
Jun He
Department of Computer Science
Aberystwyth University, Aberystwyth, SY23 3DB, U.K.
jun.he@ieee.org

Tianshi Chen
Institute of Computing Technology
Chinese Academy of Sciences, Beijing 100190, China.
chentianshi@ict.ac.cn

July 29, 2011

Abstract

Population-based Random Search (RS) algorithms, such as Evolutionary Algorithms (EAs), Ant Colony Optimization (ACO), Artificial Immune Systems (AIS) and Particle Swarm Optimization (PSO), have been widely applied to solving discrete optimization problems. A common belief in this area is that the performance of a population-based RS algorithm may improve if increasing its population size. The term of population scalability is used to describe the relationship between the performance of RS algorithms and their population size. Although understanding population scalability is important to design efficient RS algorithms, there exist few theoretical results about population scalability so far. Among those limited results, most of them belong to case studies, e.g. simple RS algorithms for simple problems. Different from them, the paper aims at providing a general study. A large family of RS algorithms, called ARS, has been investigated in the paper. The main contribution of this paper is to introduce a novel approach based on the fundamental matrix for analyzing population scalability. The performance of ARS is measured by a new index: spectral radius of the fundamental matrix. Through analyzing fundamental matrix associated with ARS, several general results have been proven: (1) increasing population size may increase population scalability; (2) no super linear scalability is available on any regular monotonic fitness landscape; (3) potential super linear scalability may exist on deceptive fitness landscapes; (4) “bridgeable point” and “diversity preservation” are two necessary conditions for super linear scalability.
on all fitness landscapes; and (5) “road through bridges” is a sufficient condition for super linear scalability.

1 Introduction

Population-based Random Search algorithms (RS), such as Evolutionary Algorithms (EAs), Ant Colony Optimization (ACO), Artificial Immune Systems (AIS) and Particle Swarm Optimization (PSO), have been widely applied to solving discrete optimization problems. A common belief in this field is that the performance of an RS algorithm may improve if increasing its population size. The term of population scalability is used to describe the relationship between the performance of RS algorithms and their population size. Given a family of RS algorithms with population sizes $\mu = 1, 2, \cdots$, population scalability is measured intuitively in the ratio

$$R(\mu) := \frac{\text{Performance of RS with a population size 1}}{\text{Performance of RS with a population size } \mu}. \quad (1)$$

The basic question in population scalability is to investigate how $R(\mu)$ varies as the population sizes $\mu$. There are two different ways to measure the performance of an RS algorithm: one is the number of iterations (hitting time); another is the number of function evaluations (running time). Then population scalability may be measured intuitively in the ratios

$$R_{ht}(\mu) := \frac{\text{Hitting times of RS with a population size 1}}{\text{Hitting times of RS with a population size } \mu}, \quad (2)$$
$$R_{rt}(\mu) := \frac{\text{Running times of RS with a population size 1}}{\text{Running times of RS with a population size } \mu}. \quad (3)$$

Suppose that the number of function evaluations at each generation is fixed to $\mu$, then the difference between $R_{ht}(\mu)$ and $R_{rt}(\mu)$ is a factor $1/\mu$. So for a population size $\mu > 1$ if population scalability under running times is greater than 1, then population scalability under hitting times must be super linear, i.e. greater than $\mu$. Hence whether and when super linear population scalability under hitting times happens is an important question. Both hitting times and running times are hard to calculate rigorously, and to provide an accurate description of $R(\mu)$ is very difficult.

In evolutionary computation, the study of population scalability can be traced to early 1990s. Goldberg et al. presented a population sizing equation to show how a large population size helps an EA to distinguish between good and bad building blocks on some test problems [1]. Mühlenbein and Schlierkamp-Voosen studied the critical (minimal) population size that can guarantee the convergence to the optimum [2]. Arabas et al. proposed an adaptive scheme for controlling the population size, and the effectiveness of the proposed scheme was validated by an empirical study [3]. Eiben et al. reviewed various techniques of parameter controlling for EAs, where the adjustment of population size was
considered as an important research issue \cite{4}. Harik et al. linked the population size to the quality of solution by the analogy between one-dimensional random walks and EAs \cite{5}. The approximate population sizing models proposed by the above investigations may shed some light on deducing the "promising" population size. However, the effectiveness of the models was validated only by case studies based upon specific optimization problems.

Recently there exist a few rigorous results about population scalability. Jansen et al. studied the population scalability of the \((1 + \lambda)\) EA on three pseudo-Boolean functions, Leading-Ones, One-Max and Suf-Samp \cite{6}. Lässig and Sudholt presented runtime analysis of a \((1 + \lambda)\) EA with an adaptive offspring size \(\lambda\) on several pseudo-Boolean functions \cite{7}. Jägersküpper and Witt analyzed how the running time of a \((\mu + 1)\) EA on the Sphere function scales up with respect to the problem size \(n\) \cite{8}. Jansen and Wegener showed that the running time of the \((\mu + 1)\) EA with a crossover operator on the Real Royal Road function is polynomial on average, while that of an EA with mutation and selection only is exponentially large with an overwhelming probability \cite{9}. Witt proved theoretically that the running time of the \((\mu + 1)\) EA on a specific pseudo-Boolean function is polynomial with an overwhelming probability, when \(\mu\) is large enough \cite{10}. Storch presented a rigorous runtime analysis of the choice of the population size with respect to the \((\mu + 1)\) EA on several pseudo-Boolean functions \cite{11}. Oliveto et al. presented runtime analysis of both \((1 + \lambda)\) and \((\mu + 1)\) EA on some instances of Vertex Covering Problems \cite{12}. Friedrich et al. analyzed the running time of \((\mu + 1)\) EAs with diversity-preserving mechanisms on the Two-Max problem \cite{13}. He and Yao theoretically investigated how the running time of EAs varies when the population size is changed from 1 to \(\mu\) \cite{14}. The link between scalability and parallelism was discussed in \cite{15}. For the \((\mu + \mu)\) EA, an upper bound on the first hitting time has been obtained on two well-known unimodal problems, Leading-Ones and One-Max \cite{16}. However all these available results are mainly restricted to several simple algorithms for simple problems. In other words current studies belong to case studies \cite{17}.

Different from previous case studies, this paper aims at providing a general study on population scalability of a large family of RS algorithms, called ARS, for maximizing an objective function on a finite set (discrete optimization). ARS is an abstraction of RS algorithms consisting of global search and strict elitist selection. Because currently variant RS algorithms have been applied into a wide range of optimization problems ranging from knapsack packing to protein folding prediction, it is important and necessary to establish a universal theory which could cover RS algorithms as widely as possible.

A new approach based on the fundamental matrix is proposed to analyze population scalability. The approach is based on the following idea: the search process of an RS algorithm using elitist selection can be modeled by an absorbing Markov chain (e.g. see the analysis of EAs \cite{18,19,20}). According to the theory of absorbing Markov chains, the hitting time of the chain to absorbing states starting from a transient state equals the sum of staying times of the Markov chain in all transient states (starting from the same transient state). Thus
population scalability \( R_{ht}(\mu) \) under hitting times is equivalent to

\[
R_{ht}(\mu) := \frac{\text{Staying times of RS with a population size } 1}{\text{Staying times of RS with a population size } \mu}. \tag{4}
\]

A trouble in applying formulas (2), (3) and (4) is that hitting times and staying times vary as a Markov chain starts from different transient states. Therefore \( R(\mu) \) will take multiple values. A solution to this trouble is to use the fundamental matrix. For an absorbing Markov chain, each entry of the fundamental matrix stands for a staying time of the Markov chain in a transient state \[21, 22\]. Therefore spectral radius of the fundamental matrix is a natural index of measuring staying times in transient states. By taking this new index, population scalability now is measured by

\[
R(\mu) := \frac{\text{Spectral radius of RS with a population size } 1}{\text{Spectral radius of RS with a population size } \mu}. \tag{5}
\]

Thereafter this paper focuses on applying the theory of absorbing Markov chains to analyze population scalability under spectral radius.

The paper is organized as follows: Section 2 introduces ARS and gives the definition of population scalability under spectral radius. Section 3 lists several preliminary lemmas about fundamental matrix associated with ARS and also proposes time-based fitness landscapes. Section 4 analyzes population scalability of ARS. Section 5 concludes the paper and discusses future research issues.

2 Abstract Population Based Random Search and Population Scalability

2.1 Abstract Population-based Random Search

Consider the problem of maximizing an objective function \( f(x) \):

\[
\max \{ f(x); x \in S \}, \tag{6}
\]

where \( S \) is a finite set. In evolutionary computation, \( f(x) \) is often called fitness function. Let the fitness function take \( L + 1 \) values \( f_0 > f_1 > \cdots > f_L \), which are called fitness levels.

For the convenience of theoretical analysis, suppose that all constraints in the above problem have been removed through constraint handling, e.g. by a penalty function method. Under this circumstance, all solutions in \( S \) are feasible solutions. Practical or theoretical analysis of constraint handling in evolutionary computation can be found in references such as \[23, 24\].

The following notations are used in the paper.

- Denote \((\mu + \mu)\)-ARS to be an ARS algorithm using the population size \( \mu \) for both parent and children population\(^1\). When necessary, marks (1)

\(^1\)Notation \((\mu + \lambda)\) follows that used in evolutionary strategies \[25\]: \( \mu \) parent individuals generate \( \lambda \) children individuals. Then \( \mu \) individuals are selected from these \((\mu + \lambda)\) individuals to form the next generation parent. In this paper \( \mu = \lambda \) is a special case.
and (\(\mu\)) will be added in order to distinguish between (1 + 1)-ARS and \((\mu + \mu)\)-ARS algorithms, e.g. \(S^{(1)}\) and \(S^{(\mu)}\).

- \(S^{(1)} := S\) is called individual space, \(S^{(\mu)} := \prod_{i=1}^{\mu} S\) population space, \(\mu\) population size;

- \(x, y, z \in S^{(1)}\) denote individuals, \(X, Y, Z \in S^{(\mu)}\) populations. They are called states in individual and population spaces respectively. \(X\) consists of \(\mu\) individuals: \((x_1, \cdots, x_\mu)\) with \(f(x_1) \geq \cdots \geq f(x_\mu)\). The first individual is called super individual, which is used to keep the best found solution. It is also denoted by \(\pi\) for convenience. The rest of individuals are called non-super individuals.

- Let the optimal set \(S_{\text{opt}} \subseteq S\) be the set consisting of all states (individuals) which are optimal solutions to Problem \(P\) and non-optimal set \(S_{\text{nop}} := S \setminus S_{\text{opt}}\). Let the optimal set \(S^{(\mu)}_{\text{opt}} \subseteq S^{(\mu)}\) be the set of states (populations) containing at least one optimal solution to Problem \(P\) and non-optimal set \(S^{(\mu)}_{\text{nop}} := S^{(\mu)} \setminus S^{(\mu)}_{\text{opt}}\). Let \(\eta\) be the cardinality of \(S_{\text{nop}}\) and \(\eta(\mu)\) the cardinality of \(S^{(\mu)}_{\text{nop}}\).

- Let \(t\) be the time or generation counter. \(\Phi_t\) (or \(\Psi_t\)) represents the population at the \(t\)-th generation. \(\Phi_{t+1/2}\) the intermediate population generated by the search operator.

A family of RS algorithms consisting of global search and elitist selection is used to solve Problem \(P\). The procedure of RA are described below.

**Algorithm 1 Abstract Population-based Random Search (ARS)**

1. **input**: fitness function;
2. generation counter \(t \leftarrow 1\);
3. initialize \(\Phi_1\);
4. **while** (no optimal solution is found) **do**
5. \(\Phi_{t+1/2} \leftarrow\) global search (\(\Phi_t\));
6. evaluate the fitness of each individual in \(\Phi_{t+1/2}\);
7. \(\Phi_{t+1} \leftarrow\) strict elitist selection (\(\Phi_t, \Phi_{t+1/2}\));
8. \(t \leftarrow t + 1\);
9. **end while**
10. **output**: the maximal value of the fitness function.

The operators used in ARS are global search and elitist selection, described as follows.

- **Global Search**: Let \(\Phi_t = X\). For each individual \(x\) in the population \(X\), it generates a child \(y\) with a probability

\[p_m(x, y),\]

5
where the search probability satisfies the following conditions

\[ p_m(x, y) > 0, \quad \forall x, y \in S, \]  
\[ \sum_y p_m(x, y) = 1, \quad \forall x \in S. \]  

The children population is denoted by \( \Phi_{t+1/2} \).

- **Strict Elitist Selection:** Let \( \Phi_t = X \) and \( \Phi_{t+1/2} = Y \). Select \( \mu \) individuals from \( X \) and \( Y \) to form the next generation population as follows:

  **Super individual** The super individual in the next generation population is selected as follows: if the best individual in \( Y \) is better than the super individual in \( X \), then it will be select as the new super individual in the next generation population; otherwise the super individual is kept unchanged. In case that the number of such best individuals is more than one, choose one of them at random.

  **Non-super individuals** Denote the set \( U(X,Y) \) to be \( U(X,Y) := \{ (z_1, \cdots, z_{\mu-1}); z_i \in X \cup Y, i = 1, \cdots, \mu - 1 \} \).

Choose the next generation population \( Z \in U(X,Y) \) from \( U(X,Y) \) with a selection probability

\[ p_s(Z | X \cup Y), \]

where the selection probability satisfies the following condition

\[ \sum_{Z \in U(X \cup Y)} p_s(Z | X \cup Y) = 1. \]  

Let \( \Phi_{t+1} \) represent the next generation population. Without losing generality, arrange individuals in a population in the order of their fitness from high to low.

ARS is an abstraction of RS algorithms, which the population size of both parent and children is \( \mu \). It is called abstract because it merely provides an abstract requirement of search and selection operators, but no implementation detail.

Actually global search in ARS covers all types of mutation operators with the sole requirement: starting from any state, it is possible to reach any other state. Strict elitist selection in ARS covers all types of selection operators with the sole requirement: to keep the best found solution by the super individual. The first requirement is to guarantee the best solution is reachable by the search; and the second one is to guarantee the search is convergent. Both requirements are essential for RS to convergence (see convergence conditions of EAs [18, 26, 27]).
Example 1. An example of global search. Consider pseudo-Boolean optimization \([28]\).

- Bitwise Mutation: Given a binary string with the length \(n\), flip each of its bits independently with a probability \(1/n\).

Example 2. Two examples for selecting non-super individuals.

- Selection with the Highest Pressure: copy the super individual \((\mu - 1)\) times.

- Selection with the Lowest Pressure: given two populations \(X\) and \(Y\), all individuals are selected with an equal probability.

In the \((1 + 1)\)-ARS algorithm, strict elitist selection becomes

- Strict Elitist Selection: the parent will be replaced only if the offspring is better than it.

Remark: In the rest of the paper, we always take the family of \((\mu + \mu)\) EAs (where \(\mu = 1, 2, \cdots\)) as an example to illustrate general concepts and results. The \((\mu + \mu)\) EAs utilize bitwise mutation with a mutation rate \(1/n\) and strict elitist selection. The \((1 + 1)\) EA here is a little different from that in \([29]\), which utilizes non-strict elitist selection.

The stopping criteria is that the running of ARS will stop once an optimal solution is found. This criteria is taken only for the convenience of theoretical analysis. It is not necessary to have any prior knowledge about any optimal solution during the analysis. In fact the optimal solution to Problem \([6]\) is unknown because \(f(x)\) is unknown. But it does not affect our analysis.

Adaptive random search related with the time \(t\) will not be analyzed in this paper. It will be discussed in the future. Hence we always suppose that global search and elitist selection are homogeneous (i.e. independent on \(t\)) in the paper.

- Given two states \(x, y \in S^{(1)}\), the search probability \(p_m(x, y)\) is independent on both \(t\) and \(\mu\);

- Given two states \(X, Y \in S^{(\mu)}\), the selection probability \(p_s(Z | X \cup Y)\) is independent on \(t\) but dependent on \(\mu\).

2.2 Population Scalability under Spectral Radius

Given an ARS algorithm, the population sequence \(\{\Phi_t, t = 0, 1, \cdots\}\) can be modeled by an absorbing homogeneous Markov chain (e.g. see the analysis of EAs \([18, 20, 30]\)). An individuals or population including an optimal solution is called absorbing state; and other individuals or populations are called transient states. Let \(P\) be its transition probability matrix, whose entries are given by

\[P^{(\mu)}(X, Y) = P(\Phi_{t+1} = Y | \Phi_t = X), \quad \forall X, Y \in S^{(\mu)}.\]

The number of generations to find an optimal solution is called hitting time \([31, 30]\) which is defined as follows:
Definition 1. The stopping time
\[
\tau(X) := \min\{t; \Phi_t \in S_{\text{opt}}(\mu) \mid \Phi_0 = X\},
\]
is called hitting time to the absorbing set \( S_{\text{opt}}(\mu) \) starting from the initial state \( X \).

The mean value of \( \tau(X) \) is denoted by
\[
m(\mu)(X) := E[\tau(X)].
\]

\( \forall X \in S_{\text{nop}}(\mu) \), the hitting time \( m(\mu)(X) \) represents how long the Markov chain will enter the absorbing set \( S_{\text{opt}}(\mu) \) starting from the transient state \( X \). In contrast, we also define its counterpart: staying times in transient states as follows (Definition 3.2.3 in [21]).

Definition 2. Given \( X \in S_{\text{nop}}(\mu) \), \( Y \in S_{\text{nop}}(\mu) \), the staying time \( N(\mu)(X,Y) \) is defined by the mean value of the total number of generations that the Markov chain is in the transient state \( Y \) if starting in the transient state \( X \).

Let's arrange all populations in the order of their super individuals’ fitness from high to low (where the populations with the same super individual are arranged together but in any order) and write it in a vector form: \( (X_1, X_2, \cdots)^T \), then the hitting times can be written in a vector form:
\[
m(\mu) = (m(\mu)(X_1), m(\mu)(X_2), \cdots)^T.
\]

Denote it in short by \( m(\mu) = [m(\mu)(X)] \).

Write the transition matrix \( P(\mu) \) in the following canonical form [22, 30],
\[
P(\mu) = \begin{pmatrix} I & 0 \\ \ast & T(\mu) \end{pmatrix},
\]
where \( I \) is a unit matrix and \( 0 \) a zero matrix. The matrix \( T(\mu) \) denotes probability transition among transient states, whose entries are given by
\[
P(\mu)(X,Y), \quad X \in S_{\text{nop}}(\mu), Y \in S_{\text{nop}}(\mu).
\]
The part \( \ast \) plays no role in the analysis.

Since \( \forall X \in S_{\text{opt}}(\mu), m(\mu)(X) = 0 \), we only need to consider \( m(\mu)(X) \) for transient states \( X \in S_{\text{nop}}(\mu) \). For the simplicity of notations, we still let \( m(\mu) \) denote the vector whose entries are given by
\[
[m(\mu)(X)], X \in S_{\text{nop}}(\mu).
\]
Then we know that (Theorem 3.2 in [22])

Lemma 1. The hitting time vector satisfies
\[
m = (I - T(\mu))^{-1}1. \tag{11}
\]
Following Definition 3.2.2 in [21], we have

**Definition 3.** For an absorbing Markov chain, define its fundamental matrix to be

\[ N(\mu) := (I - T(\mu))^{-1}. \]

From Theorem 3.2.4. in [21], we have

**Lemma 2.** Denote \( N(\mu)(X,Y) \) to be an entry of the fundamental matrix, then \( N(\mu)(X,Y) \) is the same as the staying time of the absorbing Markov chain in the transitive state \( Y \) when starting from the transitive state \( X \).

From its name, we know the fundamental matrix plays a fundamental role in analyzing any absorbing Markov chain. Staying times are its entries and hitting times is the sum of its rows. Therefore spectral radius of the fundamental matrix can be used to determine the magnitude of staying times and hitting times.

From the above two lemmas, we see that the hitting time \( m(\mu)(X) \) equals the sum of staying times at all transient states:

\[ m(\mu)(X) = N(\mu)(X, S_{\text{nop}}^{(\mu)}). \]

The shorter the Markov chain stays in transient states, the shorter the Markov chain hits the absorbing set. These two times are equivalent but with meanings in opposite directions: one emphasizes on leaving and the other on staying.

Based on the above observation, we define a new index of measuring the performance of ARS: staying times in transient states. Because the staying time \( N(X,Y) \) depends on the starting state \( X \) and transient state \( Y \), so its value varies as \( X \) and \( Y \). Hence staying times cannot be directly employed in measuring population scalability, otherwise \( R(\mu) \) might take multiple values. Instead we use spectral radius\(^3\) of the fundamental matrix to measure the total staying time and this can guarantee a unique value of \( R(\mu) \).

**Definition 4.** The performance of an \((\mu + \mu)\)-ARS algorithm under spectral radius is defined by spectral radius of the fundamental matrix \( \rho(N(\mu)) \).

The performance of an ARS algorithm under spectral radius is between the best and worst case hitting times (its proof is given in Lemma 5), i.e.

\[ \min\{m(\mu)(X); X \in S_{\text{nop}}^{(\mu)}\} \leq \rho(N(\mu)) \leq \max\{m(\mu)(X); X \in S_{\text{nop}}^{(\mu)}\}. \]

\(^2\text{Notation used in the formula and thereafter is: given a subset } S_{\text{sub}}^{(\mu)} \subseteq S^{(\mu)}, \text{ denote } \)

\[ p(\mu)(X, S_{\text{sub}}^{(\mu)}) := \sum_{Y \in S_{\text{sub}}^{(\mu)}} p(\mu)(X, Y). \]

\(^3\text{The spectral radius of an } n \times n \text{ square matrix (say, } A), \text{ denoted by } \rho(A), \text{ is defined by } \)

\[ \rho(A) := \max\{|\lambda_i|; i = 1, \ldots, n\}, \text{ where } \lambda_1, \lambda_2, \ldots, \lambda_n \text{ are eigenvalues of matrix } A. \]
Remark: The role of spectral radius is similar to that of the average case hitting time; however there is a difference between them. In the appendix, we will give an example to display the performance of an ARS algorithm under spectral radius might equal either the best hitting times or the worst case hitting time.

We define the population scalability under spectral radius as follows.

**Definition 5.** Given a family of ARS algorithms with population sizes $\mu = 1, 2, \cdots$, population scalability of the $(\mu + \mu)$-ARS algorithm under spectral radius is defined by the ratio

$$R(\mu) := \frac{\rho(N^{(1)})}{\rho(N^{(\mu)})}. \quad (12)$$

As intuitively described in formulas (2) and (3), population scalability under hitting times and running times is different in a factor $1/\mu$. If population scalability under running times is greater than 1 for a population size $\mu > 1$, then population scalability under hitting times must be super linear. This leads to the following concept of super linear population scalability under spectral radius.

**Definition 6.** Given a family of ARS algorithms with population sizes $\mu = 1, 2, \cdots$, the $(\mu + \mu)$-ARS algorithm in the family is called to have super linear scalability under spectral radius if

$$R(\mu) > \mu. \quad (13)$$

### 3 Fundamental Matrix and Spectral Radius

#### 3.1 Spectral Radius in Abstract Population-based Random Search

Given fundamental matrix of the Markov chain associated with an ARS algorithm, its spectral radius is regarded as an index for measuring staying times of the Markov chain in transient states. In the following, we give more details to explain this intuitive meaning. First we discuss the case of $(1 + 1)$-ARS.

**Lemma 3.** Given the Markov chain associated with an $(1 + 1)$-ARS algorithm, and denote fundamental matrix of the Markov chain by

$$N^{(1)} := (I - T^{(1)})^{-1}.$$

1. Let $\lambda_T$ be an eigenvalue of $T^{(1)}$, and $\lambda_N$ an eigenvalue of $N^{(1)}$, then

$$(1 - \lambda_T)^{-1}$$

is an eigenvalue of $N^{(1)}$ and

$$\lambda_N = (1 - P^{(1)}(x, x))^{-1}, x \in S^{(1)}_{\text{nop}}.$$
2. Let $x_\rho$ be the state such that
\[
x_\rho := \text{arg max}\{N^{(1)}(x, x); x \in S_{\text{nop}}^{(1)}\},
\]
then
\[
\rho(N^{(1)}) = N^{(1)}(x_\rho, x_\rho) = 1/(1 - P^{(1)}(x_\rho, x_\rho)).
\]
and
\[
x_\rho = \text{arg max}\{1/(1 - P^{(1)}(x, x)); x \in S_{\text{nop}}^{(1)}\},
\]
\[
= \text{arg min}\{1 - P^{(1)}(x, x); x \in S_{\text{nop}}^{(1)}\}.
\]

Proof. (1) From $N^{(1)} := (I - T^{(1)})^{-1}$, we know that $(1 - \lambda_T)^{-1}$ is an eigenvalue of $N^{(1)}$.

Let’s arrange all individuals in $S_{\text{nop}}^{(1)}$ in the order of their fitness from high to low (where the individuals at the same fitness level may be arranged in any order), and write them in a vector form $(x_1, x_2, \ldots, x_\eta)^T$. Due to strict elitist selection, the transition probability satisfies: $\forall x, y$ with $f(y) \leq f(x)$
\[
P^{(1)}(x, y) = 0.
\]

Then the matrix $T^{(1)}$ is a lower triangular matrix, which can be written in the form:
\[
T^{(1)} = \begin{pmatrix}
P^{(1)}(x_1, x_1) & 0 & \cdots & 0 \\
P^{(1)}(x_2, x_1) & P^{(1)}(x_2, x_2) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
P^{(1)}(x_\eta, x_1) & P^{(1)}(x_\eta, x_2) & \cdots & P^{(1)}(x_\eta, x_\eta)
\end{pmatrix}.
\]

Thus $\forall x \in S_{\text{nop}}^{(1)}$, $P^{(1)}(x, x)$ is an eigenvalue of $T^{(1)}$ and then $\lambda_N = (1 - P^{(1)}(x, x))^{-1}$ is an eigenvalue of $N^{(1)}$.

(2) It can be drawn directly from the fact the matrix $N^{(1)}$ is a lower triangular matrix. $\square$

From the definition of $x_\rho$, we see $x_\rho$ is the transient state where the Markov chain stays there with the longest staying time. Therefore $\rho(N^{(1)})$ in $(1 + 1)$-ARS represents an index of measuring staying times of the Markov chain in transient states in the worst case.

Next we discuss the case of $(\mu + \mu)$-ARS (where $\mu > 1$).

**Lemma 4.** Given the Markov chain associated with an $(\mu + \mu)$-ARS algorithm, and denote the fundamental matrix by
\[
N^{(\mu)} := (I - T^{(\mu)})^{-1}.
\]
\[
\forall x \in S_{\text{nop}}^{(\mu)}, \text{ denote } T^{(\mu)}_{x,x} \text{ to be the sub-matrix of } T^{(\mu)} \text{ whose entries are given by }
\]
\[
P^{(\mu)}(X, Y)
\]
where the super individuals in both populations $X$ and $Y$ take the value of $x$. 11
1. Let $\lambda_T$ be an eigenvalue of the sub-matrix $T_{x,x}^{(\mu)}$, then $(1 - \lambda_T)^{-1}$ is an eigenvalue of $N^{(\mu)}$. And all eigenvalues of the sub-matrix $T_{x,x}^{(\mu)}$ over $x \in S_{\text{nop}}^{(1)}$ consist the eigenvalue set of $N^{(\mu)}$.

2. Let $N_{x,x}^{(\mu)} := (1 - T_{x,x}^{(\mu)})^{-1}$ and $\lambda(N_{x,x}^{(\mu)})$ be the maximum real eigenvalue of $N_{x,x}^{(\mu)}$, then

$$
\rho(N^{(\mu)}) = \max\{\rho(N_{x,x}^{(\mu)}); x \in S_{\text{nop}}^{(1)}\} = \max\{\lambda(N_{x,x}^{(\mu)}); x \in S_{\text{nop}}^{(1)}\}.
$$

**Proof.** (1) Let’s arrange all populations in $S_{\text{nop}}^{(\mu)}$ in the order of the fitness of their super individual from high to low (where populations with the same super-individual are arranged together), and write them in a vector form:

$$(X_1, X_2, \ldots, X_{\eta(\mu)})^T.$$

Due to strict elitist selection, the transition probability satisfies:

$$P^{(\mu)}(X, Y) = 0, \quad \forall X, Y \in S_{\text{nop}}^{(\mu)} \text{ with } f(Y) \leq f(X),$$

where $X$ and $Y$ are the super individuals in $X$ and $Y$ respectively. Hence the matrix $T^{(\mu)}$ is a block lower triangular matrix, which can be written in the form:

$$T^{(\mu)} = \begin{pmatrix}
T_{x_1,x_1}^{(\mu)} & 0 & \cdots & 0 \\
T_{x_2,x_1}^{(\mu)} & T_{x_2,x_2}^{(\mu)} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
T_{x_{\eta},x_1}^{(\mu)} & T_{x_{\eta},x_2}^{(\mu)} & \cdots & T_{x_{\eta},x_{\eta}}^{(\mu)}
\end{pmatrix}, \quad (15)$$

where $0$ is a zero sub-matrix. Given $x \in S_{\text{nop}}^{(1)}$ and $y \in S_{\text{nop}}^{(1)}$, $T_{x,y}^{(\mu)}$ a sub-matrix whose entries are given by

$$P^{(\mu)}(X, Y), \quad \forall X \in S_{\text{nop}}^{(\mu)} : \overline{X} = x \text{ and } Y \in S_{\text{nop}}^{(\mu)} : \overline{Y} = y,$$

where $X$ and $Y$ are super individuals in populations $X$ and $Y$ respectively.

Let $\lambda_T$ be an eigenvalue of the matrix $T^{(\mu)}$. Since $T^{(\mu)}$ is a block lower triangular matrix, then $\lambda_T$ must be an eigenvalue of a sub-matrix $T_{x,x}^{(\mu)}$. Let $\lambda_N$ be an eigenvalue of the matrix $N^{(\mu)}$. From $N^{(\mu)} := (I - T^{(\mu)})^{-1}$, we know $\lambda_N = (1 - \lambda_T)^{-1}$ is an eigenvalue of $I - T^{(\mu)}$, where $\lambda_T$ is an eigenvalue of a sub-matrix $T_{x,x}^{(\mu)}$.

(2) The first formula $\rho(N^{(\mu)}) = \max\{\rho(N_{x,x}^{(\mu)}); x \in S_{\text{nop}}^{(1)}\}$ can be drawn from the conclusion above. Because $T^{(\mu)}$ is a block lower triangular matrix, so $N^{(\mu)}$ is a block lower triangular matrix too.

The second formula $\rho(N^{(\mu)}) = \max\{\lambda(N_{x,x}^{(\mu)}); x \in S_{\text{nop}}^{(1)}\}$ can be drawn from Perron and Frobenius’ theorem. Since $N_{x,x}^{(\mu)}$ is a nonnegative matrix, its spectral radius is one of its eigenvalues. \qed
\textbf{Definition 7.} \( \forall x \in S^{(1)}_{\text{nop}}, \) denote the set of transient states

\[ S^{(\mu)}_{x} := \{ Y; \bar{y} = x \}, \]

where \( \bar{y} \) is the super individual of \( Y \).

Then according to Lemma 4, the spectral radius \( \rho(N^{(\mu)}) \) may be interpreted as an index of measuring the Markov chain to stay in a transient states set \( S^{(\mu)}_{x} \) with the longest staying time or in the worst case.

Summarizing the above two lemmas, we conclude that the performance of ARS under spectral radius can be viewed as an index of measuring how long an ARS algorithm spends in a transient states set \( S^{(\mu)}_{x} \) with the longest staying time. This set may be viewed as the most difficult area for the ARS algorithm to search. The performance of ARS under spectral radius is different from the existing performance measure, e.g. the best case hitting time (to measure the performance from the best transient states), the worst case hitting time (to measure the performance from the worst transient states), and the average hitting time (an arithmetic mean of all hitting times). In the appendix, we will provide an example of explaining the performance of ARS under spectral radius.

Finally let’s give a short discussion of average staying times in transient states. Let \( \lambda_i \) \( (i = 1, 2, \cdots, \eta(\mu)) \) be the set of eigenvalues of \( N^{(\mu)} \), then the spectral radius of the fundamental matrix is the maximum value of eigenvalues.

\[ \rho(N^{(\mu)}) = \max\{|\lambda_i|; i = 1, \cdots, \eta(\mu)\}. \]

Let’s investigate the average value of eigenvalues

\[ \frac{\sum_{i=1}^{\eta(\mu)} \lambda_i}{\eta(\mu)}. \]

From the relationship between the trace and eigenvalues of a matrix, we know that

\[ \frac{\sum_{i=1}^{\eta(\mu)} \lambda_i}{\eta(\mu)} = \frac{\text{trace}(N^{(\mu)})}{\eta(\mu)} = \frac{\sum_{X \in S^{(\mu)}_{\text{nop}}} N^{(\mu)}(X, X)}{\eta(\mu)}. \]

Thus \( \frac{\text{trace}(N^{(\mu)})}{\eta(\mu)} \) is the average staying time for the Markov chain in transient states.

\subsection{3.2 Lower and Upper Bounds on Spectral Radius}

Given a fundamental matrix associated with an ARS algorithm, its spectral radius can be bounded by hitting times.

\textbf{Lemma 5.} Spectral radius of the fundamental matrix associated with an \((\mu + \mu)\)-ARS algorithm is between the best and worst case performances, i.e.

\[ \min\{m^{(\mu)}(X); X \in S^{(\mu)}_{\text{nop}}\} \leq \rho(N^{(\mu)}) \leq \max\{m^{(\mu)}(X); X \in S^{(\mu)}_{\text{nop}}\}. \]
Proof. In the proof, we drop the mark ($\mu$) without causing confusion.

(1) The proof of $\rho(N) \leq \max\{m^{(\mu)}(X); X \in S^{(\mu)}_{\text{nop}}\}$.
It is directly derived from the following inequality:

$$\rho(N) \leq \|N\|_{\infty} = \max\{m^{(\mu)}(X); X \in S^{(\mu)}_{\text{nop}}\}.$$ 

(2) The proof of $\min\{m(X); X \in S^{(\mu)}_{\text{nop}}\} \leq \rho(N)$.
Let

$$m := \min\{m^{(\mu)}(X); X \in S^{(\mu)}_{\text{nop}}\}.$$ 

From Lemma [1] we have:

$$N1 = m \geq m1,$$

then we get

$$\frac{1}{m}N1 \geq 1.$$ 

Let $Q := N/m$. Because $Q \geq 0$, so we get $\forall k \geq 1$

$$Q^k1 \geq 1,$$

and then

$$\|Q^k\|_{\infty} \geq 1.$$ 

From Gelfand’s spectral radius formula

$$\rho(Q) = \lim_{k \to +\infty} \|Q^k\|_{\infty}^{1/k} \geq 1,$$

and $Q := N/m$, we get $\rho(N) \geq m$. \ \Box

Now we generalize the above lemma through Collatz’s method. Collatz’s method is described in the following lemma (Theorems 1 and 2 in [32]).

Lemma 6. Suppose that $A$ is an irreducible non-negative matrix. Let

$$a_k := A^k1, \quad k = 0, 1, 2, \ldots,$$

denote the vector $a_k := [a_k(i)]$ and

$$\lambda_k = \min_i \left\{ \frac{a_{k+1}(i)}{a_k(i)} \right\},$$

$$\overline{\lambda}_k = \max_i \left\{ \frac{a_{k+1}(i)}{a_k(i)} \right\}.$$

Then

$$\lambda_0 \leq \lambda_1 \leq \cdots \leq \rho(A) \leq \cdots \leq \overline{\lambda}_1 \leq \overline{\lambda}_0.$$

Furthermore if $A$ is a primitive matrix, then

$$\rho(A) = \lim_{k \to +\infty} \lambda_k = \lim_{k \to +\infty} \overline{\lambda}_k.$$ 

\footnote{Notation used in the formula and thereafter is: given two matrices $A = [a_{ij}]$ and $B = [b_{ij}]$, if for all $i, j$, $a_{ij} \geq b_{ij}$, we shall write $A \geq B$. Given two vectors $a = [a_i]$ and $b = [b_i]$, if for all $i$, $a_i \geq b_i$, we shall write $a \geq b.$}
Now we apply the above lemma into the fundamental matrix of the Markov chain associated with ARS.

Definition 8. \( \forall x \in S_{nop}^{(1)} \), define its higher fitness level set to be

\[
S_x^{(1)}(f_\succ) := \{ y; f(y) > f(x) \}, \\
S_x^{(\mu)}(f_\succ) := \{ Y; f(Y) > f(x) \},
\]

where \( Y \) is the super individual of the population \( Y \).

The set \( S_x^{(\mu)}(f_\succ) \) consists of all states at a higher fitness level than \( x \).

Definition 9. Given the sets \( S_x^{(\mu)} \) and \( S_x^{(\mu)}(f_\succ) \), denote

\[
\tau_x(X) := \min \{ t; \Phi_t \in S_x^{(\mu)}(f_\succ) \mid \Phi_0 = X \}, \\
m_x^{(\mu)}(X) := E[\tau_x(X)],
\]

which is the hitting time \( m_x^{(\mu)}(X) \) of the Markov chain \( \{ \Phi_t; t = 0, 1, 2, \cdots \} \) to the set \( S_x^{(\mu)}(f_\succ) \) when the Markov chain starts from \( X \in S_x^{(\mu)} \).

Let \( N_{x,x}^{(\mu)} := (I - T_{x,x}^{(\mu)})^{-1} \) be the fundamental matrix of the Markov Chain restricted in the set \( S_x^{(\mu)} \) (where \( T_{x,x}^{(\mu)} \) is given in Lemma [4]). The entry \( N_{x,x}^{(\mu)}(X,Y) \) of \( N_{x,x}^{(\mu)} \) represents the staying time of the Markov chain in the transient state \( Y \) when staring from the transient state \( X \).

Let \( m_x^{(\mu)} := N_x^{(\mu)} 1 \). Then the entry \( m_x^{(\mu)}(X) \) of \( m_x^{(\mu)} \) is the hitting time to the set \( S_x^{(\mu)}(f_\succ) \) when the Markov chain starts from \( X \in S_x^{(\mu)} \). And

\[
m_x^{(\mu)}(X) = \sum_{Y \in S_x^{(\mu)}} N_{x,x}^{(\mu)}(X,Y).
\]

Without losing generality, suppose that \( N_{x,x}^{(\mu)} \) is an irreducible matrix. Otherwise, its sub-matrix is taken into consideration and similar results to the following theorem will hold.

Lemma 7. Suppose that \( N_{x,x}^{(\mu)} \) is an irreducible matrix. Let

\[
q_k := (N_{x,x}^{(\mu)})^k 1, \quad k = 0, 1, 2, \cdots,
\]

denote the vector \( q := [q_k(X)] \) (where \( X \in S_x^{(\mu)} \)) and let

\[
\lambda_k = \min \left\{ \frac{q_{k+1}(X)}{q_k(X)}, X \in S_x^{(\mu)} \right\}, \\
\overline{\lambda_k} = \max \left\{ \frac{q_{k+1}(X)}{q_k(X)}, X \in S_x^{(\mu)} \right\}.
\]

Then

\[
\lambda_0 \leq \lambda_1 \leq \cdots \leq \rho(N_{x,x}^{(\mu)}) \leq \cdots \leq \overline{\lambda_1} \leq \overline{\lambda_0}.
\]
Furthermore if $N^{(\mu)}_{x,x}$ is a primitive matrix, then
\[ \rho(N^{(\mu)}_{x,x}) = \lim_{k \to +\infty} \lambda_k = \lim_{k \to +\infty} \tilde{\lambda}_k. \]

**Proof.** It is a direct corollary of Lemma 6. \qed

Now we discuss the meaning of Inequality (20) further. Let’s investigate the cases of $k = 0, 1$.

When $k = 0$, Inequality (20) becomes
\[ \min\{m^{(\mu)}_x(X); X \in S^{(\mu)}_{x,x}\} \leq \rho(N^{(\mu)}_{x,x}) \leq \max\{m^{(\mu)}_x(X); X \in S^{(\mu)}_{x,x}\}, \]
which is an extension of Lemma 5.

When $k = 1$, Inequality (20) becomes
\[
\begin{align*}
\rho(N^{(\mu)}_{x,x}) &\geq \min \left\{ \frac{\sum_{Y \in S^{(\mu)}_{x,x}} N^{(\mu)}_{x,x}(X,Y)m^{(\mu)}_x(Y)}{m^{(\mu)}_x(X)}; X \in S^{(\mu)}_{x,x} \right\}, \\
\rho(N^{(\mu)}_{x,x}) &\leq \max \left\{ \frac{\sum_{Y \in S^{(\mu)}_{x,x}} N^{(\mu)}_{x,x}(X,Y)m^{(\mu)}_x(Y)}{m^{(\mu)}_x(X)}; X \in S^{(\mu)}_{x,x} \right\}.
\end{align*}
\]

The upper and lower bounds on spectral radius at $k = 0$ are tighter than those at $k = 0$.

### 3.3 Fundamental Matrix and Drift Analysis

Given the Markov chain associated with an ARS algorithm, its fundamental matrix has a direct link to drift analysis [31].

**Definition 10.** \(\forall X \in S^{(\mu)}\), a function \(d(X)\) is called distance function if it satisfies the following condition
\[ d(X) = \begin{cases} 0 & \text{if } X \in S^{(\mu)}_{\text{opt}}, \\ \geq 0 & \text{if } X \in S^{(\mu)}_{\text{nop}}. \end{cases} \]

\(\forall X \in S^{(\mu)}_{\text{nop}},\) the mean one-step drift towards set \(S^{(\mu)}_{\text{opt}}\) is defined by
\[ \Delta d(X) := \sum_{Y \in S^{(\mu)}} (d(X) - d(Y))P^{(\mu)}(X,Y). \]

Let’s arrange all populations in \(S^{(\mu)}_{\text{nop}}\) in the order of the fitness of their super individuals from high to low, and write them in a vector form:
\[ (X_1, X_2, \ldots, X_{\eta(\mu)})^T. \]

And let \(d^{(\mu)} = [d(X)]\) be the distance vector where \(X \in S^{(\mu)}_{\text{nop}}.\) Using the fundamental matrix, we can easily draw fundamental drift lemmas for estimating upper and lower bounds respectively (Theorems 2 and 3 in [30]).
Lemma 8. Let $d^{(\mu)} = [d(X)]$. If $(I - T^{(\mu)})d^{(\mu)} \leq 1$, then $m^{(\mu)} \geq d^{(\mu)}$.

Lemma 9. Let $d^{(\mu)} = [d(X)]$. If $(I - T^{(\mu)})d^{(\mu)} \geq 1$, then $m^{(\mu)} \leq d^{(\mu)}$.

In the above two lemmas, $\leq$ (and $\geq$) can be replaced by $<$ (and $>$) respectively.

From Lemma 1, we get

Lemma 10. Let $d^{(\mu)} = [m(X)]$, then

$$\Delta d^{(X)} = 1.$$ 

As a simple application of drift analysis, the following general lower and upper bounds on hitting times can be obtained.

Lemma 11. $\forall x \in S_{\text{nop}}^{(\mu)}$, let $N^{(\mu)}_{x,x} := (I - T^{(\mu)})^{-1}$ be a sub-matrix of the fundamental matrix of the Markov chain. Denote $m^{(\mu)} := N^{(\mu)}_{x,x} 1$. Then

$$\min \left\{ \frac{1}{P^{(\mu)}(X, S^{(\mu)}(f_\mu))}; X \in S^{(\mu)}(x) \right\} \leq m^{(\mu)}(X) \leq \max \left\{ \frac{1}{P^{(\mu)}(X, S^{(\mu)}(f_\mu))}; X \in S^{(\mu)}(x) \right\}.$$ 

Proof. (1) First we prove the second inequality. Denote

$$\overline{m}_x := \max \left\{ \frac{1}{P^{(\mu)}(X, S^{(\mu)}(f_\mu))}; X \in S^{(\mu)}(x) \right\}.$$ 

Define a distance function $d(X)$ to be

$$d(X) = \begin{cases} 0 & \text{if } X \in S^{(\mu)}(f_\mu), \\ \overline{m}_x & \text{if } X \in S^{(\mu)}. \end{cases}$$

Then we have

$$\Delta d(X) = P^{(\mu)}(X, S^{(\mu)}(f_\mu)) \overline{m}_x \geq 1.$$ 

From Lemma 8, we get $m^{(\mu)}(X) \leq \overline{m}_x$.

(2) We can prove the first inequality in a similar way. Let

$$\underline{m}_x := \min \left\{ \frac{1}{P^{(\mu)}(X, S^{(\mu)}(f_\mu))}; X \in S^{(\mu)}(x) \right\}.$$ 

Define a distance function $d(X)$ to be

$$d(X) = \begin{cases} 0 & \text{if } X \in S^{(\mu)}(f_\mu), \\ \underline{m}_x & \text{if } X \notin S^{(\mu)}. \end{cases}$$

Then we have

$$\Delta d(X) = P^{(\mu)}(X, S^{(\mu)}(f_\mu)) \underline{m}_x \leq 1.$$ 

From Lemma 9, we get $m^{(\mu)}(X) \geq \underline{m}_x$. $\square$
3.4 Time-Based Fitness Landscape

The space $S$ given in Section 2 is only required to be a finite set. No topological structure, such as neighborhood or distance, has been assigned to the set $S$. In order to study population scalability, a new type of fitness landscape, called time-based fitness landscape, is introduced in the paper. Because only $(1+1)$-ARS algorithms are analyzed in this section, no superscript $(\mu)$ is used below for simplicity.

**Definition 11.** Given a fitness function $f(x)$ and an $(1+1)$-ARS algorithm, its associated time-based fitness landscape is defined by the set of pairs

$$\{(m(x), f(x)); x \in S\},$$

where $m(x)$ is the hitting time of the $(1+1)$-ARS algorithm starting from the state $x$ and $f(x)$ its fitness.

The above fitness landscape is determined by two factors: fitness function and hitting time. A $(1+1)$-ARS algorithm is chosen here because it is the benchmark for comparing all other $(\mu+\mu)$-ARS ($\mu = 2, 3, \ldots$) algorithms. Hitting times play the role of “distance” on time-based fitness landscapes.

If search and selection operators in the $(1+1)$-ARS algorithm change, then the related time-based fitness landscape will change. The introduction of time-based fitness landscapes is purely for the convenience of understanding population scalability theoretically, because its definition depends on hitting times of the $(1+1)$-ARS algorithm, usually unknown.

Time-based fitness landscapes are classified into two categories:

**Monotonic landscape:** $\forall x \in S_{\text{nop}}, y \in S_{\text{nop}},$

$$m(x) < m(y) \iff f(x) > f(y).$$

**Non-monotonic landscape:** $\exists x \in S_{\text{nop}}, y \in S_{\text{nop}},$

$$m(x) < m(y) \implies f(x) \leq f(y).$$

**Example 3.** Consider pseudo-Boolean optimization. Given the $(1+1)$ EA using bitwise mutation with a mutation rate $1/n$ and strict elitist selection for solving the following One-Max function: let $x = (b_1 \cdots b_n)$ be a binary string with the length $n$,

$$f(x) = \sum_{i=1}^{n} b_i,$$

then the related time-based fitness landscape is monotonic.

An important non-monotonic fitness landscape is the deceptive fitness landscape, defined as follows:
Deceptive landscape: $\forall x \in S_{\text{nop}}, y \in S_{\text{nop}},$
\[
m(x) < m(y) \iff f(x) < f(y).
\]

**Example 4.** Consider pseudo-Boolean optimization. Given the (1+1) EA using bitwise mutation with a mutation rate $1/n$ and elitist selection for maximizing the following Fully-Deceptive function: let $x = (b_1 \cdots b_n)$ be a binary string with the length $n,$
\[
f(x) = \begin{cases} 
n + 1 & \text{if } \sum_{i=1}^{n} b_i = 0; \\
k & \text{if } \sum_{i=1}^{n} b_i = k, k = 1, \cdots, n.
\end{cases}
\]
then the related time-based fitness landscape is non-monotonic.

Next we introduce the concept of fitness gap on the time-based fitness landscapes.

**Definition 12.** $\forall x \in S_{\text{nop}},$ define its gap between its current fitness level and higher fitness level to be the staying time $N(x, x)$ of the Markov chain in the transient state $x.$

Finally we discuss the regularity of time-based fitness landscapes. In the definition of global search, a search operator is only required to meet two essential conditions:
\begin{enumerate}
\item $\forall x \in S, y \in S : p_m(x, y) > 0,$
\item $\sum_{y \in S} p_m(x, y) = 1.$
\end{enumerate}

Using such kinds of global search sometimes will lead to irregular fitness landscapes. For example, for different states, we can design completely different probability distributions, then the related time-based fitness landscape will be very irregular and extremely difficult to analyze. Hence we introduce regular fitness landscapes which have a good property.

**Definition 13.** Given a time-based fitness landscape. $\forall x \in S_{\text{nop}},$ denote
\[
S_x(m_<) := \{y ; m(y) < m(x)\}.
\]

The set consists of all states which are closer to the optimal set than $x$ on the time-based fitness landscape.

**Definition 14.** Given a time-based fitness landscape $(m, f),$ it is called regular if $\forall x, y$ such that $m(x) \leq m(y),$ it holds
\[
P(x, S_x(m_<)) \geq P(y, S_x(m_<)).
\]

The meaning of “regularity” is that: given a state $x,$ if another state $y$ is farther away than the state $x$ (i.e. $m(y) \geq m(x)$), then when starting at the state $y,$ the probability for the Markov chain to move towards the set $S_x(m_<)$ is smaller than that when staring at the state $x.$

**Example 5.** Consider pseudo-Boolean optimization. Given the (1+1) EA using bitwise mutation with a mutation rate $1/n$ and strict elitist selection for solving the One-Max function and Fully-Deceptive function, then the related time-based fitness landscapes are regular.
4 Analysis of Population Scalability under Spectral Radius

Using fundamental matrix associated with ARS, now we can analyze population scalability of ARS under spectral radius and draw several general results.

4.1 Using Population can Increase Population Scalability

Using population will reduce the staying times of the associated Markov chain in transient states and thus can increase population scalability. The following theorem prove this claim.

Theorem 1. Given a family of \((\mu + \mu)\)-ARS algorithms (where \(\mu = 1, 2, \cdots\)) for maximizing the function \(f\). Then \(\forall \mu \geq 2\), it holds:

\[ R(\mu) > 1. \]

Proof. For the \((1 + 1)\)-ARS algorithm, from Lemma \(3\) we know there exists an \(x_\rho \in S^{(1)}_{\text{nop}}\) such that

\[
x_\rho := \arg \max \{N^{(1)}(x, x); x \in S^{(1)}_{\text{nop}}\}
\]

\[
= \arg \max \{1/(1 - P^{(1)}(x, x)); x \in S^{(1)}_{\text{nop}}\}
\]

\[
= \arg \min \{1 - P^{(1)}(x, x); x \in S^{(1)}_{\text{nop}}\},
\]

and

\[
\rho(N^{(1)}) = 1/(1 - P^{(1)}(x_\rho, x_\rho)).
\]

For the \((\mu + \mu)\)-ARS algorithm where \(\mu = 2, 3, \cdots\), from Lemma \(4\) we know that the eigenvalues of the matrix \(N^{(\mu)}\) are the same as those of sub-matrices \((I - T^{(\mu)}_{x, x})^{-1}\) over all \(x \in S^{(1)}_{\text{nop}}\).

Let \(\lambda^{(\mu)}\) be an eigenvalue of a sub-matrix \(I - T^{(\mu)}_{x, x}\), then from Gershgorin’s Theorem, we know that let \(X = (x, \cdots, x)\), then it holds:

\[
| \lambda^{(\mu)} - (1 - P^{(\mu)}(X, X)) | \leq P^{(\mu)}(X, S^{(\mu)}_x \setminus \{X\}),
\]

and then

\[
| \lambda^{(\mu)} | \geq 1 - P^{(\mu)}(X, X) - P^{(\mu)}(X, S^{(\mu)}_x \setminus \{X\})
\]

\[
= 1 - P^{(\mu)}(X, S^{(\mu)}_x)
\]

\[
= P^{(\mu)}(X, S^{(\mu)}_x(f_x)).
\]

From the above inequality, we know there exists a state \(y \in S^{(1)}_{\text{nop}}\) and a population \(Y = (y, \cdots, y)\) satisfying

\[
\min\{| \lambda^{(\mu)} |; \lambda^{(\mu)} \text{ is an eigenvalue of } I - T^{(\mu)}\} \geq P^{(\mu)}(Y, S^{(\mu)}_y(f_y)).
\]

20
Due to global search, \( \forall \mu > 1 \), we have
\[
P^{(\mu)}(y, S^{(\mu)}_y(f_\succ)) > P^{(1)}(y, S^{(1)}_y(f_\succ)).
\] (29)

Because
\[
P^{(1)}(y, S^{(1)}_y(f_\succ)) = 1 - P^{(1)}(y, y)
\geq 1 - P^{(1)}(x_\mu, x_\rho).
\]

Hence
\[
\min\{| \lambda^{(\mu)} ; \lambda^{(\mu)} \text{ is an eigenvalue of } I - T^{(\mu)} \} > 1/\rho(N^{(1)}),
\]
From \( N^{(\mu)} := (I - T^{(\mu)})^{-1} \), we have
\[
\rho(N^{(\mu)}) < \rho(N^{(1)}).
\]
That is \( R(\mu) > 1 \). \( \square \)

An open question is whether \( R(\mu) \) is a monotonically decreasing function of \( \mu \)? In general it does not hold because selection strategies may be variant when \( \mu \geq 2 \). So the question will be asked more precisely as under what kind of conditions \( R(\mu) \) is a monotonically decreasing function of \( \mu \)?

### 4.2 No Diversity, No Super Linear Population Scalability

Here the meaning of “no diversity” refers to elitist selection with the highest pressure. Using “no diversity” selection is often regarded as a bad choice for designing RS. This can be confirmed from the following theorem.

**Theorem 2.** Given a family of \((\mu + \mu)\)-ARS algorithms (where \( \mu = 1, 2, \cdots \)) using strict elitist selection with the highest pressure for maximizing the function \( f \). Then for \( \mu = 2, 3, \cdots \), it holds:

\[ R(\mu) < \mu. \]

**Proof.** For the \((1 + 1)\)-ARS algorithm, from Lemma 3 we know there exists an \( x_\rho \in S^{(1)}_{\text{nop}} \) such that
\[
x_\rho := \arg\max\{N^{(1)}(x, x); x \in S^{(1)}_{\text{nop}}\}
= \arg\max\{1/(1 - P^{(1)}(x, x)); x \in S^{(1)}_{\text{nop}}\}
= \arg\min\{1 - P^{(1)}(x, x); x \in S^{(1)}_{\text{nop}}\},
\]
and
\[
\rho(N^{(1)}) = 1/(1 - P^{(1)}(x_\rho, x_\rho)).
\]
Due to global search, we know that
\[
\rho(N^{(1)}) = 1/(1 - P^{(1)}(x_\rho, x_\rho)) > 1.
\]
For the \((\mu + \mu)\)-ARS algorithm with \(\mu = 2, 3, \ldots\), from Lemma 4 we know that the eigenvalues of the matrix \(N^{(\mu)}\) are the same as those of sub-matrices \((I - T_{x,x}^{(\mu)})^{-1}\) where \(x \in S_{\text{nop}}^{(1)}\).

\(\forall x \in S_{\text{nop}}^{(1)}\), let \(\lambda^{(\mu)}\) be an eigenvalue of a sub-matrix \(I - T_{x,x}^{(\mu)}\).

Because the ARS algorithm uses elitist selection with the highest pressure, the sub-matrix \(I - T_{x,x}^{(\mu)}\) is a lower triangular matrix. Its eigenvalues satisfy:

- Either \(\lambda^{(\mu)} = 1\).
  
  In this case it holds
  \[ \lambda^{(\mu)} < 1/\rho(N^{(1)}). \] (30)

- Or there exists an \(X = (x, \ldots, x)\) such that
  \[ \lambda^{(\mu)} = 1 - P^{(\mu)}(X, X) \]
  \[ = P^{(\mu)}(X, S^{(\mu)}_x(f_>)). \]

Let’s analyze the second case and prove that \(\lambda^{(\mu)} < 1/\rho(N^{(1)})\) too.

Let \(X_\rho = (x_\rho, \ldots, x_\rho)\) and then there exists an eigenvalue of \(I - T_{x_\rho,x_\rho}^{(\mu)}\) such that

\[ \lambda^{(\mu)} = P^{(\mu)}(X_\rho, S^{(\mu)}_{x_\rho}(f_>)) \]
\[ = 1 - (1 - P^{(1)}(x_\rho, S^{(1)}_{x_\rho}(f_>)))^\mu \]
\[ < \mu P^{(1)}(x_\rho, S^{(1)}_{x_\rho}(f_>) \]
\[ = \mu(1 - P^{(1)}(x_\rho, x_\rho)) \]
\[ = \mu/\rho(N^{(1)}). \] (31)

Hence from Inequalities (30) and (31), we have

\[ \min\{ |\lambda^{(\mu)}| : \lambda^{(\mu)} \text{ is an eigenvalue of } I - T^{(\mu)} \} < 1/\rho(N^{(1)}). \]

It follows that from \(N^{(\mu)} := (I - T^{(\mu)})^{-1}\)

\[ \mu \rho(N^{(\mu)}) > \rho(N^{(1)}). \]

In other words, \(R(\mu) < \mu. \)

\[ \square \]

**Example 6.** Consider pseudo-Boolean optimization. Given the \((\mu + \mu)\) EA using bitwise mutation with a mutation rate \(1/n\) and strict elitist selection with the highest pressure for solving any function, then there is no super linear scalability.

\[ ^5 \text{A } (\mu + \mu) \text{ EA using strict elitist selection with the highest pressure is equivalent to a } (1 + \lambda) \text{ EA (where } \mu \text{ is replaced by } \lambda). \]
4.3 Super Linear Population Scalability is Impossible on Regular Monotonic Fitness Landscapes

Easy fitness functions such as the One-Max function are so easy to solve in some sense that using population is unnecessary. We prove super linear scalability is impossible on regular monotonic time-based fitness landscapes for any ARS algorithm.

**Theorem 3.** Given a family of \((\mu + \mu)-ARS\) algorithms (where \(\mu = 1, 2, \cdots\)) for maximizing the function \((6)\). Suppose that the time-based fitness landscape associated with the \((1 + 1)\)-ARS is regular monotonic. Then for \(\mu = 2, 3, \cdots\), it holds:

\[
R(\mu) < \mu.
\]

**Proof.** For the \((1 + 1)\)-ARS algorithm, from Lemma 3, we know there exists an \(x_{\rho} \in S_{\text{nop}}^{(1)}\) such that

\[
x_{\rho} := \arg \max \{ N^{(1)}(x, x); x \in S_{\text{nop}}^{(1)} \}
\]

\[
= \arg \max \{1/(1 - P^{(1)}(x, x)); x \in S_{\text{nop}}^{(1)} \}
\]

and

\[
\rho(N^{(1)}) = 1/(1 - P^{(1)}(x_{\rho}, x_{\rho})).
\]

For the \((\mu + \mu)\)-ARS algorithm, let’s consider the sub-matrix \(T^{(\mu)}_{x,x}\) where \(x \in S_{\text{nop}}^{(1)}\) and let

\[
N^{(\mu)}_{x,x} := (I - T^{(\mu)}_{x,x})^{-1}.
\]

Without losing generality suppose that \(T^{(\mu)}_{x,x}\) is irreducible, and then \(N^{(\mu)}_{x,x}\) is irreducible too.

From Inequality (21), we know that

\[
\rho(N^{(\mu)}_{x,x}) \geq \min\{m^{(\mu)}_{x}(X); X \in S_{x}^{(\mu)}\}. \tag{32}
\]

From Lemma 11 we know that

\[
\min\{m^{(\mu)}_{x}(X); X \in S_{x}^{(\mu)}\} \geq \min\left\{ \frac{1}{P^{(\mu)}(X, S_{x}^{(\mu)}(f_{>}))}; X \in S_{x}^{(\mu)} \right\}. \tag{33}
\]

\[
\forall X = (x_1, \cdots, x_\mu) \text{ such that } x_1 = x_{\rho} \text{ and } f(x_1) \geq f(x_2) \geq \cdots \geq f(x_\mu), \text{ because the fitness landscape is monotonic, then we have }
\]

\[
m^{(1)}(x_1) \leq m^{(1)}(x_2) \leq \cdots \leq m^{(1)}(x_\mu).
\]

Let \(X_\rho := (x_\rho, \cdots, x_\rho)\). Because the fitness landscape is regular, it holds \(\forall \mu > 1\)

\[
P^{(\mu)}(X, S_{x_\rho}^{(\mu)}(f_{>})) \leq P^{(\mu)}(X_\rho, S_{x_\rho}^{(\mu)}(f_{>})) < \mu(1 - P^{(1)}(x_{\rho}, x_{\rho})), \tag{34}
\]
Then from Inequalities (32), (33) and (34), we get
\[
\rho(N_{x,\rho}) > \frac{1}{\mu(1 - P_{x,\rho})} = \frac{1}{\mu} \rho(N^{(1)}),
\]
and then
\[
\rho(N^{(\mu)}) > \frac{1}{\mu} \rho(N^{(1)}).
\]

So we finish the proof of $R(\mu) < \mu$. \hfill \qed

**Example 7.** Consider pseudo-Boolean optimization. Given the $(\mu + \mu)$ EA using bitwise mutation with a mutation rate $1/n$ and strict elitist selection for solving the One-Max function, then there is no super linear scalability.

### 4.4 Bridgeable Point and Diversity Preservation: Necessity of Super Linear Population Scalability

In the previous two subsections, we have proven two points:

1. Super linear scalability cannot happen on regular monotonic fitness landscapes.
2. ARS without population diversity cannot achieve super linear scalability.

Based on the above observation, we propose two necessary conditions for super linear scalability: “bridgeable point” and “diversity preservation”. In the following we give a detailed description about these two conditions.

**Definition 15.** Given a state $x \in S_{\text{nop}}^{(1)}$, a state $y$ is called a bridgeable point of $x$ if $y$ satisfies two conditions:

1. $f(y) \leq f(x)$;
2. $P^{(1)}(y, S_x^{(1)}(f_>) > P^{(1)}(x, S_x^{(1)}(f_>)$.

The name of “bridgeable point” comes from an intuitive meaning: given $x \in S_{\text{nop}}^{(1)}$, there is a gap between its current fitness level and higher fitness level on the time-based fitness landscape. The Markov chain associated with the $(1 + 1)$-ARS algorithm jumps directly over the gap towards the higher fitness level set $S_x^{(1)}(f_>)$. But in order to achieve super linear population scalability, it is necessary for the Markov chain associated with the $(\mu + \mu)$-ARS algorithm (where $\mu > 1$) to pass through at least one of bridgeable points.

**Theorem 4.** Given a family of $(\mu + \mu)$-ARS algorithms (where $\mu = 1, 2, \cdots$) for maximizing the function $f$. If for some population size $\mu > 1$, super linear scalability happens, then $\forall \rho$ such that

\[
x_\rho := \arg \max\{N^{(1)}(x, x); x \in S_{\text{nop}}^{(1)}\},
\]

there exists a bridgeable point of $x_\rho$. 24
Proof. For the \((1 + 1)\)-ARS algorithm, from Lemma 3, we know \(x_\rho \in S_{\text{nop}}^{(1)}\) satisfies

\[
x_\rho := \arg \max \{ N^{(1)}(x, x); x \in S_{\text{nop}}^{(1)} \} = \arg \max \{ 1/ (1 - P^{(1)}(x, x)); x \in S_{\text{nop}}^{(1)} \} = \arg \min \{ 1 - P^{(1)}(x, x); x \in S_{\text{nop}}^{(1)} \},
\]

and

\[
\rho(N^{(1)}) = 1/(1 - P^{(1)}(x_\rho, x_\rho)).
\]

Assume that there exists no bridgeable point of \(x_\rho\), then \(\forall x\) with \(f(x) \leq f(x_\rho)\), we have \(m^{(1)}(x) \geq m^{(1)}(x_\rho)\).

Consider the sub-matrix \(T_{x_\rho, x_\rho}^{(\mu)}\). Without losing generality, we suppose that \(T_{x_\rho, x_\rho}^{(\mu)}\) is irreducible. Otherwise if the sub-matrix is reducible, we will consider its sub-matrices until they are irreducible.

From Inequality (21) and Lemma 11, we know that \(\forall X = (x_1, \ldots, x_\mu)\) with \(x_1 = x_\rho\) and \(f(x_1) \geq \cdots \geq f(x_\mu)\), it holds

\[
\rho(T^{(\mu)}_{x_\rho, x_\rho}) \geq \min \{ P^{(\mu)}(X, S_{x_\rho}^{(\mu)}); X \in S_{x_\rho}^{(\mu)} \} = 1 - \max \{ P^{(\mu)}(X, S_{x_\rho}^{(\mu)}(f_\geq)); X \in S_{x_\rho}^{(\mu)} \} > 1 - \max \{ \sum_{i=1}^{\mu} P^{(1)}(x_i, S_{x_\rho}^{(1)}(f_\geq)) \}.
\] (36)

Because we assume there is no bridgeable point of \(x_1\), and \(f(x_1) \geq \cdots \geq f(x_\mu)\), then we get

\[
P^{(1)}(x_1, S_{x_\rho}^{(1)}(f_\geq)) \leq P^{(1)}(x_\rho, S_{x_\rho}^{(1)}(f_\geq)),
\]

so Inequality (36) becomes

\[
\rho(T^{(\mu)}_{x_\rho, x_\rho}) > 1 - \mu P^{(1)}(x_\rho, S_{x_\rho}^{(1)}(f_\geq)).
\]

For the \((1 + 1)\)-ARS algorithm, from Lemma 3, we have

\[
\rho(T^{(1)}) = 1 - P^{(1)}(x_\rho, x_\rho) = P^{(1)}(x_\rho, S_{x_\rho}^{(1)}(f_\geq)).
\]

For the \((\mu + \mu)\)-ARS algorithm, we know that

\[
\rho(T^{(\mu)}) \geq \rho(T^{(\mu)}_{x_\rho, x_\rho}).
\]

Thus we get

\[
\rho(T^{(\mu)}) > 1 - \mu(1 - \rho(T^{(1)})).
\]

It yields that

\[
\frac{1}{1 - \rho(T^{(\mu)})} > \frac{1}{\mu(1 - \rho(T^{(1)})},
\]
which equivalent to
\[ R(\mu) < \mu. \]

This contradicts the definition of super linear scalability: \( R(\mu) > \mu \). Hence the assumption doesn’t hold. In other words, a bridgeable point of \( x_\rho \) must exist.

Example 8. Consider pseudo-Boolean optimization. Given the \((1 + 1)\) EA using bitwise mutation with a mutation rate \( 1/n \) and strict elitist selection for solving the Fully-Deceptive function. There exists “bridgeable points” on the related fitness landscape.

The following theorem shows “diversity preservation” is another necessary condition for super linear scalability.

Definition 16. In a \((\mu + \mu)\)-ARS algorithm (\( \mu > 1 \)), given two populations \( X \) and \( Y \). Suppose that there exists at least one individual in \( X \cup Y \) which is different from the super individual. “Diversity preservation” refers to that besides the super individual, at least one of other individuals should be selected into the next generation population with a probability greater than 0.

Theorem 5. Given a family of \((\mu + \mu)\)-ARS algorithms (where \( \mu = 1, 2, \cdots \)) for maximizing the function \( f \). Let \( x_\rho \) be any individual such that
\[ x_\rho := \arg \max \{ N(1)(x, x); x \in S_{\text{nop}}(1) \}, \tag{37} \]
and \( X_\rho := (x_\rho, \cdots, x_\rho) \).

If for some population size \( \mu > 1 \), super linear scalability happens, then the “diversity preservation” condition holds
\[ P(\mu)(X_\rho, S_{x_\rho}(\mu) \setminus \{ X_\rho \}) > 0. \]

Proof. Because \( N(\mu) := (I - T(\mu))^{-1} \), we have
\[ \rho(N(\mu)) = \frac{1}{1 - \rho(T(\mu))}, \]
than from the super linear scalability
\[ R(\mu) > \mu, \]
we get
\[ \rho(T(\mu)) < 1 - \mu(1 - \rho(T(1))). \tag{38} \]

Consider the sub-matrix \( T_{x_\rho \rho}(\mu) \). According to Gershgorin’s Theorem, we know there exists an eigenvalue \( \lambda(\mu) \) of \( T_{x_\rho \rho}(\mu) \) such that
\[ | \lambda(\mu) - P(\mu)(X_\rho, X_\rho) | \leq P(\mu)(X_\rho, S_{x_\rho}(\mu) \setminus \{ X_\rho \}), \]
and then
\[ | \lambda(\mu) | \geq P(\mu)(X_\rho, X_\rho) - P(\mu)(X_\rho, S_{x_\rho}(\mu) \setminus \{ X_\rho \}). \tag{39} \]
From \(38\) and \(39\), we get
\[
P(\mu) (X_{\rho}, X_{\rho}) - P(\mu) (X_{\rho}, S^{(\mu)}_{X_{\rho}} \setminus \{X_{\rho}\}) < 1 - \mu(1 - \rho(T^{(1)})) ,
\]
and then
\[
P(\mu) (X_{\rho}, S^{(\mu)}_{X_{\rho}} \setminus \{X_{\rho}\}) > \mu(1 - \rho(T^{(1)})) - 1 + P(\mu) (X_{\rho}, X_{\rho}). \tag{40}
\]
Because
\[
P(\mu) (X_{\rho}, S^{(\mu)}_{X_{\rho}} (f_{\geq})) < \mu P(1) (x_{\rho}, S^{(\mu)}_{X_{\rho}} (f_{\geq})) ,
\]
and
\[
P(\mu) (X_{\rho}, S^{(\mu)}_{X_{\rho}} (f_{\geq})) = 1 - P(\mu) (X_{\rho}, X_{\rho}) ,
\]
we have
\[
\mu(1 - \rho(T^{(1)})) - 1 + P(\mu) (X_{\rho}, X_{\rho}) > 0 ,
\]
and combining it with Inequality \(40\), we get
\[
P(\mu) (X_{\rho}, S^{(\mu)}_{\text{nop}} \setminus \{X_{\rho}\}) > 0 ,
\]
which is the result that we need.

**Example 9.** Consider pseudo-Boolean optimization. Given the \((\mu + \mu)\) EA using bitwise mutation with a mutation rate \(1/n\) and strict elitist selection with the lowest pressure, then it satisfies the “diversity preservation” condition.

### 4.5 Potential Super Linear Scalability on Deceptive Landscapes

Although using population is useless for solving easy functions such as the One-Max function, it is very helpful for solving hard functions such as the Fully-Deceptive functions. The following theorem will prove this claim.

**Theorem 6.** Given a family of \((\mu + \mu)\)-ARS algorithms (where \(\mu = 1, 2, \ldots\)) using strict elitist selection with diversity preservation (specified below) for maximizing the function \(f\). Suppose that the related time-based fitness landscape is deceptive. If the following two conditions are satisfied:

1. Fitness Diversity Preservation: given populations \(X\) and \(Y\), if there exists one or more individuals in \(X \cup Y\) whose fitness is different from that of the super-individual, then at least one of these individuals must be selected into the next generation population;

2. Bridgeable Point: denote \(S^{(1)}_1\) to be set of individuals with the second highest fitness. For some population size \(\mu > 1\), \(\forall x \in S^{(1)}_1\), and \(\forall y \in S^{(1)}_{\text{nop}} \setminus S^{(1)}_1\) it holds
\[
p_m(x, S^{(1)}_{\text{nop}} \setminus S_1) \geq \mu p_m(x, S^{(1)}_{\text{opt}}) , \tag{41}
\]
\[
p_m(y, S^{(1)}_{\text{opt}}) \geq \mu p_m(x, S^{(1)}_{\text{opt}}) . \tag{42}
\]
Then for the given population size \( \mu \), super linear scalability will happen.

**Proof.** For the \((1+1)\)-ARS algorithm, because the fitness landscape is deceptive, then \( \forall x \in S^{(1)}_{n_{\text{opt}}} \),

\[
x = \arg \max \{ m^{(1)}(y); y \in S^{(1)}_{n_{\text{opt}}} \} = \arg \max \{ 1 - P^{(1)}(y, y); y \in S^{(1)}_{n_{\text{opt}}} \}.
\]

According to the definition of \( x_\rho \) presented in (14), we know

\[
x = x_\rho := \arg \max \{ N^{(1)}(x, x); x \in S^{(1)}_{n_{\text{opt}}} \}.
\]

From Lemma 3, we know that \( \rho(N^{(1)}) = 1/(1 - P^{(1)}(x_\rho, x_\rho)) = 1/p_m(x_\rho, S^{(1)}_{\text{opt}}) \).

For the \((\mu + \mu)\)-ARS algorithm, first let’s consider the sub-matrix \( T^{(\mu)}_{x_\rho, x_\rho} \).

Let \( X_\rho := (x_\rho, \ldots, x_\rho) \). \( \forall Y \in S^{(\mu)}_{x_\rho} \setminus \{ X_\rho \} \), because the ARS uses selection with diversity preservation, we have

\[
P^{(\mu)}(Y, X_\rho) = 0,
\]

then \( T^{(\mu)}_{x_\rho, x_\rho} \) is reducible, which can be written in the following from

\[
T^{(\mu)}_{x_\rho, x_\rho} = \begin{pmatrix} P^{(\mu)}(X_\rho, X_\rho) & * \\ \tilde{T}^{(\mu)}_{x_\rho, x_\rho} & 0 \end{pmatrix},
\]

(43)

where \( \tilde{T}^{(\mu)}_{x_\rho, x_\rho} \) is the sub-matrix representing probability transition among transient states in \( S^{(\mu)}_{x_\rho} \setminus \{ X_\rho \} \). The part * plays no role in the analysis.

Denote

\[
\tilde{N}^{(\mu)}_{x_\rho, x_\rho} := (I - \tilde{T}^{(\mu)}_{x_\rho, x_\rho})^{-1},
\]

\[
N^{(\mu)}_{x_\rho, x_\rho} := (I - T^{(\mu)}_{x_\rho, x_\rho})^{-1}.
\]

Let \( \lambda^{(\mu)} \) be an eigenvalue of the matrix \( T^{(\mu)}_{x_\rho, x_\rho} \).

Then the first eigenvalue of \( T^{(\mu)}_{x_\rho, x_\rho} \) is

\[
\lambda^{(\mu)} = P^{(\mu)}(X_\rho, X_\rho).
\]

Because

\[
1 - \lambda^{(\mu)} = 1 - P^{(\mu)}(X_\rho, X_\rho)
\]

\[
= P^{(\mu)}(X_\rho, S^{(\mu)}_{x_\rho} \setminus \{ X_\rho \}) + P^{(\mu)}(X_\rho, S^{(\mu)}_{x_\rho}(f_>) \),
\]

and from Inequality 11,

\[
P^{(\mu)}(X_\rho, S^{(\mu)}_{x_\rho} \setminus \{ X_\rho \}) > \mu p_m(x_\rho, S^{(1)}_{\text{opt}}),
\]

28
then we get
\[
\frac{1}{1 - \lambda(\mu)} < \frac{1}{\mu p_m(x_\rho, S_{\text{opt}})} = \frac{\rho(N^{(1)})}{\mu}.
\] (44)

Now let’s estimate the spectral radius of matrix $\tilde{T}^{(\mu)}_{x_\rho, x_\rho}$. Without losing
generality, suppose that $\tilde{T}_{x_\rho, x_\rho}$ is irreducible. Otherwise like the analysis of the
first eigenvalue, we will remove those eigenvalues of of the matrix $T^{(\mu)}_{x_\rho, x_\rho}$ which
can be calculated directly from diagonal entries.

Applying Inequality (21) and Lemma 11 into the irreducible matrix $\tilde{T}_{x_\rho, x_\rho}$, we know that
\[
\rho(\tilde{T}_{x_\rho, x_\rho}) \leq \max \left\{ P(\mu)(X, S_{x_\rho}(\mu) \setminus \{X_\rho\}) \}
\leq 1 - \min \left\{ P(\mu)(X, S_{x_\rho}(\mu) \setminus \{X_\rho\}) \right\}.
\]

Using Inequality (42), we get
\[
\rho(N_{x_\rho, x_\rho}) = \frac{1}{1 - \rho(\tilde{T}_{x_\rho, x_\rho})}
\leq \frac{1}{\min \left\{ P(\mu)(X, S_{x_\rho}(\mu) \setminus \{X_\rho\}) \}}
\leq \frac{1}{\mu p_m(y, S_{\text{opt}})} \quad \text{(where } y \in X \text{ but } y \neq x_\rho)\n\leq \frac{1}{\mu p_m(x_\rho, S_{\text{opt}})}
\leq \frac{\rho(N^{(1)})}{\mu}.
\] (46)

From Inequalities (44) and (46), we get that
\[
\frac{\rho(N^{(1)})}{\rho(N^{(\mu)}_{x_\rho, x_\rho})} > \mu.
\]

Considering other sub-matrices $T^{(\mu)}_{x_\rho, x_\rho}$ of $T^{(\mu)}$ where $x \neq x_\rho$ and following a
similar analysis, we can prove that:
\[
\frac{\rho(N^{(1)})}{\rho(N^{(\mu)}_{x_\rho, x_\rho})} > \mu.
\]

Because
\[
\rho(N^{(\mu)}) = \max \{ \rho(N^{(\mu)}_{x_\rho, x_\rho}) : x \in S_{\text{nop}}^{(1)} \},
\]
hence we have
\[
R(\mu) > \mu,
\]
which means super linear scalability happens. □
Example 10. Consider pseudo-Boolean optimization. Given the family of \((\mu + \mu)\) EAs using bitwise mutation and strict elitist selection with diversity preservation for solving the Fully-Deceptive function:

- Fitness Diversity Preservation: given populations \(X\) and \(Y\), if there exists one or more individuals in \(X \cup Y\) whose fitness is different from that of the super-individual, then at least one of these individuals must be selected into next generation population.

Since the \((1+1)\) EA uses bitwise mutation and the function is Fully-Deceptive function, we have

\[
p_m(x, S_{opt}^{(1)}) = \left(\frac{1}{n}\right)^n, \quad \forall x \in S_{1}^{(1)},
\]

\[
p_m(x, S_{nop}^{(1)} \setminus S_{1}^{(1)}) = 1 - \left(\frac{1}{n}\right)^n - \left(1 - \frac{1}{n}\right)^n, \quad \forall x \in S_{1}^{(1)},
\]

\[
p_m(y, S_{opt}^{(1)}) > \left(1 - \frac{1}{n}\right) \left(\frac{1}{n}\right)^{n-1}, \quad \forall y \in S_{nop}^{(1)} \setminus S^{(1)}.
\]

From the above inequalities we can verify Conditions (41) and (42) for \(\mu < n\). So the super linear scalability can happen for \(\mu < n\).

4.6 Road through Bridge: Sufficient Condition for Super Linear Scalability

In the following we provide a general sufficient condition for super linear population scalability.

Lemma 12. Given a family of \((\mu + \mu)\)-ARS algorithms (where \(\mu = 1, 2, \cdots\)) for maximizing the function \(6\). Let \(x_\rho\) be an individual such that

\[
x_\rho := \arg \max \{N^{(1)}(x, x); x \in S_{nop}^{(1)}\}.
\]

If for some a population size \(\mu > 1\), there exists an integer \(k > 0\), \(\forall x \in S_{nop}^{(1)}\), it holds:

\[
\| (T_{x,x}^{(\mu)})^k \|_\infty < \left(1 - \mu(1 - P^{(1)}(x_\rho, x_\rho))\right)^k,
\]

then for the given population size \(\mu\), super linear scalability will happen.

Proof. From Inequality (47), \(\forall x \in S_{nop}^{(1)}\) we get

\[
\left(\| (T_{x,x}^{(\mu)})^k \|_\infty\right)^{1/k} < 1 - \mu(1 - P^{(1)}(x_\rho, x_\rho)).
\]

Because\(^6\)

\[
\rho(T_{x,x}^{(\mu)}) \leq \left(\| (T_{x,x}^{(\mu)})^k \|_\infty\right)^{1/k},
\]

\(^6\)We utilize the following fundamental fact: for an \(n \times n\) square matrix \(A\), \(\rho(A) \leq (\| A^k \|_\infty)^{1/k}\) holds.
then we get
\[
\rho(T_{x,x}^{(\mu)}) < 1 - \mu \left( 1 - P^{(1)}(x_\rho, x_\rho) \right).
\]

From Lemma 3 we know for the (1+1)-ARS algorithm,
\[
\rho(N^{(1)}) = \frac{1}{1 - P^{(1)}(x_\rho, x_\rho)},
\]
then we have
\[
1 - \rho(T_{x,x}^{(\mu)}) > \mu/\rho(N^{(1)}). \tag{48}
\]

Because
\[
\rho(N_{x,x}^{(\mu)}) = \frac{1}{1 - \rho(T_{x,x}^{(\mu)})},
\]
then from Inequality (48), we get
\[
\frac{\rho(N^{(1)})}{\rho(N_{x,x}^{(\mu)})} > \mu.
\]

From Lemma 3 we know for the (1+1)-ARS algorithm,
\[
\rho(N^{(\mu)}) = \max \{ \rho(N_{x,x}^{(\mu)}): x \in S_{\text{nop}}^{(1)} \},
\]
then we get
\[
\frac{\rho(N^{(1)})}{\rho(N^{(\mu)})} > \mu,
\]
which means super linear scalability happens.

The above theorem has an intuitive and equivalent expression by using the concept of “road” [26].

**Definition 17.** Given two states \(X, Y\), if there exists \(k\) states \(\{X_0, \cdots, X_k\}\) where \(X_0 = X, X_k = Y\), and for each \(i = 0, \cdots, k-1\): \(P^{(\mu)}(X_i, X_{i+1}) > 0\), then \(\{X_0, \cdots, X_k\}\) is called a road from \(X\) to \(Y\), denoted by \(\text{Road}(X, Y, k)\) and \(k\) is called the length of the road.

Denote
\[
\text{Road}(X, S_{x}^{(\mu)}(f_\geq), k) = \{ \text{Road}(X, Y, k): Y \in S_{x}^{(\mu)}(f_\geq) \}.
\]

Given an population \(X\) with \(\pi = x\), \(\text{Road}(X, S_{x}^{(\mu)}(f_\geq), k)\) can be classified into two categories:

1. **Road passing through bridge:** all of \(\text{Road}(X, Y, k) \in \text{Road}(X, S_{x}^{(\mu)}(f_\geq), k)\) where at least one of states \(X_1, \cdots, X_{k-1}\) includes a bridgeable point of \(x\).

2. **Road jumping over gap:** all of \(\text{Road}(X, Y, k) \in \text{Road}(X, S_{x}^{(\mu)}(f_\geq), k)\) where none of states \(X_1, \cdots, X_{k-1}\) includes a bridgeable point of \(x\).
Then we have the following sufficient condition for super linear scalability.

**Theorem 7.** Given a family of \((\mu + \mu)\)-ARS algorithms (where \(\mu = 1, 2, \cdots\)) for maximizing the function \((6)\). Let \(x_\rho\) be an individual such that

\[
x_\rho := \arg \max \{ N^{(1)}(x,x); x \in S_{\text{nop}}^{(1)} \}.
\]

If for some population size \(\mu > 1\), there exists an integer \(k > 0\), \(\forall X \in S_{\text{nop}}^{(\mu)}\) with the super individual \(\overline{x} = x\), it holds:

\[
P^{(\mu)}(\text{Road}(X, S_{\text{nop}}^{(\mu)}(f_\geq), k) \text{ jumping over gap})
+ P^{(\mu)}(\text{Road}(X, S_{\text{nop}}^{(\mu)}(f_>, k) \text{ passing through bridge})
> 1 - (1 - \mu(1 - P^{(1)}(x_\rho, x_\rho))^k),
\]

then supper linear scalability will happen for the population size \(\mu\).

**Proof.** This is an alternative expression of Lemma 12 by using the term “road’.

Inequalities (49) and (47) are equivalent. \(\square\)

We call Inequalities (49) “road through bridge” condition. Theorem 7 shows “road through bridge” condition is sufficient for super linear scalability.

**Example 11.** Consider pseudo-Boolean optimization. Given the family of \((\mu + \mu)\) EAs using bitwise mutation and strict elitist selection with diversity preservation for solving the Fully-Deceptive function:

- Fitness Diversity Preservation: given populations \(X\) and \(Y\), if there exists one or more individuals in \(X \cup Y\) whose fitness is different from that of the super-individual, then at least one of these individuals must be selected into next generation population.

Then it satisfies the “road through bridge” condition with \(k = 2\) and \(1 < \mu < n\).

## 5 Conclusions and Discussions

The paper provides a general discussion of population scalability for a large family of RS algorithms, called ARS. The main contribution of this paper is to introduce a new approach based on the fundamental matrix for analyzing population scalability. The performance of ARS is measured by a new index: spectral radius of the fundamental matrix. Through analyzing the fundamental matrix of Markov chains associated with ARS, the following results have been proven.

1. Increasing population size may increase population scalability.

2. No super linear scalability is available on any regular monotonic fitness landscape. This implies that using population will not bring any benefit to solve easy problems such as the One-Max function.
3. Potential super linear scalability may exist on deceptive fitness landscapes. This means that using population might be great help to solve hard problems such as the Fully-Deceptive function.

4. "Bridgeable point" and "diversity preservation" are two necessary conditions for super linear scalability on all time-based fitness landscapes. This hints that the performance of $(1 + \lambda)$ EAs under spectral radius is never better than $(1 + 1)$ EA. So using population is useless in $(1 + \lambda)$ EAs.

5. “Road through bridges” is a sufficient condition for super linear scalability. These results are consistent with common sense from the practice. For example using a population is better than an individual for solving a hard problem; it is important to maintain population diversity. Population scalability analysis provides a theoretical support to those claims from practice.

It should be stressed again that ARS covers all types of global search and strict elitist selection, and the objective function could be any one on a finite set. Hence the assertions derived from the paper are true for a wide range of RS algorithms and discrete optimization problems.

There are still several research questions needed to answer in the future. First, ARS discussed in this paper is based on strict elitist selection. However in practice, some algorithms take non-strict elitist selection, i.e., the super individual may be replaced by an individual with the same fitness level. In this case, the fundamental matrix of the benchmark algorithm $(1 + 1)$-ARS is not lower triangular [30].

A further argument about the benchmark algorithm is that a $(1+1)$-ARS algorithm using elitist selection does not accept a worse solution, but $(\mu + \mu)$-ARS algorithms accept a worse one. Thus it seems better to allow the benchmark algorithm to accept a worse solution with a probability. In this case the benchmark algorithm becomes a simulated annealing algorithm [33].

Secondly, a more challenge work is to analyze population scalability of ARS using recombination operators like crossover. Because a $(1 + 1)$-ARS algorithm doesn’t include a recombination operator, therefore it is not an appropriate candidate as the benchmark for investigating population scalability. Instead the comparison will be based on an ARS-(2) algorithm with recombination, thus population scalability is redefined by

\[
R(\mu) := \frac{\rho(N^{(2)})}{\rho(N^{(m)})}.
\]

Finally, a research issue is to study population scalability through the trace of the fundamental matrix of ARS. As pointed in Section [3] the trace of the fundamental matrix represents staying times for the Markov chain in transient states in the average case, while \(\rho(N)\) represents staying times for the Markov chain in transient states in the worst case.
Appendix

In this appendix, we take an example to demonstrate that the performance of an ARS algorithm under spectral radius might equal the hitting time either in the best case or in the worst case. This example also shows the performance of ARS under spectral radius can be interpreted as an index of measuring how long an ARS algorithm spends in the most difficult search area.

Consider pseudo-Boolean optimization. Given the \((1 + 1)\) EA using bitwise mutation with a mutation rate \(1/n\) and strict elitist selection.

First we investigate the case of maximizing the One-Max function, which takes \(n + 1\) fitness levels: \(f_0 > f_2 > \cdots > f_n\). Denote \(S_l\) to be the set of states at the level \(f_l\).

According to Lemma \ref{lem:one-max}, we know spectral radius of the fundamental matrix is

\[
\rho(N) = \max\left\{ \frac{1}{1 - P(x, x)} ; x \in S_{\text{nop}} \right\}
= \frac{1}{P(x, S_0)} \quad (x \in S_1)
= (n - 1) \left( \frac{n}{n - 1} \right)^n.
\]

The hitting time in the best case is that of the Markov chain starting in \(S_1\), which equals

\[
\min\{m(x) ; x \in S_{\text{nop}}\} = (n - 1) \left( \frac{n}{n - 1} \right)^n.
\]

Secondly we investigate the case of maximizing the Fully-Deceptive function, which takes \(n + 1\) fitness levels: \(f_0 > f_2 > \cdots > f_n\). Denote \(S_l\) to be the set of states at the level \(f_l\).

According to Lemma \ref{lem:full-deceptive}, we know spectral radius of the fundamental matrix is

\[
\rho(N) = \max\left\{ \frac{1}{1 - P(x, x)} ; x \in S_{\text{nop}} \right\}
= \frac{1}{P(x, S_0)} \quad (x \in S_1)
= n^n.
\]

The hitting time in the worst case is that of the Markov chain starting in \(S_1\), which equals

\[
\max\{m(x) ; x \in S_{\text{nop}}\} = n^n.
\]

Lemma \ref{lem:ars-spectral} proves that the performance of ARS under spectral radius is between the best and worst case hitting times. The above analysis demonstrates that the performance might equal either the worst case or the best case hitting time.
The above results also indicates the performance of ARS under spectral radius is an index of measuring how long an ARS algorithm spend in the most difficult search area.

In the One-Max function, the most difficult search area for the \((1 + 1)\) EA is between \(S_1\) and \(S_0\). Spectral radius of the fundamental matrix equals to the hitting time from \(S_1\) to \(S_0\).

In the Fully-Deceptive function the most difficult search area for the \((1 + 1)\) EA is between \(S_1\) and \(S_0\). Spectral radius of the fundamental matrix equals to the hitting time from \(S_1\) to \(S_0\).

Acknowledgments

We would especially like to thank Prof. Günter Rudolph who has been involved in initializing the topic of super linear scalability. We also wish to thank Dr. Carsten Witt, Dr. Dan Lin and Prof. Xin Yao who gave us useful suggestions. J. He is supported by the EPSRC under Grant EP/I009809/1.

References


