 381–388, New York, NY, USA, 2009. ACM.

[15] A Finkelstein, M Harman, S. A Mansouri, J Ren, and Y Zhang. A search based approach to fairness analysis in requirement assignments to aid negotiation, mediation and decision making. *Requir. Eng.*, 14(4):231–245, 2009.