A Foundational Study on the Applicability of Genetic Algorithm to Software Engineering Problems

Hsinyi Jiang\textsuperscript{1}, Carl K. Chang\textsuperscript{1}, Dan Zhu\textsuperscript{2}, and Shuxing Cheng\textsuperscript{1}
\textsuperscript{1}Department of Computer Science
\textsuperscript{2}College of Business
Iowa State University
\{hsinyij, chang, scheng\}@cs.iastate.edu, dzhu@iastate.edu

Abstract

Many problems in software engineering (SE) can be formulated as optimization problems. Genetic algorithm (GA) is one of the more effective tools for solving such optimization problems and has attracted the attention of SE researchers in recent years. However, there is a general lack of sound support theory to help SE researchers investigate the applicability of GA to certain classes of SE problems. Without such a theory, numerous attempts to conduct a wide spectrum of experiments for solution validation appear to be ad hoc and the results are often difficult to generalize. This paper reports a foundational study to develop such a support theory. Some preliminary results are also given.

Keywords: evolutionary computation, genetic algorithm, Markov Chain, optimization, software engineering

1 Introduction

Genetic algorithms (GAs) were studied by John Holland [23] in the early 1970s. His original work was the study of cellular automaton which is a discrete model often investigated in computability theory, mathematics, and theoretical biology. Within the model, each cell has a finite number of states. Time is also considered to be discrete in that study. Obviously, GAs has inherited those features and become a popular optimization method in use today.

More and more researchers now apply GAs as the optimization method to their specific problem domains. Therefore, the foundational study on how well GAs can perform for each of the varying problem domains becomes more and more crucial. Although many papers have proposed various approaches to analyzing the behaviors of GAs, most existing GA theory lacks concern with its practicality. For instance, the convergence of GAs has been studied by Rudolph [32]. However, it is only meant for a theoretical study since the time is assumed to go to infinity. In the real world, it is impossible for any application to wait for infinite time to obtain the optimal solution. To the best of our knowledge, applications of GAs to certain problem domains are rarely studied with the support of a foundational study in the past. This is particularly true for the problem domains in software engineering (SE).

We define the SE optimization problem as the problem in SE which can be solved by more than one feasible solution. There is at least one criterion to evaluate solutions, and the goal of the problem is to search the best solution within the domain of all the feasible solutions. Note that in order to frame the optimization problem in SE as a computable problem, the factors considered in the problem need to be quantified as parameters and the criteria of the problem need to be specified as a function with value range pertaining to the physical meaning (such as project parameters) in SE.

Since 1995 [12] [13], GAs have attracted the attention of SE researchers more and more. Problems, such as optimal scheduling, optimal test data generation, and optimal module clustering, etc., were suggested as targets to be solved with GAs. Some researchers in these areas claimed to have provided empirical results to show that their application of GAs are appropriate and made the conclusion that GAs can find better solutions compared to other methods (e.g., [4] and [26]). However, simply presenting seemingly positive experimental results is not convincing since the trials may be biased due to the limited scope of experiments, and the inconclusive nature of determining local or global optima in such experiments. In general, a sound support theory to help SE researchers investigate the applicability of GAs to SE optimization problems is overdue. Such a theory can be treated as a bridge between the theoretical results and the applicability of GAs. Without such a theory, numerous attempts to conducting a wide spectrum of experiments for solution validation appear to be ad hoc and the results are often difficult to generalize.
Basically, there may exist some problem domains which have similar common features as the SE problems. The proposed theory can also be applied to those fields. However, due to the nature of our research, our primary goal is to analyze the utility levels of GAs and improve the performance of GAs with respect to the SE problems.

In this paper, we exploit the transition matrix which has been used to prove the convergence of canonical genetic algorithms (CGAs) of the problems illustrated in [32], define the criteria with a probability-theoretic perspective to decide the hardness of GAs applied to the problems of SE, show that, as a preliminary research result, the convergence rate of such problems is related to one of the eigenvalues of the transition matrix with its physical meaning explained, and provide some findings of common features in the SE optimization problems. With such a quantitative analysis, we hope to better speculate and eventually derive a complete picture of SE problems in which GAs are applicable. As a foundational study, this paper intends to draw attention of other SE researchers to collectively investigate the applicability theory of GA in SE, instead of simply applying GA to SE with unfounded optimism.

The paper is organized as follows: in section 2, a literature survey is given; in section 3, the background, preliminary assumptions, and existing theorems are provided; in section 4, the current research ideas and the findings on common features in SE problem domain are proposed; and in section 5, the paper is summarized with the future work suggested.

2 Literature Survey

2.1 Optimizations in SE Problem Domain

In [13], the authors surveyed some existing work and found that metaheuristic techniques (GA is one of them) are widely applied in three areas of SE - test data generation, module clustering, and cost/effort prediction. The optimal scheduling problem is one of the typical problems within a sub-area of SE called project management (PM) [9] [10] [12] [18]. It was explored for the purpose to devise ways to intelligently match employees to tasks with respect to the factors gathered in the beginning development of a software project. Usually, the factors include salaries of employees (cost), capabilities of employees, learning curves of employees, and potential hazard levels of assignments [38], etc. The researchers can decide which one to focus and it depends on their research goals.

In the traditional waterfall model, testing is a key component of the verification and validation (V&V) activities [29]. Verification is to inspect whether specific functions are built correctly in the software, and validation is to examine whether the software meets customer requirements. To conduct testing related to a piece of software, various strategies have been proposed. Some of those are formulated as optimization problems [2] [3] [6] [7] [13] [36]. For instance, with regard to the branch testing in a structural testing problem [13], each branch of the code is quantified, and the researchers measure the “distance” of branches based on those quantities. If they plan to obtain the closest branch to the intended branch, they can properly design the objective function and optimize it.

Owing to the rapid development of hardware and network technologies, more and more industrial applications are equipped with highly complex software systems, which often consist of a large number of components. To streamline the design phase, a large scale system can be designed hierarchically. Instead of directly integrating all of the components into a system, a set of congruent components are first grouped into a subsystem. The resulting software system is then composed of a set of subsystems. How to cluster the original set of components into several subsystems has attracted the attention of researchers [11]. Oftentimes, they try to maximize cohesion inside a component and minimize coupling among components. Therefore, this kind of problems is also a class of optimization problems in SE.

While academic approaches to optimization problems in SE are many, a commercial tool called @Risk (Palisade) [31] is a decision tool for software project managers. The RISKOptimizer [31] uses GAs to come up with trial solutions, and then automatically performs a Monte Carlo simulation on each trial solution. With it, the users can explicitly include the uncertainties, and the result shows all possible outcomes to users.

It is believed that there are still a lot of optimization problems which have not yet been explored in the SE problem domain [20] [28] since only the “trade-off” strategy is considered [29]. Those problems can be converted into optimization problems. Intuitively, the more the optimization problems can be discovered and formulated in SE, the more important the proposed work is in SE as it is the fundamental to the applicability of GA.

Researchers investigating GAs have tried to understand and predict the behavior of GA since it was invented. Based on our literature review, it is found that many attempts have failed in some specific problems. We conclude the survey of the previous work and classify them into two categories: analyzing the behaviors of GAs using Markov chains, and analyzing the behaviors of GAs using other methods.

2.2 Analyzing the Behaviors of GAs Using Markov Chains

Some of GA researchers have exploited finite state homogeneous Markov chains to analyze the behavior of GA. There are two kinds of transition matrices which are com-
monly adopted.

Michael D. Vose and Joe Suzuki et al. have applied the transition matrix with the states representing the occurrences of some individuals [34] [35] [37]. The cardinality of different populations, i.e., the dimension of the state space, becomes [34]

$$|S| = \binom{m + 2^l - 1}{m},$$

where $S$ is the state set, $m$ is the size of a population, and $l$ is the length of binary strings.

Günter Rudolph and David B. Fogel [16] [32] et al. have proposed the transition matrix in which the states are defined by every possible configuration of an entire population of bit strings. Therefore, there are $2^m$ states, where $m$ is the size of a population, and $l$ is the length of binary strings.

Although both of them possess similar concepts and functionalities, and they can be converted to each other, the representations are different. In our view, each of the approaches has its own advantages. In Vose’s approach, we can get the distributions of the individuals easily. But the histories of individuals from one generation to another may not be easily obtained in Vose’s approach. However, those histories can be acquired in Rudolph’s approach.

### 2.3 Analyzing the Behaviors of GAs Using Other Methods

In 1975, Holland introduced the notion of schemas to formalize the informal notion of “building blocks” [14] [27]. His building block hypothesis states that GAs attempt to find highly fit solutions to the problem by the short, low-order, and above-average schemata. However, the schema theory merely shows us a rough idea in high level that better performing schemata will receive an increasing number of trials in the next generation. It does not give us much information about the detailed analysis of the behaviors of GAs. Moreover, in 1987, the term deception problem was coined by Goldberg. It is said that a problem is deceptive if certain hyperplanes guide the search toward some solutions or genetic building block that is not globally competitive [19]. If we compare this statement with the building block hypothesis, the deception problem seems to contradict to Holland’s building block hypothesis in some sense. Hence, the deception problems are considered as the “hard” problems for GAs. Nevertheless, the deception is neither sufficient nor necessary conditions for the problems that are hard for GAs [21].

Actually, Goldberg isolated the deception problems from Bethke’s work [5] [17]. Bethke used discrete Walsh functions to analyze the fitness functions of GAs. He developed the Walsh-Schema transform to calculate schema average fitness efficiently and used it to characterize functions as easy or hard for GAs to optimize. This method is helpful to get some ideas for solving a problem. However, sometimes, it is hard to convert functions to Walsh polynomials. Moreover, this method only analyzes the flat population, where every possible string is assumed to be represented in equal proportion [8]. This type of static analysis fails to capture the more dynamic aspects of GAs. Hence, some people proposed another approach called the Nonuniform Walsh Transform [8]. However, as in the case of Walsh transformation, it is hard to convert the functions to those forms.

In general, the developed GA theory shows that it is difficult to fully capture the behaviors of GAs, especially in finite time with different types of fitness functions. Besides that, GA theory still remains in the theoretical level. It is not easy to directly generalize the theory to application level. Our narrowing down of the problem domain helps us to comprehend more in both theory and application of GAs. In this study, our objective is to develop a support theory that is deemed necessary to effectively bridge the gap between evolutionary computation, where GA plays an important role, and SE.

### 3 Review of Related Theories and Research Assumptions

Based on the literature review, we know that the behaviors of GAs are dynamic. Analyzing their flat population, which is static, does not capture the complete picture. Hence, it is difficult to predict the behaviors of GAs by merely observing their fitness functions. To handle the dynamic properties of GAs, the transition matrix, adopted in [32] to show the convergence of CGAs, is often applied. With this matrix, both dynamic populations and fitness function, embedded in the selection transition matrix, are controlled.

In order to continue the discussions and show the preliminary results, the following assumptions need to be made and existing theories are reviewed.

#### 3.1 Assumptions

Our research will be based on the transition matrix in [32]. For this study, we have three assumption given below.

**Assumption 1** [Problem Definition] The problems $\text{MAX} \{ f(b) | b \in IB^l \}$, where $0 < f(b) < \infty$ for all $b \in IB^l = \{0, 1\}^l$, and $l$ is the length of the binary strings which represent feasible solutions, are discussed in this paper.

**Assumption 2** [Choice of Solution Method] The CGA, which only has selection, crossover, and mutation opera-
itors, with the best solutions maintained, is the algorithm analyzed in this paper.

From [32], it is obtained that a CGA with the best solutions maintained will eventually converge to its global optimal solution. Since our goal is to determine whether or not GAs are good tools for the problems in SE, the number of generation (n) cannot be arbitrarily large. Otherwise, it is only a matter of time. Hence, Assumption 3 is necessary.

Assumption 3 The number of generations of GAs is a reasonably large number, and it is fixed.

3.2 The Transition Matrices

In the following, we briefly review the transition matrices of finite homogeneous Markov chains proposed in [32] and some properties.

In CGAs, three operators (selection, crossover, and mutation) are applied.

The selection operator, in which the roulette wheel selection (proportional selection) is assumed, forms a transition matrix $S$. Suppose we have $m$ (an even number) individuals for each generation. By Assumption 1, $S$ is a $2^{ml} \times 2^{ml}$ matrix with the element

$$s_{ij} = \frac{\prod_{k=1}^{m} f(\pi_k(j))}{(\sum_{k=1}^{m} f(\pi_k(i)))^m}$$

if \(\{\pi_1(i), \pi_2(i), \ldots, \pi_m(i)\}\) \(\subseteq\) \(\{\pi_1(j), \pi_2(j), \ldots, \pi_m(j)\}\), where $\pi_k(i), k \in \{1, \ldots, m\}$ are the $k$th segment of length $l$ from the state $i$ and $f(\cdot)$ is the fitness function. Otherwise, $s_{ij} = 0$.

By the same token, the crossover transition matrix $C$ is also a $2^{ml} \times 2^{ml}$ matrix. Since we have $m$ individuals in each generation, there are at most $m!$ ways to couple the individuals. Note that order is considered in this paper. Let $I_i$ be the index set in which each element is a binary string with length $ml$ representing a method to couple the individuals of state $i$. If $p_c$ is the crossover probability, then

$$c_{ij} = (1 - p_c)\delta_{ij} + p_c \cdot \frac{1}{|I_i|},$$

$$\sum_{r \in I_i} \prod_{k=1}^{m} P\{Cr(\pi_{2k-1}(r), \pi_{2k}(r)) = (\pi_{2k-1}(j), \pi_{2k}(j))\},$$

where

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases},$$

$Cr(\cdot, \cdot)$ is the result of the crossover operation, and $P[\cdot]$ is the probability of the event. In (1),

$$P\{Cr(\pi_{2k-1}(r), \pi_{2k}(r)) = (\pi_{2k-1}(j), \pi_{2k}(j))\}$$

depends on the crossover method applied.

Bit mutation is designed to serve as a background operator to ensure that all possible alleles can be entered to the population [16]. It is important since once the population of chromosomes reaches a configuration such that crossover no longer produces offspring outperforming their parents, it is the only operator which leads the population to leap out of the homogeneous populations. The mutation transition matrix is denoted as $M$, which is a $2^{ml} \times 2^{ml}$ matrix. Let $P_m \in (0, 1)$ be the mutation probability, then

$$m_{ij} = \prod_{k=1}^{m} P_m^{H(\pi_k(i), \pi_k(j))} (1 - P_m)^{1 - H(\pi_k(i), \pi_k(j))},$$

where $H(\cdot, \cdot)$ is the Hamming distance of the strings (chromosomes).

CMS (the product of $C$, $M$, and $S$) is a matrix of order $2^{ml}$ and forms a positive stochastic matrix [32]. Therefore, it is regular (primitive). In order to show that the CGA with best solution maintained converges to its global optimum, the state space is extended from $2^{ml}$ to $2^{ml+1}l$. In the new transition matrix $P$, there are $2^l$ square transition matrices CMS in the diagonal.

$$P = \begin{pmatrix} CMS & CMS & \cdots \\ CMS & CMS & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

$$= \begin{pmatrix} D_1 & D_2 & \cdots \\ D_2 & D_1 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix},$$

where $D_i, i \in \{1, 2, \ldots, 2^l\}$ is the $i$th diagonal CMS matrix, and they represent fitness values in descending order. That is, the square matrix $D_1$ represents the highest fitness value, and the square matrix $D_2$ represents the second highest fitness value, and so on.

The upgrade matrix $U$ is for maintaining the best solution. If the state in $D_i$ has the best fitness value which is higher than the fitness value represented by $D_i$, this state is upgraded to $D_j, j < i$ in which the fitness value is equal to the best fitness value of the state. The structure of $U$ is

$$U = \begin{pmatrix} U_{11} & U_{12} & \cdots \\ U_{21} & U_{22} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix},$$

which is a lower triangular matrix. Moreover, since it is assumed that there is a unique optimal solution for the optimization problem in [32], $U_{11}$ is the only $2^{ml} \times 2^{ml}$ identity matrix. That is, none of the states in $D_1$ needs to be upgraded.
3.3 The Existing Theorems and Lemma

**Theorem 1** [25] [Perron’s Formula] If \( A \) is a square matrix of order \( r \), \( \lambda_1, \lambda_2, \ldots, \lambda_q, q \leq r \), are the eigenvalues of \( A \), and \( m_1, m_2, \ldots, m_q \) are the (algebraic) multiplicities of the eigenvalues, respectively, \( m_1 + m_2 + \cdots + m_q = r \), then

\[
a_{ij}^{(n)} = \sum_{k=1}^{q} \frac{1}{(m_k - 1)!} \left( \frac{d^{m_k-1}}{d\lambda^{m_k-1}} \left( \prod_{\lambda \neq \lambda_k} (\lambda - \lambda_k)^{m_k} \right) \right)_{\lambda = \lambda_k},
\]

where \( a_{ij}^{(n)} \) is the element in the \( i \)th row and \( j \)th column of \( A^n \), and \( A_{ij}(\lambda) \) is the element in the \( i \)th row and \( j \)th column of the adjoint of the matrix \((\lambda I_r - A)\), that is, \( A_{ij}(\lambda) \) is equal to the product of \((-1)^{i+j}\) and determinant of the minor of \((\lambda I_r - A)_{ij} \). Note that the minor of \((\lambda I_r - A)_{ij} \) is derived by deleting the \( j \)th row and \( i \)th column from the matrix \((\lambda I_r - A)\), and \( I_r \) is the identity matrix of order \( r \).

**Theorem 2** [25] If \( A \) is a regular matrix, then there exists a real eigenvalue \( \lambda_1 > 0 \) which is simple (i.e., of algebraic multiplicity 1) and which exceeds the absolute values of all other eigenvalues of \( A \).

**Lemma 1** [25] If \( A \) is a stochastic matrix, i.e., \( A \) is non-negative and the sum of the elements in any row of \( A \) is 1, then the eigenvalues of \( A \) are in absolute value at most equal to 1. Moreover, 1 is an eigenvalue of \( A \).

**Theorem 3** [24] [Schur] If \( A \) is a square matrix of order \( r \) with eigenvalues \( \lambda_1, \lambda_2, \ldots, \lambda_r \) in any prescribed order, there is a unitary matrix \( S \) (square matrix) of order \( r \) such that

\[
S^{*}AS = T = [t_{ij}]
\]

is an upper triangular, with diagonal entries \( t_{ii} = \lambda_i \) \( i = 1, 2, \ldots, r \). That is, every square matrix \( A \) is unitarily equivalent to a triangular matrix whose diagonal entries are the eigenvalues of \( A \) in a prescribed order.

4 Preliminary Results and Discussions

Based on the above review of existing theorems and lemma, we resolve to proposing the following new definition of “GA-solvable” in the context of solving optimization problems in the SE domain.

4.1 GA-Solvable Problems and Convergence Rates

**Definition 1** Given a large (fixed) number \( n \) and a small positive (fixed) number \( \epsilon \), a problem of SE is GA-solvable if after \( n \) generations, a GA will converge to its global optimal solution with the probability equal to or greater than \( 1 - \epsilon \).

The reason that we define GA-solvable this way is that the problems of SE, such as scheduling and testing, allow GAs to fail, but the eventual success rate must be high.

Based on this definition, the following solution is suggested to determine GA’s hardness of a problem in SE.

From Theorem 1 [Perron’s Formula], without loss of generality, we can bound \( a_{ij}^{(n)} \) by estimating the term,

\[
\left( \frac{d^{m_k-1}}{d\lambda^{m_k-1}} \left( \prod_{\lambda \neq \lambda_k} (\lambda - \lambda_k)^{m_k} \right) \right)_{\lambda = \lambda_k},
\]

where \( k \in \{1, 2, \ldots, q\} \), as follows. Since \( A_{ij}(\lambda) \) is equal to the product of \((-1)^{i+j}\) and determinant of the minor of \((\lambda I_r - A)_{ij}\), it is a polynomial of order \((r - 1)\). Let

\[
f(\lambda) = \frac{\lambda^n A_{ij}(\lambda)}{\prod_{\lambda \neq \lambda_k} (\lambda - \lambda_k)^{m_k}}.
\]

Then we can get the order of the function \( f(\lambda) \) is \( n + (r - 1) - (r - m_k) = n + (m_k - 1) \). Therefore, the order of \((m_k - 1)\)th derivative of \( f(\lambda) \) in equation (2) is \( n \). Moreover, a multiplier of \( O(n) \) is applied every time during the process in which the derivative is obtained. Hence, it is concluded that if \( A \) is a square matrix of order \( r \), \( \lambda_1, \lambda_2, \ldots, \lambda_q \), \( q \leq r \), are the eigenvalues of \( A \), and \( m_1, m_2, \ldots, m_q \) are the (algebraic) multiplicities of the eigenvalues, respectively, \( m_1 + m_2 + \cdots + m_q = r \), then there exists a positive number \( K \) such that

\[
|a_{ij}^{(n)}| \leq K n^{m_{\text{max}} - 1} |\lambda_{\text{max}}|^n,
\]

where \( a_{ij}^{(n)} \) is the element of \( A^n \) in the \( i \)th row and \( j \)th column, \( m_{\text{max}} = \max\{m_1, m_2, \ldots, m_q\} \), and \( |\lambda_{\text{max}}| = \max\{|\lambda_1|, |\lambda_2|, \ldots, |\lambda_q|\} \). This inequality will be exploited later to show that the convergence rate is related to one of the eigenvalues of the transition matrix.

From section 3, the transition matrix of the CGA with the best solution maintained is \( PU \). Since it is a lower triangular matrix, we can get that the eigenvalues of \( PU \) are the eigenvalues of the diagonal blocks. Moreover, in the matrix \((PU)^n\), where \( n \) is the fixed number of generations, the elements of the first \( 2^m \) columns are the probabilities of convergence to the global optimum. Hence, to compute the probability of convergence to the global optimum, we have to sum up those elements. As an alternative, we compute the summation of the elements in the columns other than the first \( 2^m \) columns.

In fact, the matrix \( PU \) can be represented as

\[
PU = \begin{pmatrix} CMSU_{11} & 0 \\ R & A \end{pmatrix},
\]

where

\[
R = \begin{pmatrix} CMSU_{21} \\ \vdots \\ CMSU_{2,m} \end{pmatrix},
\]

and

\[
CMSU_{11} = CMSU_{21} = CMSU_{22}.
\]
and
\[
A = \begin{pmatrix}
CMSU_{22} & CMSU_{23} & CMSU_{22} \\
CMSU_{32} & CMSU_{33} & CMSU_{32} \\
\vdots & \vdots & \vdots \\
CMSU_{2n,2} & CMSU_{2n,3} & \cdots & CMSU_{2n,2}
\end{pmatrix}.
\]

Then,
\[
(PU)^n = \left( \sum_{i=0}^{n-1} A^i R(CMSU_{11})^{(n-1)-i} A \right)^n.
\]

Since A is a square matrix, for the element \(a_{ij}^{(n)}\) of \(A^n\), the inequality (3) holds. Hence, we have
\[
\sum_{i=1}^{2^{(m+1)}-2^m} \sum_{j=1}^{2^{(m+1)}-2^m} a_{ij}^{(n)} \leq (2^{2^m} K \sum_{i=1}^{2^m} i^{m_{\text{max}}-1} |\lambda_{\text{max}}|^n), \quad (4)
\]

where \(m_{\text{max}}\) is the maximal (algebraic) multiplicity of the eigenvalues of A, and \(|\lambda_{\text{max}}|\) is the maximal absolute value of the eigenvalues of A. The following shows \(|\lambda_{\text{max}}| < 1\).

Since the upgrade matrix upgrades some columns of A to the first \(2^m\) columns of \(PU\), the sum of each row of A is less than 1. Therefore, if \(\lambda\) is an eigenvalue of \(A = (a_{ij}^{(1)})\) and \(u = (u(i))\) is its left eigenvector, we have
\[
\lambda u(j) = \sum_{i=1}^{2^{(m+1)}-2^m} u(i) a_{ij}^{(1)}, \quad j = 1, 2, \ldots, 2^{(m+1)}-2^m.
\]

That is,
\[
\sum_{j=1}^{2^{(m+1)}-2^m} |\lambda u(j)| \leq \sum_{i=1}^{2^{(m+1)}-2^m} |u(i)| \sum_{j=1}^{2^{(m+1)}-2^m} a_{ij}^{(1)} < \sum_{i=1}^{2^{(m+1)}-2^m} |u(i)|.
\]

Hence, \(|\lambda| < 1\). Since the eigenvalues of \(PU\) are the eigenvalues of the diagonal blocks, we can get that the set of eigenvalues of \(A\) is contained in the set of eigenvalues of \(PU\). Moreover, since \(CMSU_{11}\) is a regular (stochastic) matrix (Theorem 3 in [32]), from Lemma 1 and Theorem 2, we obtain that \(\lambda_1 = 1\) is an eigenvalue of \(CMSU_{11}\), i.e., \(\lambda_1 = 1\) is an eigenvalue of \(PU\), and \(\lambda_1\) is simple. If we agree that the eigenvalues, \(\lambda_1, \ldots, \lambda_k\), \(k \leq 2^{(m+1)}\) of \(PU\) are in the order \(1 = \lambda_1 > \lambda_2 \geq \cdots \geq \lambda_k\) with \(m_1, m_2, \ldots, m_k\) being the (algebraic) multiplicities, respectively, the following inequality can be derived.

\[
\sum_{i=1}^{2^{(m+1)}-2^m} \sum_{j=1}^{2^{(m+1)}-2^m} a_{ij}^{(n)} \leq K' n^{m_{\text{max}}-1} |\lambda_2|^n,
\]

where \(K' > 0\) and \(m_{\text{max}}\) is the maximal (algebraic) multiplicity of the eigenvalues of \(PU\). That is, regardless of the initial distribution of the populations, the probability of convergence to the global optimum is greater than or equal to \(1 - K' n^{m_{\text{max}}-1} |\lambda_2|^n\).

If \(K' n^{m_{\text{max}}-1} |\lambda_2|^n > 1\), that means either the fixed \(n\) is not large enough, or \(\lambda\) is not small enough, and this equation is meaningless. For a sufficiently large fixed \(n\), the smaller value the \(\lambda_2\) has, the larger the convergence probability.

Hence, for a problem in SE, we first need to compute its transition matrix \(CMS\), extend it to \(PU\), find the eigenvalue \(\lambda_2\), which is related to the convergence rate of the problem, of \(PU\), and then apply \(\lambda_2\) to \(K' n^{m_{\text{max}}-1} |\lambda_2|^n\).

With the value \(K' n^{m_{\text{max}}-1} |\lambda_2|^n \leq 1\), we can obtain the GA’s hardness of the problem.

This above result is not only a theoretical one, but also amenable to empirical study. The application of the above result to experiments in SE is illustrated in section 4.3.

4.2 Eigenvalues of Transition Matrices and Fix Points of Genetic Algorithm

In the previous subsection, it is shown that the convergence rate is related to the second largest eigenvalue of the transition matrix. This subsection provides the relationship between the second largest eigenvalue of the transition matrix and the fix points of GA.

Theorem 3 demonstrates that every square matrix is unitarily equivalent to a triangular matrix whose diagonal entries are the eigenvalues of the matrix. Hence, \((PU)^n\) (in generations of GA) can be written as

\[
(NU)^n = (ST S^*)^n = ST^n S^*,
\]

where \(S\) is a unitary matrix and \(T\) is upper triangular. The diagonal entries of \(T\) are the eigenvalues of \(PU\). Let the eigenvalues of \(PU\) be denoted as \(\lambda_1 = 1, \lambda_2, \ldots, \lambda_{2^{(m+1)}}\) (in descending order). Then \(T^n\) is an upper triangular matrix with diagonal entries \(\lambda_1^n, \lambda_2^n, \ldots, \lambda_{2^{(m+1)}}^n\). Moreover, the trace of \((PU)^n\) is given by

\[
\text{Trace}((PU)^n) = \text{Trace}((ST^n S^*))
\]

\[
= \text{Trace}(S^* ST^n) = \text{Trace}(T^n) = \sum_{i=1}^{2^{(m+1)}} \lambda_i^n. \quad (6)
\]
By the inequality
\[ | \sum_{i=1}^{2^{(m+1)t}} \lambda_2^n | \leq | \sum_{i=1}^{2^{(m+1)t}} |\lambda_2|^n | \leq 1 + (2^{(m+1)t} - 1) |\lambda_2|^n, \]
it is obtained that
\[ \text{Trace}((PU)^n) \leq 1 + (2^{(m+1)t} - 1) |\lambda_2|^n. \] (7)
That is, the trace of the transition matrix is bounded by \(\lambda_2\). Note that the trace of the transition matrix represents the probabilities of the fix points of GA. For searching purpose, we don’t want GA to stay within a state too long since the state is already visited. The revisiting of a state will cost some time without any improvement of solutions. In other words, if the probabilities of fixed points of GA can be reduced, the probability of GA to search other candidate states will be increased for the number of the states is finite. Hence, the smaller the trace, the better the GA performance is. According to the inequality (7), bounding \(\lambda_2\) is equivalent to bounding the trace of the transition matrix. And it explains the result in the previous subsection that if \(\lambda_2\) is smaller, the convergence of GA with respect to the optimization problem will be faster.

4.3 The Implementation of Experiments in SE

To obtain the eigenvalue \(\lambda_2\) of \(PU\), it is usually difficult for computing since \(PU\) is a large square matrix. Sometimes it is necessary to apply theorems in mathematics to reduce the complexities of computations. Intermediate Value Theorem, for instance, is a practical one. With this theorem, we can get
\[ a < \lambda_2 < b, \]
where \(a, b \in [0, 1]\), and the smaller the interval \([a, b]\), the more accurate the result. Therefore, we can get the range of the convergence probability for \(PU\). This range can be used to estimate the expected value derived below.

Suppose the convergence probability for \(PU\), a transition matrix of an SE optimization problem, is \(1 - \delta\), where \(\delta\) is corresponding to the aforementioned \(K^{n\delta^{n+1}} |\lambda_2|^n\). Then we need
\[ \sum_{i=1}^{\infty} i \cdot \delta^{i-1} (1 - \delta) = \frac{1}{1 - \delta}, \] (8)
the average number of times to run GAs, to get the optimal solutions. Since we already know the range for \(1 - \delta\), we can get the expected value for GAs to generate the optimal solutions. This expected value can be a useful reference for the researchers to predict the number of times required to run GAs to get the optimal solutions with some specific settings in SE experiments. Based on the derivation of the expected value, the following definition is equivalent to Definition 1 and it is more practical.

Definition 2 Given a large (fixed) number \(n\) and a small positive (fixed) number \(\epsilon\), a problem of SE is GA-solvable iff the expected value (i.e. the average number of times for GAs to get the optimal solutions) is less than or equal to \(\frac{1}{\epsilon n}\) with \(n\) generations per run.

From Definition 2, the average number of times for GAs to get the optimal solutions for SE problems can be estimated. In fact, the performance of GAs to SE problems can be better in the real world since SE problems allow some tolerance. In the next subsection, the preliminary finding on computing the applicabilities of GAs with thresholds and some common features in SE problem domain are discussed.

4.4 Common Features in SE Problem Domain

Generally, we believe that there are some common features in the SE optimization problem domain. Two of them are concerned most in current research phase - thresholds, and multi-objective optimizations.

4.4.1 Thresholds of the Optimization Problems in SE

For most of the optimization problems in SE, although it is certainly preferable to reach the exact optimal solutions, it is not a necessity since most of the problems in SE generally accept some tolerance of measurements. Near-optimal solutions, defined in Definition 3, may be good enough, as in the case of many other engineering problems. Even so, it is essential to study the behavior of GA in order to understand the applicability of GA to SE problems.

Definition 3 Given a threshold in terms of the proportion of all the feasible solutions in a SE problem, a solution is called a near-optimal solution to the SE problem iff the proportion of the solution among all the feasible solutions, sorted by fitness values in ascending order, is greater than or equal to the threshold.

Note that the definition of near-optimal solutions in SE is slightly different from most of the literature in which the behaviors of GAs are investigated. For most of the literature, the near-optimal solution is defined as the neighborhoods of the global optimal solutions according to the Hamming distance. In our perspective (the perspective in SE), each solution, whose fitness value is greater than or equal to the threshold, is a near optimal solution. For example, in Figure 1 (X-axis and Y-axis represent two parameters of a particular fitness function, and Z-axis represents fitness values),
if we set the threshold to 0, the solutions with nonnegative fitness values are near-optimal solutions. As we can see, the fitness values of local and global optimal solutions are greater than 0. In this case, all solutions contained in the neighborhoods of both local and global optimal solutions are near optimal solutions.

Figure 1. The Curved Surface of a Fitness Function

Problems in different areas in SE require different thresholds. For instance, if the problems we consider are scheduling problems, the thresholds can be looser than the risk minimization problems since there may be a safety concern.

With our analysis method, no matter how the threshold is exploited for the problem, if it can be converted to the percentage of the number of all the feasible solutions, we can compute the probability of convergence to near-optimal or optimal solutions as well.

If the threshold is $X\%$, the top $1 - X\%$ of the fitness values are considered as acceptable solutions. Since there are $2^i$ fitness values, we can get that $(1 - X\%) \cdot 2^i$ is the number of top $1 - X\%$ of the fitness values, but it may not be a natural number. Let $k = \max\{i \in N | i \leq (1 - X\%) \cdot 2^i\}$, and set $U_i$, $i = 1, 2, \ldots, k$ to the identity matrices of order $2^{ml}$, then in the transition matrix $PU$, the first $2^{ml}$ $k$ columns are corresponding to the states of near-optimal or optimal solutions. If we repeat the computations in section 4.1, the probability of convergence to near-optimal or optimal solutions can be obtained. Since we can get the probability of convergence to near-optimal or optimal solutions, Definition 2 in section 4.3 can be also applied here.

4.4.2 Multi-objective Optimizations

Because of the “trade-off” strategy, the SE problems often have more than one objective. Other than single objective optimization, the multi objective optimization has more than one element in which two elements are not guaranteed to be comparable [33]. For example, with project scheduling problems, one may want to maximize the quality of products, and minimize cost, time, and risk. For testing problems, one may attempt to maximize quality and minimize time [1]. Quality, cost, time, and risk are not necessarily comparable. Some researchers have normalized them by dividing their maximal values based on the historical data. However, the projects are different from one to another. We cannot precisely categorize the projects and determine the similar projects from the historical data for the current one. Hence, this method may not be suitable.

In fact, normalization is not the only approach for multi-objective optimizations [15]. How to properly design the multi-objective fitness function is another problem for the applicability of GAs to the SE problem domain. In the future, the fitted scale rule with respect to the SE problem domain will be investigated.

5 Conclusion and Future Work

GAs have been applied to solve the optimization problems in SE for a decade. Theoretical proofs and discussions with respect to SE applications were rarely presented in the literature. That means, every time GAs are applied, experiments are inevitable. To conduct the experiments, we need to collect some data in SE projects first. Oftentimes, such data are not readily available. Validations will be hard in that case. Moreover, for most of the problems in SE, the optimal solutions are not trivial. To the best of our knowledge, existing literature seldom shows that the so-called optimal solutions found by running GAs are really near-optimal or optimal solutions. The solutions may just be local optima. Hence, doing the experiments merely is not appropriate. The only way to solve this problem is to lay down a theoretical foundation. With this view, GAs can be applied to the problems in SE without any experiment to validate the solution method.

In this paper, we proposed a measure to predict the hardness of GAs to the optimization problems in SE. With this measure, we get an expected value which is the average number of times required for SE researchers running GAs to obtain the optimal or near-optimal solutions. Moreover, we provide the physical meaning of the measure so that the foundation of the measure can be well understood. In addition, some common features in SE problem domain are discussed. Our preliminary research result shows that SE researchers who apply GAs to solve their optimization problems can predict how many times they need to run GAs (i.e., via Definition 2) and get optimal solutions based on the expected value (i.e., (8)) obtained. Also, they can recognize why the measure is utilized for predicting purpose. However, from the empirical perspective, there are limitations of
the measurement. Sometimes, if the matrix $PU$ is too large, it will be difficult to obtain its $\lambda_2$. Therefore, an alternative strategy, discussed in section 4.3, is necessary. Another limitation is related to the precisions of computing tools (for instance, computers). If $\lambda_2$ of $PU$ is too small (close to 0) so that the computing tools cannot estimate it precisely, it will be difficult to get the value of $K_{\text{r}m_{\text{max}}}^{-1}|\lambda_2|^n$. To deal with this limitation, we need an efficient algorithm.

Our future research directions are summarized as follows.

5.1 Crossover and Mutation Rates

In the preliminary result, it is already obtained that the convergence rate is related to one of the eigenvalues of the encoded transition matrix. However, we merely know the encoding of a problem impacts the eigenvalues. How the crossover and mutation rates influence the eigenvalues haven’t been studied yet. From the literature review, we found that although some researchers have considered those two operators, they only reported some results through limited experiments. Some of the researchers even provide examples and suggest that any fixed set of algorithmic parameters of GA cannot enable GA to efficiently optimize an arbitrary function [22]. Hence, the theoretical evidence on how mutation and crossover rates impact the convergence of the problem need to be studied. That will be the first goal of future work.

Our initial thought to achieve the goal is to investigate the special classes of fitness functions, and then generalize the results. Since mutation is the background operator of crossover (i.e., it is used for dealing with the convergence to local optima), we believe that the setting of mutation rates is related to different types of fitness functions. However, a lot of experiments in SE show that setting the mutation rate to 0.01 outperforms other values [18]–[30]. Hence, it is predicted that the fitness functions of SE problems may have some common features. Nevertheless, we need further research results to support it.

Since the mutation transition matrix $M$ is symmetric, and it is known that symmetric matrix has some good properties, we will attempt to investigate how the setting of mutation rates influences the matrix $PU$ with those properties. Our research goal on mutation rates will also be focused on the characteristic of symmetric properties of the mutation transition matrix $M$.

5.2 Rules for Encoding SE Optimization Problems

Since more and more SE researchers apply GA to solve the optimization problems, we believe that the problems in SE may not be hard for GA. But why GA fits most of the problems in SE has never been studied theoretically. We have already defined the term “GA-solvable”. However, a strong connection between GA and SE needs to be built. We plan to build it theoretically and define some rules for encoding SE optimization problems based on the theoretical results. With those rules, GA can be applied more precisely and the researchers will have a solid foundation and more confidence in making use of the power of GA.

References


[31] @RISK 4.5 for excel, palisade corporation, 2007.


