Chaotic populations in genetic algorithms

Zhanshan (Sam) Ma

Computational Biology and Medical Ecology Lab, State Key Laboratory of Genetic Resources and Evolution, Kunming Institute of Zoology, Chinese Academy of Sciences, Kunming, 650223, PR China

A R T I C L E   I N F O

Article history:
Received 25 July 2008
Received in revised form 9 February 2012
Accepted 4 March 2012
Available online 3 April 2012

Keywords:
Genetic algorithm
Chaos theory
Logistic Chaos map
Dynamic populations
Fitness aggregation
Power Law

A B S T R A C T

We set two objectives for this study: one is to emulate chaotic natural populations in GA (Genetic Algorithms) populations by utilizing the Logistic Chaos map model, and the other is to analyze the population fitness distribution by utilizing insect spatial distribution theory. Natural populations are so dynamic that one of the first experimental evidences of Chaos in nature was discovered by a theoretical ecologist, May (1976, Nature, 261,459–467) [30], in his analysis of insect population dynamics. In evolutionary computing, perhaps influenced by the stable or infinite population concepts in population genetics, the status quo of population settings has dominantly been the fixed-size populations. In this paper, we propose to introduce dynamic populations controlled by the Logistic Chaos map model to Genetic Algorithms (GA), and test the hypothesis – whether or not the dynamic populations that emulate chaotic populations in nature will have an advantage over traditional fixed-size populations.

The Logistic Chaos map model, arguably the simplest nonlinear dynamics model, has surprisingly rich dynamic behaviors, ranging from exponential, sigmoid growth, periodic oscillations, and aperiodic oscillations, to complete Chaos. What is even more favorable is that, unlike many other population dynamics models, this model can be expressed as a single parameter recursion equation, which makes it very convenient to control the dynamic behaviors and therefore easy to apply to evolutionary computing. The experiments show result values in terms of the fitness evaluations and memory storage requirements. We further conjecture that Chaos may be helpful in breaking neutral space in the fitness landscape, similar to the argument in ecology that Chaos may help the exploration and/or exploitation of environment heterogeneity and therefore enhance a species’ survival or fitness.

© 2012 Elsevier B.V. All rights reserved.

1. Background

In almost every major field of evolutionary computing, population size is one of the parameters that a researcher has to deal with. In nature, natural selection as the driving force of evolution acts upon populations, and in computing science, group search is arguably the unique feature of evolutionary computing. Therefore, it was no surprise that the pioneers of evolutionary computing, such as De Jong [6] and Holland [15], and a little bit later by Baker [4], Goldberg [10], etc. had already studied the population sizing from the very beginning of evolutionary computing. What is perhaps surprising is that more than three decades later, we still largely depend on the experience or ad-hoc trial-and-error approach to set the population size. In their recent monograph for evolutionary computing, Eiben and Smith [8] indicated: “In almost all EC applications, the population size is constant and does not change during the evolutionary search”. Does the status quo imply that population size is not an important parameter?

The answer to the previous question seems a definite no. Current practice of manual setting of population size in evolutionary computation is experience-based, but not robust. Too small of populations can lead to premature convergence, and too big of populations can be computationally costly. In particular, in Genetic Programming (GP), big population is often a culprit for the too early occurrence of code bloat, which may cause computation failure or even crash the system. This manual setting by experienced programmers is acceptable for design modeling or ad-hoc applications; it is problematic for real-time applications, which require high predictability and robustness. In the latter scenarios, automatic, adaptive and robust population sizing (also other parameters such as mutation and crossover rates) is necessary. On the practical side, what is equally important is the robustness and adaptability of the algorithms. Furthermore, many of the problems we face are NP-hard problems; efficient population sizing may have significant impacts on the success of the heuristic algorithms. However, an optimal population size effective in exploring fitness space and efficient in utilizing computing resources is theoretically very
intriguing. Achieving the balance in effectiveness and efficiency is such a dilemma that prompted Harik et al.'s [14] approach to the problem with a gambler's ruin random walk model, one of the most rigorous analyses of the population size problem.

Rylander [33] reviewed the population size issues in GA and stated: "There has yet to be a proven or even generally accepted optimum population size for all problems. Since earlier work indicated that no single optimum existed, there has been little recent work attempting to determine such an optimum. The most notable tradeoffs are as follows: the larger the population the more quickly a solution space can be explored, the smaller the population the more quickly it can converge to optimum once it is found. Unfortunately, there is no agreement as to how large the large is or conversely how small the small is."

Early pioneering studies of population sizing are largely ad-hoc, problem-oriented and empirical. Many of the studies were simple sample comparisons of multiple population sizes or were compared with De Jong [6] research (e.g., Grefenstette [43], Schaffer [34], Gates et al. [9], Odetayo [32], Yuen and Ma [42], Costa et al. [5]). Goldberg instead analyzed the optimal population size theoretically, and found that the optimal population size increases exponentially and is rather large for even moderate chromosome size [10]. Goldberg was also the first to study the relationship between fitness variance and population size, and later derived a conservative bound for the convergence quality of the Gas [11]. Goldberg et al. further analyzed population-sizing based on recombination mixing, disruption, deception, population diversity, and selective pressure [13]. As argued by Alander, the results seem rather pessimistic [2]. Alander cited Goldberg's own assessment on his theoretical analysis: "too few empirical studies have been performed to know whether the theory provides quantitatively accurate predictions" [2].

Alander [2] reasoned from linear algebra that to cover each point of a given $n$ dimensional space, at least $n$ base vectors are needed. Alander further argued that for small values of $n$, the optimal population size may be approximately equal to $n$. Alander [2] also mentioned that in nature, large populations are more stable and resistant to evolution than small populations that may be founded by a few colonists, or the "founder effect". Although the founder effect implies that the population is small initially, and it may not tell much information about the population changes in later stages. In fact, the Chaos population model implies that initial population size has little values in predicting population dynamics because the extreme sensitivity to initial population size is a signature of chaotic populations. Alander [2] also attempted to correlate population size with problem complexity, but the characterization was solely dependent on the experiment curves and contained no biological/algorithimic arguments. Arabas et al. [3] pointed out that there are two important issues in the genetic search: population diversity and selective pressure, both of which are strongly related to each other. Both are also influenced by population sizes. It is important to achieve a balance between diversity and selective pressure. Arabas et al. [3] introduced the concept of "age" of a chromosome, which is equivalent to the number of generations the chromosome stays "alive". The age of the chromosome replaces the concept of selection. Their experiments showed that to achieve high fitness, a large population has to be maintained. Arabas et al. [3] concluded that while the variable population seemed favorable over the fixed population, the major problem of determining the lifetime parameter remained unresolved.

Dynamic population sizing in multi-objective evolutionary algorithms (MOEA) is considered an open problem (Khor et al. [17]). Khor et al. [17] approached the dynamic size for MOEA with the following intuitive concept: in an $m$-dimensional objective space, assuming that the desired population size at generation $n$ is $\text{dps}(n)$, and the population density per unit volume is $\text{ppv}$, the approximated trade-off hyper-area $A(n)$ discovered by the population at generation $n$ can be defined as, $\text{lowbps} \leq \text{dps}(n) \leq \text{uppbs}$, where $\text{lowbps}$ and $\text{uppbs}$ are the lower and upper bounds for the $\text{dps}(n)$. They developed a formula for computing $A(n)$ based on the average of the longest and shortest diameters of the hyper-area (Tan et al. [36,37]). Lu and Yen [19] pointed out that the estimation of $\text{dps}(n)$ by previous authors is a heuristic approximation and its robustness has to be tested with various initial population sizes, and in some occasions may lead to premature convergence. They proposed an alternative approach in which a multi-objective optimization problem is converted into a bi-objective optimization problem in terms of individual's rank and density values (Lu and Yen [19,20]).

Ma and Krings [26] compared simple dynamic populations (such as random, increasing, decreasing, bell-shaped, or inverse bell-shaped fluctuating schemes) with the fixed-size populations and found that the dynamic populations outperform the fixed-size populations and the results consistently outperformed the fixed-size populations with the same average population size. There was not any mathematical model used in that simple dynamic population, and the five schemes were somewhat more similar to fine-tuned static populations, although the term "dynamic population" was used. For example, the "decreasing population" scheme was adopted to perform more intensive exploration in the early stage of GA. In this study, we greatly expand the simple dynamic population schemes with the mathematical models derived from modeling of the natural populations. Ma [27] briefly reviewed biological/ecological principles and models that can be inspirational for developing a unified population dynamics theory in evolutionary computing.

In stochastic approach, Harik et al.'s [14] "gambler's ruin" formulation, as well as the further follow-up study by Ahn and Ramakrishna's [44] seem to be among the most rigorous analytic approaches to the population-sizing problem. Despite this excellent analytic exploration, it appears that further relaxation of the assumptions used in the studies is needed to apply the results in GA practice. However, further relaxation of the assumptions may easily lead to stochastic problems that are intractable analytically.

In summary, it appears that one of the earliest consensuses is that relative small population size in the magnitude of chromosome length is a feasible choice (e.g., De Jong [6], Schaffer [34], Alander [2], Goldberg [101]), but later studies found that population size should exponentially increase with chromosome (string) size (e.g., Goldberg [10,12], Gates [9]). This latter conclusion paints a very pessimistic picture about the efforts of obtaining optimum population size. It might partially explain why three decades after De Jong's [6] and Holland's [15] first examination of the issue, the fixed-sized populations are still dominantly used in practice as indicated by Eiben and Smith [8], despite the extreme significance of this parameter. Rigorous analysis with random walk model by Harik et al. [14], Ahn and Ramakrishna [44] brought fresh insights to the field; however, the problem is still far from solved.

2. Ecological theory: population dynamics and Logistic Chaos map model

2.1. Population dynamics

Population dynamics or the spatial-temporal change of the population size or density is the central topic of population ecology. The mathematical modeling of population dynamics can be traced back to Thomas Malthus's (1798) "An Essay on the Principle of Populations" (cited in Kot [18]), in which Malthus proposed now well-known Malthusian population growth model [Eq. (2) below]. It is well documented that Malthus' work had significant influence
on Darwin’s formation of the idea of “struggle for life” and natural selection. Pierre Verhulst (1845) derived the famous Logistic population growth model, which was re-discovered by Pearl and Read in 1920 [cites in Kot [18], Eq. (1) below]. Logistic model is the solution of a first-order nonlinear differential equation, whose difference counterpart is found to be able to demonstrate chaotic behavior in the 1970s [May 1976] [30] [Eqs. (1), (2), (4) and (5) below].

There are several categories of models in population dynamics. Logistic differential equation and its difference equation counterpart form the foundation for deterministically studying the population dynamics of single species. It has extraordinary rich mathematical properties from stable growth, oscillation, and attractors to Chaos, and its applications have been expanded well beyond biology. Logistic differential equation is also the basis for forming the famous Lotka–Volterra differential equations system for interspecific interactions such as predation and competition. Differential equation or deterministic modeling is traditionally used to model population dynamics, and is still the dominant approach. An alternative to the traditional differential equation modeling is the stochastic modeling with stochastic process models such as birth–death process, but it is often more difficult to get analytic solutions from stochastic models. The third group of analytical model is the matrix model, which can be formed in either deterministic or stochastic, and it turns out that a matrix model is often equivalent to either differential equations or stochastic process models (such as Markov Chains). The fourth type of analytical model is the optimization and game theory model, which was initially applied to study animal behavior or ethology.

Besides the above four categories of analytical population models, there are two other types of major modeling approaches. One is the statistical modeling such as regression and time series analysis of population data, which often involves the collection of experiment or sampling data. The other important approach is computer simulation modeling, which could be the only approach when the analytical solution is not available. When space is incorporated in ecological modeling, simulation is often utilized. The so-called spatially explicit models, individual-based models, agent-based simulation, are the typical examples. In addition, more formal computation paradigm or algorithms, such as cellular automata, Interacting Particle Systems (IPS) can also be used for simulating population dynamics in space and time.

2.2. Population regulation

Population regulation was one of the most contested theories in the history of ecology (e.g., Kingsland [45]), and the debates started in the 1950s and culminated in the 1960s. Even these days, the antagonistic arguments from both schools occasionally appear in ecological publications (e.g., Berryman [5], White [41]). The debate may sound simple from an engineering perspective. The core of the debate lies in the fundamental question: is natural population regulated by feedback mechanisms such as density-dependent effects of natural enemies or is simply limited by its environment constraints. Within the regulation camp, there are diverse theories on what factors (intrinsic such as competition, natural enemies, gene, behavior, movement, migration, etc.) and how these factors regulate populations. The importance of population regulation cannot be overemphasized, since it reveals the mechanisms of population dynamics. Even more important is to treat population dynamics from the time-space paradigm, not just the temporal changes of population numbers. In addition, the concept of metapopulation is also crucial, which implies that local population extinction and recolonization happen often in nature. Obviously, population regulation as control mechanisms for population size is very inspiring for the counterpart problem in evolutionary computation.

2.3. Logistic differential and difference equations

The well-known Logistic model has the differential equation form:

\[ \frac{dN(t)}{dt} = rN(t) \left( 1 - \frac{N(t)}{K} \right) \]  

(1)

\( N \) is the population number at time \( t \) or \( t + 1 \), \( K \) is referred to as environment capacity in biological literature. \( r \) is the growth rate per generation and is similar to \( r_m \), the population intrinsic (or innate) rate of growth, or the intrinsic exponential growth rate of a small population. The \( r \) is a limited version of \( r_m \) and this is clear from the fact that \( r_m \) is derivable from the well-known Malthusian equation,

\[ \frac{dN(t)}{dt} = r_mN(t) \]  

(2)

which is the linear counterpart of the Eq. (1).

2.4. One-parameter logistic difference equation – Logistic Chaos map

Eq. (1) has a unique solution:

\[ N(t) = \frac{K}{1 + a e^{-rt}} \]  

(3)

To facilitate analysis, let

\[ x = \frac{N_r}{K(1+r)} \]  

(4)

Substituting \( x \) into Eq. (3) derives the one-parameter dimensionless Logistic Map, also known as one-hump nonlinear function, or one-dimensional quadratic map.

\[ x_{n+1} = ax_n(1-x_n) \]  

(5)

where \( a = r+1 \). To avoid trivial dynamic behavior, the model requires \( 1 < a < 4 \) and \( 0 < x < 1 \). The population size (\( x \)) is converted to the (0, 1) interval, and the conversion also eliminates the other parameter \( K \), which makes the analysis more convenient. The extremely rich and complex behavior represented by the deceptively simple Eq. (5) was discovered by Robert May [30]. Fig. 13 in the Appendix shows the behavior of the model across various thresholds. One may refer to a monograph on Chaos theory (e.g., Schuster [35]) for in-depth details or some recent publications (e.g., Ma [29]).

The emphasis on population has been explored in many fields of evolutionary computing (e.g., Yao and Liu [40]). With respect to the application of Chaos theory in GA, De Kan and Foster [7] already explored its effects through the settings of mutation rates. In the present study, the focus is centered on using the Chaos model to set the sizes for GA populations, instead.

2.5. Population spatial distributions

In previous sections, we focus on the temporal dynamics of natural populations. In reality, a biological population exists or distributes in its habitat space. Realistically, population dynamics should be characterized by both temporal and spatial changes. Indeed, spatial dynamics is much more difficult to study than the temporal dynamics, and we have to take simplified approach to study it. The so-called spatial distribution patterns or simply spatial distribution is such a simplified view of spatial dynamics of natural populations. Two significant simplifications are made: one is the “point” abstraction, which simplifies the spatial distribution of population individuals in three-dimensional habitat space to two-dimensional plane, and the other is that we start with a “snapshot” of the distribution at a time point. The resulting problem is known as spatial point pattern in spatial statistics. The study of insect spatial
distribution patterns can be traced back to the 1930s; this is well before the recent establishment of spatial statistics. Some of the all-time greatest mathematical statisticians, including Polya and Neyman [31], Fisher, Student, Greenwood, Yule, and Skellam, had paid attention to or directly studied the field (cited in Taylor [39]). For this and other reasons, we borrow the theory from the study of insect populations.

In nature, the population individuals such as insects distribute in their habitat space in some patterns that are believed to enhance their survival. The so-called spatial distribution patterns or simply spatial distributions are considered as emergent behavior at population level. The patterns are fine tuned by natural selection to maximize a species’ fitness. What is happening in nature is that the population individuals constantly redistribute themselves in habitat space in responding to the stimuli from the environment or their peers, and this redistribution leads to the spatial dynamics of population. Spatial distribution pattern is simply a snapshot of population at a particular moment.

We envision that each individual in a GA population of size $N$ carries a fitness $f_i$ ($i = 1, 2, ..., N$) and the distribution of these individual fitness $f_i$ in fitness landscape may have a distribution pattern similar to the spatial distribution patterns of natural populations. In natural populations, favorable habitat space is occupied by more individuals and the crowding or aggregation degree is relatively high, and vice versa in the less favorable habitat space. In natural populations such as insect populations, the spatial distribution patterns are often distinguished as three types: regular, random and aggregated (Fig. 14 in the Appendix).

We first need to define basic metrics, which are measurable in practice, and then build the mathematical model for these metrics. We define three basic metrics based on the counterparts in insect population distribution theory. Since these metrics and the models essentially measure the aggregation or crowding of population in space in natural populations, several terms such as distribution, aggregation, crowding, are used interchangeably in literature of insect study, we simply follow the similar convention.

Mean Fitness ($M$) is the average of the individual fitness values within a GA population. Similarly, the (fitness) variance ($V$) is the statistical (sample) variance of the fitness values among the individuals in a population. Formally, assuming there are $N$ individuals in a population, with fitness value, $f_1, f_2, ..., f_N$, the mean fitness ($M$) and variance ($V$) are computed as:

$$M = \frac{\sum_{i=1}^{N} f_i}{N}$$

$$V = \frac{\left(\sum_{i=1}^{N} (f_i - M)^2\right)}{N}$$

Mean (Fitness) Crowding ($M'$) is computed as:

$$M' = \frac{M + V}{M - V}$$

It measures the average neighborhood crowdedness or aggregation level of fitness, similar to the original definition in biology first proposed by Lloyd [46].

We borrow the following three models from insect spatial distribution theory. Since these models actually measure the population aggregation or dispersion in insect population distribution, the terms aggregation, distribution, or dispersion are often used interchangeably, here we adopt the same convention.

Iwao’s [16] Mean Crowding Linear Model:

$$M' = \alpha + \beta M$$

where $\alpha$, $\beta$ are parameters, and $M'$ and $M$ are mean fitness crowding and mean fitness, respectively. Generally, $\beta < 1$, $\beta = 1$, and $\beta > 1$ correspond to regular (Fig. 14a in the Appendix), random (Fig. 14b in the Appendix) and aggregated (Fig. 14c in the Appendix) distributions, respectively.

Taylor’s [38] Power law model is:

$$V = aM^b$$

or equivalently,

$$\ln(V) = \ln a + b \ln(M)$$

the latter simplifies the fitting of nonlinear Power law. The $a$ and $b$ are model parameters, and $M$ and $V$ are mean fitness and fitness variance, respectively. Generally, $b < 1$, $b = 1$, and $b > 1$ corresponds to regular, random and aggregated distributions, respectively.

Ma’s [21,23,25], PACD (population aggregation critical density) is computed as:

$$m_0 = \exp \left[ \frac{\ln(a)}{1 - b} \right].$$

where $a$ and $b$ are the same as the parameters in Taylor’s Power law. Actually, this is a reinterpretation of Taylor’s Power law by classifying population distributions into three types: aggregation density-dependent (ADD), aggregation inversely density-dependent (AIDD) and aggregation density-independent (ADI). Each of the dependent types may have any of the three standard distributions (regular, random, and aggregated) depending on the relationship between population density and the PACD. In the context of fitness aggregation, PACD may represent the fitness threshold at which the fitness aggregation crosses the random distribution point.

The above three models: Mean Crowding, Power Law, and PACD constitute the first category of approaches we use for characterizing fitness distribution. In the following, we introduce the second approach, the so-called aggregation indexes.

Several aggregation or dispersion indexes have been proposed to measure the population aggregation degree in insect populations. They measure the degree of heterogeneity or aggregation of individuals in space. The simplest one is the ratio of population variance ($V$) to mean ($M$), $V/M$, which equals 1 for random distribution (Fig. 14b in the Appendix), greater than 1 for aggregated distribution (Fig. 14c in the Appendix) and less than 1 for regular distribution (Fig. 14a in the Appendix). Another widely used statistics is $M'/M$, where $M'$ is the mean crowding, as defined in Eq. (8). For random distribution (Fig. 14b in the Appendix), both $V/M = 1$ and $M'/M = 1$, and the mean crowding equals the population mean (fitness). For aggregated (Fig. 14c in the Appendix) and regular distributions (Fig. 14a in the Appendix), the $M'/M$ is greater and less than 1 respectively. The relationship between $M'/M$ and $V/M$ is somewhat complicated (Taylor [39]), and we will make necessary comments in the later sections. In this article, we are more interested in the modeling of the aggregation dynamics across generations in GA populations, similar to the study of population aggregation dynamics in insect populations (Ma [24]).

3. Chaotic populations in genetic algorithms and fitness aggregation

From previous introduction of the ecological theory, we propose two conjectures to test in GA populations: one is the controlling GA population with the Logistic Chaos map model. This is to emulate natural population in the hope that GA populations will behave like natural populations with the increased efficiency in exploring and/or exploiting fitness landscape. This conjecture is inspired by a postulation in ecology that Chaos may promote a species’ fitness by actually reducing the chance of extinction (e.g., Allen et al. [11]). The second conjecture is the possibility that Chaos may promote the heterogeneity of fitness landscape. To examine the second conjecture, we again resort to the natural populations, in specific, the
spatial distribution of natural populations. The approach for testing this second conjecture is based on the theory introduced in previous section (Section 2.5). It should be noted that the usage of "chaotic population(s) in GA" is an analogy, which means chaotic population model, i.e., population-sizing controlled by chaotic population models. Similarly, our usage of dynamic populations in GA means dynamic population models for controlling GA population sizes.

3.1. Chaotic populations in genetic algorithms

In this study, we introduce Logistic Chaos map model [Eq. (5)] to control populations in GA by emulating natural population dynamics. The objective is to test whether or not the chaotic populations that emulate nature improve the search efficiency in GA. The reasons why we choose Logistic Chaos map model are multiple:

(1) Logistic model is the foundation for most analytic and deterministic population dynamics models. Although it is derived from the arguably the simplest nonlinear differential equation, it has extremely rich behaviors from exponential, Logistic growth, periodic oscillation, and aperiodic oscillation to Chaos.

(2) The full spectrum of the rich behaviors is controlled by a single parameter (a), which makes the emulation of extremely diverse natural population dynamics in GA very convenient. The following is a simplified summary for the behaviors of the Logistic map model, and for the detailed discussion, one should refer to Chaos literature such as (May [30], Schuster [35]). While 0 < a ≤ 1: the population converges to zero. While 1 ≤ a < 2: the population stabilizes to the equilibrium point (1−1/a). While 2 < a < 3: the population ultimately converges to the equilibrium (1−1/a), but converges only after initial fluctuation around the equilibrium. While 3 ≤ a < 3.57: the system enters bifurcation stage (cycles of periods 2^n) as show in the Fig. 13 of the Appendix. While a ≥ 3.57: the so-called “acclimation point” and a ≈ 3.83 is the appearance of the first cycle with odd period (not characteristic with 2^n). Passing a ≥ 3.57 is the “onset of Chaos.” While 3.83 < a < 4: beyond the threshold a ≈ 3.83, there are cycles with every integer period and an uncountable number of asymptotically aperiodic trajectories. To generate such a rich set of behaviors with a single parameter analytical model is hardly possible with any other alternative models.

(3) The population size (x) is scaled to [0,1] to convert the model into a single parameter model. This also enables our emulations to be much more convenient because we can simply multiply the quantity x by the maximum number of individuals we ever allow for a specific GA computation. Of course, we need to maintain the minimum population size, two individuals, which can be “rescued” by a simple addition of 2-individuals whenever the Chaos drives the population to the “extinction”.

We conjecture that Chaos may be helpful in breaking neutral space in the fitness landscape, similar to the argument in ecology that Chaos may help the exploration and/or exploitation of environment heterogeneity and therefore enhance a species’ survival or fitness.

3.2. Approach to fitness aggregation of GA populations

As introduced in Section 2.5, it is hoped that the approaches used in modeling insect spatial distribution patterns can provide insights on the fitness aggregation (distribution) in GA populations. We call this second conjecture and have two objectives with regard to it: (i) Model GA fitness distribution or aggregation with the counterpart models in insect spatial distribution modeling. In particular, we test if chaotic populations promote fitness aggregation or heterogeneity in GA populations. (ii) Explore the possibility of utilizing fitness aggregation information, such as the dynamics of fitness aggregation index across generations for adjusting GA search parameters in real time (i.e., during the search process). For the first objective, we adopt the dynamic regression models such as Power Law to characterize GA populations, and for the second objective, we harness the aggregation index approach such as M’/M or V/M.

4. Experiment design

Assume a problem is represented as a string of bits of size L = 32. This string is broken up into blocks of length B. The problem is to find blocks in which all bits are set to one. The fitness function is the number of blocks satisfying this requirement. If B does not divide L evenly, the leftmost block will be of different size. The following are four examples of bit streams and their corresponding fitness values (F).

The hypothesis we try to test in the experiment with the standard Genetic Algorithm (GA) is whether the dynamic populations controlled by Logistic Chaos map model [Eq. (5)] outperforms the conventional fixed-size populations.

To facilitate the comparison with the fixed-size population GA, we simply multiply the population size x at ith generation computed from Eq. (5) with the size of the fixed-size population. This sets the dynamic populations in new schemes to fluctuate between 2 and the size of the fixed-size population, i.e., [2, MaxPSize]. The limit of maximum population size in dynamic populations to the size of the fixed-size population is neither required by biological inspiration nor by GA. It is purely to make the experiment results easier to compare.

Many of the previous studies on population sizing focused on searching for the optimum population size, a simple procedure or formula. This strategy may not work in our opinion. Even if there is an optimum population size as is hoped for, it may be a moving target and may not be easily captured by a straightforward formula. For example, some of the existing studies often test multiple population sizes, say, 32, 64, 128,..., and then try to find the optimum one. Even if there is an optimum size, it may not be in the range being tested. Furthermore, there are potentially infinite numbers of candidates of optimum values to test. Given these recognized difficulties, our strategy is to set a fixed task, and then test which scheme is more efficient. The smaller the population size is, the more efficient the scheme is, since a small population requires fewer fitness evaluations and memory storage space.

The scheme of limiting the maximum size in dynamic populations to smaller than the fixed-size populations is fair or even at the advantage for the fixed-size populations for the following reasons. (i) Since the task is fixed, whichever has the smallest number of fitness evaluations should be declared as the “winner”. (ii) At least in the case of our test problem, bigger populations are clearly at an advantage. This is intuitively obvious for our test problem since a bigger population has more individuals working on generating the possible patterns and therefore, they are more likely to find the solution first, or the first hit to the solution. For this reason, besides comparing the fixed task load, we also compare which scheme first hits the solution. Therefore, we believe that the dual criteria, the fitness evaluations of finishing the fixed task load, as well as the arrival of the first hit to the optimum solution, should address any fairness argument. (iii) Our claim is consistent with Harik et al.’s [14] characterization of the problem – the dilemma between choosing large population to ensure solving the problem and the necessity
to control populations to ensure acceptable performance, which prompted them to formulate the problem as “gambler’s ruin” random walk problem. (iv) As a side note, in the pre-experiment drill runs, we indeed conducted ad-hoc tests of the various fixed-size populations, as well as setting the average sizes in dynamic populations equal or bigger than the fixed-size populations, that is, without limiting the maximum dynamic populations to be smaller than the fixed-population sizes. These ad-hoc tests indeed confirmed that the limit is an advantage to the fixed-size populations. (v) Ma and Krigs [26] experimental studies with ad hoc dynamics populations also supports the points (i)–(iv). The theoretic aspects of these points have been explored in Ma [27,28].

5. Results and discussions

5.1. Chaotic populations in GA

The experiment results are summarized in Tables 2–8, and the common parameter settings used in the study is listed in Table 1. In Table 1, Max trials refer to the maximum number of tests run for each treatment. The multiple trials are run to calculate the statistics of performance parameters. The other parameters are standard GA parameters, e.g., Max Generations refer to the total numbers of generations iterated per trial.

Each experiment starts with a population of randomly generated 32-bits strings of 0 and 1, or the chromosomes. The program stops whenever the optimum solution, all 1-bits string (111...111), is found, or reaches the maximum evaluations set by the product of maximum trials and the generations per trial. In the latter case, it may fail to find the optimum solution.

We define the parameter PopuEvalIndex as,

$$\text{PopuEvalIndex} = \frac{\text{psizeReal} \times \text{meanGenerations}}{\text{pSize}}$$

where $\text{psizeReal}$ is the average population size per generation. It is the same as the initial size in fixed-size populations. For dynamic populations, we actually count the population size for each generation in the program and compute the mean population size per generation. The parameter $\text{meanGenerations}$ is the mean of the generations iterated per trial, before it finds the optimum all 1-bits string in all specified trials. It is computed as the total generations divided by the total number of trials.

In the above Eq. (13), the index is computed for the fixed task [all specified trials]; therefore, multiple hits to the optimum solution are possible. This is what we intended for the fixed task load comparisons. For the arrival of the first hit to the solution, we define a similar index FirstHitPopuEvalIndex as,

$$\text{FirstHitPopuEvalIndex} = \frac{\text{psizeReal} \times \text{FHmeanGenerations}}{\text{pSize}}$$

The psizeReal has the same meaning as in Eq. (13), but the parameter FHmeanGenerations refers to the average number of generations run per trial, before the first hit to the optimum solution. In other words, after obtaining the first optimum solution, the whole experiment is stopped.

Besides comparing the above two indexes, we also compare the number of times the optimum solution is found in an experiment run (consisting of many trials). This number is termed hits per experiment, or simply hits in the Tables 2–8.

We therefore use three parameters: PopuEvalIndex, FirstHitPopuEvalIndex, and Hits defined above, as metrics to compare dynamic population schemes with the standard fixed-size populations. The small PopuEvalIndex and FirstHitPopuEvalIndex

Table 6

<table>
<thead>
<tr>
<th>Treatment (B,MaxPsize)</th>
<th>Actual PSize</th>
<th>PopuEvalIndex</th>
<th>FirstHit/PopuEvalIndex</th>
<th>hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, 64)</td>
<td>44</td>
<td>33,751</td>
<td>11</td>
<td>24</td>
</tr>
<tr>
<td>(4, 64)</td>
<td>44</td>
<td>39,106</td>
<td>3972</td>
<td>11</td>
</tr>
<tr>
<td>(7, 128)</td>
<td>87</td>
<td>79,581</td>
<td>5262</td>
<td>9</td>
</tr>
<tr>
<td>(8, 256)</td>
<td>172</td>
<td>1,59,652</td>
<td>8784</td>
<td>8</td>
</tr>
<tr>
<td>(14, 512)</td>
<td>344</td>
<td>3,35,706</td>
<td>52195</td>
<td>3</td>
</tr>
<tr>
<td>(17, 1024)</td>
<td>664</td>
<td>6,78,199</td>
<td>2,28,647</td>
<td>2</td>
</tr>
<tr>
<td>(24, 1024)</td>
<td>687</td>
<td>Fail</td>
<td>Fail</td>
<td>0</td>
</tr>
<tr>
<td>(24, 4048)</td>
<td>1372</td>
<td>Fail</td>
<td>Fail</td>
<td>0</td>
</tr>
<tr>
<td>Total hits</td>
<td></td>
<td></td>
<td></td>
<td>57</td>
</tr>
<tr>
<td>(24, 3000)</td>
<td>2411</td>
<td>2,38,7782</td>
<td>1,63,104</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 7

<table>
<thead>
<tr>
<th>Treatment (B,MaxPsize)</th>
<th>Actual PSize</th>
<th>PopuEvalIndex</th>
<th>FirstHit/PopuEvalIndex</th>
<th>hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, 64)</td>
<td>36</td>
<td>24,052</td>
<td>1088</td>
<td>34</td>
</tr>
<tr>
<td>(4, 64)</td>
<td>36</td>
<td>32,132</td>
<td>369</td>
<td>11</td>
</tr>
<tr>
<td>(7, 128)</td>
<td>71</td>
<td>68,277</td>
<td>1448</td>
<td>4</td>
</tr>
<tr>
<td>(8, 256)</td>
<td>141</td>
<td>1,29,127</td>
<td>2910</td>
<td>9</td>
</tr>
<tr>
<td>(14, 512)</td>
<td>280</td>
<td>2,77,536</td>
<td>31,136</td>
<td>3</td>
</tr>
<tr>
<td>(17, 1024)</td>
<td>559</td>
<td>5,45,293</td>
<td>23,600</td>
<td>3</td>
</tr>
<tr>
<td>(24, 1024)</td>
<td>559</td>
<td>Fail</td>
<td>Fail</td>
<td>0</td>
</tr>
<tr>
<td>(24, 2048)</td>
<td>1372</td>
<td>Fail</td>
<td>Fail</td>
<td>0</td>
</tr>
<tr>
<td>Total hits</td>
<td></td>
<td></td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>(24, 3000)</td>
<td>1636</td>
<td>1,61,7464</td>
<td>3,53,408</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 8

<table>
<thead>
<tr>
<th>Treatment (B,MaxPsize)</th>
<th>Actual PSize</th>
<th>PopuEvalIndex</th>
<th>FirstHit/PopuEvalIndex</th>
<th>hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, 64)</td>
<td>34</td>
<td>33,342</td>
<td>1707</td>
<td>2</td>
</tr>
<tr>
<td>(4, 64)</td>
<td>34</td>
<td>33,667</td>
<td>8847</td>
<td>1</td>
</tr>
<tr>
<td>(7, 128)</td>
<td>67</td>
<td>65,682</td>
<td>677</td>
<td>4</td>
</tr>
<tr>
<td>(8, 256)</td>
<td>133</td>
<td>1,31,703</td>
<td>59,883</td>
<td>1</td>
</tr>
<tr>
<td>(14, 512)</td>
<td>265</td>
<td>2,62,421</td>
<td>1,51,121</td>
<td>1</td>
</tr>
<tr>
<td>(17, 1024)</td>
<td>529</td>
<td>Fail</td>
<td>Fail</td>
<td>3</td>
</tr>
<tr>
<td>(24, 1024)</td>
<td>529</td>
<td>Fail</td>
<td>Fail</td>
<td>0</td>
</tr>
<tr>
<td>(24, 2048)</td>
<td>1056</td>
<td>Fail</td>
<td>Fail</td>
<td>0</td>
</tr>
<tr>
<td>Total hits</td>
<td></td>
<td></td>
<td></td>
<td>43</td>
</tr>
<tr>
<td>(24, 512)</td>
<td>265</td>
<td>2,62,556</td>
<td>1,83,056</td>
<td>1</td>
</tr>
</tbody>
</table>

The second type of tests is what we called ad-hoc testing; they are conducted by relaxing some constraints imposed on the dynamic populations in the systematic testing. In other words, either the maximum population size (MaxPSize) in dynamic approach is allowed to exceed the fixed-size population, or the maximum number of generations (MaxGenerations) is allowed to exceed the common value (1000) set in Table 1. However, this does not imply that the relaxation is favorable to the dynamic population, because the two metrics, PopuEvalIndex and FirstHit/PopuEvalIndex are independent of the two parameters relaxed, and they are still comparable with the fixed size populations. The only difference between the systematic and ad-hoc testing is that, in the case of ad-hoc testing, the metric hits is comparable only if the PopuEvalIndex in both schemes are the same or there is no significant difference statistically. This sufficient condition guarantees the equivalence of workloads of test cases.

The ad-hoc testing results are printed in italicized font below the systematic testing results in each table. The results from ad-hoc testing are not used in drawing any of the Figs. 1–4 or make general conclusions. They are only used in the context of supplementary discussion.

From Tables 2–8 and Figs. 1–4, we draw the following preliminary conclusions:

1. Dynamic populations controlled (modeled) by the Logistic Chaos map model significantly outperform the fixed-size populations. This is evident from comparing Table 2 (for fixed-size populations) and each of the other 6 dynamic populations (Tables 3–8) with different Logistic chaos parameters from 1.5 to 4.

As explained previously, we only compare the first 8 rows of systematic testing. For example, if we compare Table 2 (fixed-size population) and Table 4 (population with a = 2.5), the fixed
populations failed to find the optimum solution in 2 out of 8 test cases (Table 2), and the dynamic populations only failed once. The comparison of PopuEvalIndex columns reveals that the dynamic populations save approximately 40–46% off the fitness evaluations performed by the fixed-size populations in the corresponding treatments.

The comparison of FirstHitPopuEvalIndex, which measures the fitness evaluation index of the first hit to the optimum solution, shows that the dynamic populations save about 62–97% off those of the fixed-size populations, except in one of the 8 cases. In the exceptional case, the dynamic population (Table 4, row 3) performed 74% poor than the fixed-size population (Table 4, row 3) in terms of the FirstHitPopuEvalIndex; however, in this same exceptional case, the dynamic population performs 43% better than the fixed-size population in terms of PopuEvalIndex. In addition, the hits index is 9 vs. 3 in favor of dynamic population in the same exceptional case. Therefore, this exception in one of the indexes (FirstHitPopuEvalIndex) is very likely a random event, given that the other two indexes PopuEvalIndex and hits still support the general trend.

The number of hits to the optimum solution is equally favorable to the dynamic populations, with the total hits ratio of 58 vs. 26 (last columns in Table 4 vs. Table 2).

The comparisons of Table 2 (fixed-size) with the tables for other dynamic populations (Tables 3, 5–8) reveal a similar trend.

Figs. 1–3 are drawn to demonstrate the trend from the Tables 2–8. To save page space, only tests for three “block-size” (B = 1, 8, 17) are included, but each graph is drawn with the results across all 7 population-sizing schemes (i.e., pick results for the three block-size treatments from each of the Tables 2–8). Fig. 1 shows the histograms of PopuEvalIndex and FirstHitPopuEvalIndex for the blocksize (B) = 1. Apparently, the highest fitness evaluation occurs in the fixed size populations, which implies high computational cost with the fixed-size populations. Figs. 2 and 3 are for the block sizes (B) = 8 and 17 respectively, and they show a similar trend.

Fig. 4 shows the total number of hits for each population schemes, the total for each scheme is summed up from the hits column of the corresponding table.

1. There is a significant improvement for the fixed-size populations over the dynamic populations in terms of fitness evaluation. While the dynamic populations do better in terms of fitness evaluation, they still perform worse than the fixed-size populations in terms of fitness evaluation.

2. The Logistic Chaos map model used in this study has rich behaviors that capture the broad spectrum of natural biological populations. Of course, the populations described by the model are not limited to chaotic populations, as explained previously. A natural question would be which of the population dynamic types (controlled by various 1 < a < 4) is more suitable for GA. We intentionally chose the thresholds values in the experiments such as “accumulation point” or the “onset of Chaos” (a = 3.57), “start of odd cycles” (a = 3.68), “total chaos” (a = 3.83), “stable equilibrium range” (1 < a < 2, we take a = 1.5), and “stable equilibrium with initial oscillations” (2 < a < 3, we take a = 2.5).

We intentionally tested the case a = 4, which is not a legitimate parameter value for natural population dynamics. The results (Table 8 and Figs. 1–3) show that this population setting performs worst out of all the dynamic populations in general, although it still outperforms the fixed-size populations in terms of fitness evaluation. However, Fig. 4 shows that the a = 4 scheme performs worst in terms of the hits index. Therefore, it seems that this biologically illegitimate value is not helpful for GA either. We omit the parameter range 0 < a < 1 in the study, although that is a legitimate parameter range in biological populations, because the populations in this parameter range often converge to zero too fast to be useful in GA. Therefore, we believe that the useful range for GA should be between 2 and 4, i.e., 2 < a < 4. As to the best parameter value within the range, our opinion is that the answer should be problem dependent. Fig. 4 suggests that in the case of the test problem in this study, parameter a = 2.5, 3.57, 3.68 and 3.83 seem to perform equally well. It is interesting to note that the “onset of Chaos” (a = 3.57) performs best, in terms of hits index. It is even more interesting to conjecture this threshold might play similar mechanisms in GA populations as in natural biological populations.

3. The ad hoc testing cases in the Tables 2–8 are devised to relax the strict constraints by increasing the maximum population sizes or the maximum number of generations. For example, in Table 4, row 10, the maximum generations is allowed to be 2000, and the first hit to the optimum solution only took 70765 vs. 782714 for fixed population (Table 2, row 10). The improvement is more than 10-fold. However, at this stage, with regard to the ad hoc testing cases, we only try to verify that the dynamic populations, once the artificial constraints are lifted, are able to find the optimum solution in the neighborhood of failed test cases. The results of dynamic populations are always significantly better than the fixed-size populations in terms of the FirstHitPopuEvalIndex and hits, although the performance of PopuEvalIndex may vary. We do not draw any general conclusion about the ad hoc testing cases. Our opinion is that these ad hoc testing cases may be helpful in identifying further research problems.

4. While this study demonstrates the performance advantage of dynamic populations, and the feasibility of using Logistic Chaos map model to control GA populations to emulate biological population dynamics in nature,
we consider that a follow-up study on the relationship between the problem complexity and the parameter (a) of Logistic map model should reveal more interesting results.

5.2. Analysis of fitness distribution in GA populations

In previous section, we focus on the controlling GA population with the Logistic Chaos map model. This section focuses on the analysis of fitness aggregation and its dynamics over generations. The biological principles and mathematical models for this approach are introduced in Section 2.5 and Section 3.2, where we envisioned a mapping between GA fitness distribution and the spatial distribution of an insect population. We conceived two objectives in Section 3.2: (i) the characterization of fitness distribution (aggregation) of GA populations with dynamics models such as Power Law and (ii) the exploration of fitness aggregation dynamics across generations with aggregation index approach.

We collected three types of fitness distribution data for each experiment treatment (e.g., \(\alpha = 3.57\), block size \(B = 4\), corresponding to each row in the Tables 2–8): the across-trial fitness, the across-generation fitness, and the mixture fitness distributions. We use the terms fitness aggregation, fitness distribution and fitness heterogeneity interchangeably in this paper, since they simply emphasize different facets of the same process and are described with the same set of mathematical models. This is also the convention followed in insect distribution theory.

The data collection is conducted as the GA search is being performed and the analysis is conducted at the end of the GA search. More specifically, we record and compute the mean fitness (\(M\)), fitness variance (\(V\)) and fitness crowding (\(M'\)), across trials (MaxTrials = 100), across generations (MaxGenerations = 1000) and across all iterations (100 trials, each has 1000 generations), and then fit the above models [Eqs. (9), (11) and (12)]. The computation was performed in the same GA program at the end of the search.

The across-trial fitness distribution measures the heterogeneity (aggregation) between different experiment trials (MaxTrials = 100, Table 1), which reflects the random variations between trials. The across-generation fitness distribution measures the fitness heterogeneity (aggregation) between generations, which is much more complex with the heterogeneities from at least two sources. The first source is the heterogeneity caused by the fluctuation of the chaotic population from generation to generation. The second source is the ongoing evolution process in GA from generation to generation, such as the effects of mutation, recombination and selection. We also collected a third type of fitness distribution data, the mixture of both across-trial and across-generation distribution data, which is done for exploratory purposes.

We fitted Iwao’s Mean Crowding (\(M'\)) linear model [16], Taylor’s Power Law, and computed Ma’s PACD (Population Aggregation Critical Density, \(m_0\)) in the context of the fitness distribution of GA populations. The results are summarized in Table 9. All the parameters in Table 9 (except for \(R^2\)) have been explained previously, \(R^2\) is the standard coefficient of determinants, \(0 \leq R^2 < 1\). The larger the value of \(R^2\) is, the better the model fitting is. We use \(R^2\) as the criteria to judge the goodness of model fittings.

Table 9 includes the model parameters for two categories of populations: one category is the fixed-size populations and the other is the chaotic populations with parameter \(\alpha = 3.57\) as a representation of the newly introduced chaotic populations. The first column in the table specifies the treatment. For example, “B = 1” is for block size \(B = 1\), and FT is for “Fixed-size aggregation across Trials”. Each treatment is then specified by the combination of two initials: the first initial is either ‘F’ for fixed size population or ‘C’ for chaotic population, and the second is ‘T’ for across trial, ‘G’ for across generation, or ‘M’ for the mixed.

\[ R^2 = \frac{\sum (y_i - \bar{y})^2}{\sum (y_i - \bar{y})^2} \]

MaxSize = 2048 vs. 1024 in the previous treatment.

Figs. 5 and 6 show the fitting of Mean Crowding and Power Law models to the across-generation fitness distribution in a GA population controlled by Logistic chaos model with parameter \(\alpha = 3.57\).

We draw the following preliminary conclusions from Table 9:

(i) In general, both Meaning Crowding linear model (\(M' = M\)) and the Power Law models fit the fitness distribution data extremely well. Most values of Coefficients of Determinants exceed 0.99, which convincingly verifies that these two models are statistically appropriate and extremely reliable for GA fitness distribution. However, since both models fit to both
fixed-size and chaotic populations extremely well, a natural question would be: what are the characteristics of chaotic populations? Before conducting the experiments, we were initially motivated to find the difference between two categories of populations so that the advantages of chaotic populations exhibited in previous section may get an explanation. The experiment results show the universal applicability of Power law in GA.

Given that there is no significant difference between the fixed-size and the chaotic population with regard to the model applicability, a second interesting question, which we also conceived before the experiments, emerges: that is, is there any difference between the populations that successfully find the optimum solution and the populations that fail to find the optimum solution? We will return to these two questions in (iii) and (iv).

(ii) Table 9 also shows that Taylor's Power law is more robust than the Mean Crowding model, given that the latter failed to fit fitness distribution data in some cases, but the Power Law fitted the data almost universally well. Therefore, we suggest using Taylor’s Power Law as well as associated Ma’s PACD as major models for modeling the GA fitness distributions. In addition, although we do not list the results for other chaotic populations (i.e., Eq. (5) with other α values), the model fitting results are very similar to Table 9 and the conclusions we draw here are well supported by the omitted results.

(iii) From the pair-wise comparison of the PACD, it seems that the values of PACD for successful populations and chaotic populations are generally smaller than those for the fixed-size or failed (to find optimum solution) populations. This is indeed consistent with biological meanings in natural populations. However, this trend appeared only in the smaller block size populations ($B < 14$). There are exception to this trend even for the populations with $B < 14$.

In insect populations, PACD is a threshold of aggregation. A higher PACD implies that the threshold for population to aggregate is relatively high [this is only true for $b > 1$ and is opposite when $b < 1$ (see Table 11 in Appendix)]. In terms of GA, this may imply that a large PACD value signals that the threshold for fitness to aggregate is relatively high and therefore it is more difficult to achieve the aggregation of fitness. Of course, this is based on the assumption that a high aggregation of fitness leads to the approach to the optimum solution. Indeed, the results from the next section seem to support this assumption.

We emphasize that the above discussion is only true when $b > 1$, or the population is the so-called aggregation density dependent (ADD) in term of Ma’s PACD. The term density, which is used in insect populations, is equivalent to fitness density or mean fitness in GA populations. In natural populations, when $b < 1$, the aggregation is inversely density dependent (AIDD), this seem to correspond to the GA populations of block sizes larger than $17$. Unfortunately, PACD in these cases computed approach to zero, and we cannot draw meaningful conclusions at this stage. However, it appears that there are three types of fitness aggregations in GA populations, corresponding to Ma’s classification (see Table 11 in Appendix), but this needs further exploration obviously.

In Table 9, we also computed the average values of the parameters $β$, $b$, and $m_0$ (N/A treatments are excluded in the calculation). These averages show that the parameters are indeed problem-dependen, as expected.

(iv) We do not detect significant (statistically significantly different) or meaningful (which is consistent with biological inspiration or GA counterpart) differences in the parameters between chaotic populations or between the chaotic populations and the fixed-size populations. This seems to suggest that the effects of chaotic populations are not necessarily on the aggregation of fitness. In other words, we believe that fitness aggregation is a property independent of the kinds of populations one uses in GA search. However, at this stage, we do not exclude the possibility that chaotic populations may affect the fitness distributions differently from the dynamic perspective. The models fitted, except for PACD, are static in the sense that the parameters summarize the characteristics of fitness aggregation, largely in aftermath; they have limited values in characterizing the dynamic process of fitness aggregation, which we term progression or dynamics of fitness distribution (aggregation). PACD considers the relationship between current population fitness and the threshold value $m_0$, which may characterizing the current population by comparing whether $M > m_0, M = m_0$, or $M < m_0$; however, this information is not sufficient to guide GA to possibly change search parameters such as population size or generation length. The latter ability is what ultimately concerns us the most – the capability to take feedback alive from the dynamics of fitness aggregation and take adjustment accordingly in the ongoing GA-search. We will return to this problem in the next section on fitness aggregation dynamics.
In summary, the important conclusion we confirm in this section is that Taylor's Power Law and to lesser extent, Iwao's Mean Crowding model [16] are near universally applicable to the fitness distributions in GA populations. In addition, it appears that classifying populations, based on Ma's PACD, as three categories: aggregation density (fitness) dependent, aggregation-inverse density (fitness) dependent, and aggregation density (fitness) independent is meaningful in GA. There seems to be a trend in the aggregation density (fitness) dependent populations (Block size $B < 14$) with regard to the values of $PACD$, that is, the chaotic populations and successful (in finding the optimum solution) populations seems to generate smaller $PACD$ values. This is indeed consistent with the mechanisms in natural populations. Unfortunately, we can not extend the observation to the other two types of populations at this stage, and therefore, the general applicability of Ma's $PACD$ in GA populations needs further examination.

The major objective of the models we adopted in this section is to characterize the fitness distributions in general. Although a GA program may dynamically monitor the dynamics of the parameters such as $a$, $b$, $m_0$ in the search process, i.e., fitting the models periodically, the computation is relatively expensive. Even if the computation cost is not a concern, these parameters are mainly summary statistics and offline nature; they are less sensitive to the ongoing search status. Therefore, we mainly use the models in this section to analyze the fitness distribution. To dynamically monitor and potentially take feedback from the fitness aggregation dynamics, we may adopt a relatively simple method, the aggregation index dynamics. We discuss this aspect in the next section.

5.3. Dynamics of the fitness aggregation indexes

To dynamically monitor fitness aggregation (distribution), we “borrow” the so-called aggregation or dispersion indexes in insect population distribution theory. We adopt two indexes: one is the $M' / M$ and the other is $V / M$, where $M$, $V$, and $M'$ are the population mean fitness, variance, and crowding, respectively. There is a subtle difference between the two indexes. We refrain from the discussion of the difference here since both indexes represent two highly contested theories in insect spatial distribution research. Actually, this contention also exists in Iwao's Mean Crowding and Taylor's Power law since both models happen to use two different measurements of heterogeneity, mean crowding ($M'$) and variance ($V$) [Iwao [16]; Taylor [39]]. Here, we would like to just make one comment about their differences: $M' / M$ vs. $V / M$, or their counterpart in dynamic modeling, Iwao's Mean Crowding model vs. Taylor's Power law model, are largely equivalent to each other in the sense that their conclusions are consistent when $\beta > 1$ and $b > 1$. This is equivalent to Ma's aggregation density-dependent populations. When $\beta \geq 1$ or $b \leq 1$, both approaches may behave differently and even reach contradictory conclusions with a same population. The latter is largely equivalent to Ma's aggregation density-independent and aggregation inversely density-dependent populations. According to Ma [21, 23, 25], the inconsistencies between the two methods are due to the biological reality that the sets of parameters ($\alpha$ and $\beta$ or $a$ and $b$) are not sufficient to determine the aggregation distribution, because of the density-dependence effects, which motivated his proposal of the $PACD$ concept.

Unlike the previous section, it only makes sense to monitor the dynamics (or progression) of fitness aggregation index over generations. In other words, we do not think it is meaningful to assess the dynamics across trials or in the mixture mode, since the former represents random variation and the latter contains potentially conflicting progression information.

As stated previously, our motivation for exploring the dynamics of fitness distribution over generations is to adjust search parameters based on the feedback from the real-time monitoring of the ongoing search. However, further study in this direction requires extensive follow-up experiments and we cannot be comprehensive here. Therefore, we use two examples in the following to outline our strategies for the areas of further research.

1. The first example – The chaotic population: $[(B, MaxPSize) = (8, 256), \alpha = 3.57]$. Fig. 7 shows the progression of fitness aggregation index $V / M$ across generations. It exhibits the dynamics of fitness aggregation over generations. From Fig. 7, we identify that around the 250th generation, the fitness aggregation index $V / M$ reaches the maximum. With this information, we conjecture that the probability for GA search to first hit the optimum solution should dramatically increase when the GA population grows beyond 250 generations. We conducted some ad hoc tests, by changing the maximum generations. The following are three test results, with maximum generations set to 250, 500, and the default 1000 (Table 10).

The results in Table 10 show that, although the number of first hits (measured with FirstHitPopuEvalIndex) increases with the increases of maximum number of generations allowed, which is evidently natural (i.e., the longer the search, the more hits to the solution), search beyond the generation 250th is simply the “wasting” of resources because we only need to find the optimum solution once. With the maximum generation size set to the 250, the savings in population evaluation index is about 75% while achieving the same number of hits to the optimum solution as with the other two settings of the maximum generations.

Fig. 7 also shows that the fitness aggregation monotonically increases and then reaches the stable maximum. The monotonic increase at the initial stage can also be seen from the Logistic model (i.e., Eq. (17), fitted with fitness aggregation data), from which Fig. 7 is drawn. This suggests that the fitness aggregation dynamics seems to follow the Logistic growth curve, which is one of few mathematical models that are considered “law” in biology. Indeed, the
Logistic model and Power law are considered the universal laws for temporal and spatial dynamics of insect populations, respectively.

We adopt the following form of the Logistic model:

\[ I(t) = \frac{K}{1 + Ce^{-r}} \]  

(15)

where \( I(t) \) is the aggregation index, either \( V/M \) or \( M'/M \), at generation \( t \); \( K \), \( C \), and \( r \) are the three parameters of the Logistic model.

Similarly, the same model can be used to model the fitness progression, \( M(t) \):

\[ M(t) = \frac{K}{1 + Ce^{-r}} \]  

(16)

The Logistic model parameters for the fitness aggregation dynamics (over generations) in Fig. 7 are:

\[ K = 1.2287, C = 1.0105, r = 0.0109, R^2 = 0.96 \]

\[ V = K \frac{1 + C \exp(−rt)}{1 + 1.0105 \exp(−0.0109t)} \]  

(17)

The Logistic model for fitness dynamics is:

\[ K = 2.0927, r = 0.0287, C = 25.4461, R^2 = 0.99. \]

\[ M(t) = \frac{K}{1 + Ce^{-r}} \approx 2.0927 \]  

(18)

The coefficients of determinants \( R^2 \) in both models above (0.96 and 0.99) confirm that the Logistic model is appropriate for the dynamics of both fitness and fitness aggregation.

The above example reveals a very interesting and potentially very useful strategy in controlling the number of generations. By reducing the unnecessary generations iterated in GA search, the savings from fitness evaluations can be significant. It is interesting to note, that we were initially motivated to seek the clues for controlling population size, but here, the results seem to suggest that a potentially rewarding strategy is to control the number of generations iterated in GA searches. Of course, our two motivations and corresponding explorations—controlling population sizes and the number of generations—have the same objective: to improve the search efficiency of GA populations.

We raise the following questions for future studies:

Is it possible to predict the reflection point of the Logistic curve? We are cautiously optimistic for this possibility due to at least two observations. Firstly, the reflection points of both fitness aggregation and fitness seem to occur synchronously, corresponding to the same X-axis (generation), as also discussed in the previous paragraph. Secondly, the phenomenon that both the fitness aggregation and fitness dynamics can be described with the Logistic curve is a great advantage. The parameter \( K \) of the Logistic model is the asymptotical maximum the function can achieve; this is called environment capacity in biology. Although the exact prediction of \( K \) may not be possible, an approximation of \( K \) in the early stages of GA search, which is of significant value in controlling the numbers of generations iterated, is possible. To achieve such an approximation, we outline the following approach. Notice that the following constraints information is available: the early stage of the Logistic curve may be approximated with an exponential curve, which can be determined by a single parameter, say, \( r_m \). It is well-known that the Logistic curve is a non-linear extension of the exponential growth curve, and therefore, \( r_m \) is the maximum value the parameter \( r \) may achieve in the Logistic model. Obviously, in most cases, the maximum values of \( K \) can be predicted, at least for the fitness. For example, the absolute maximum (\( K_m \)) for \( K \) in the fitness model is 4 in Fig. 8. With the constraints set by \( r_m \) and \( K_m \), it is reasonable to set a meaningful upper bound for the number of generations at which the Logistic curves reach the reflection point. We call this generation \( t_m \), or the optimum generations to be iterated in the GA search.

Although the investigation of fitness aggregation dynamics performed in this section is based on the chaotic populations, a comprehensive study of the relationship between controlling populations (based on fitness aggregation dynamics) and controlling population sizes (with chaotic populations) should be rewarding.

2. The second example and discussion – In this example, we compare two populations: the fixed-size population with \( (B = 24, MaxSize = 2048) \) and corresponding chaotic population with \( (B = 24, MaxSize = 2048, ActualSize = 1327, \alpha = 3.57) \). Both treatments have been tested in previous sections (Table 2 and Table 5), but here, we take a broader view of the results and try to discover their difference in search process. Table 2 shows that this fixed-size population failed to find optimum solution, and the number of fitness evaluation experienced by this population should be 2,048,000 (2048 \times 1000 generations, Table 1). Table 5 shows that the chaotic population with comparable settings found the optimum solution after 1,320,683 fitness evaluations.

The following Fig. 9a shows the fitness progressions in both populations up to 1000 generations; they diverge quickly after approximately 250 generations. The chaotic population ultimately reached an average fitness \( M \approx 0.62 \) and the optimum solution was found once. The fixed-size population only reached a mean population fitness of \( M = 0.20 \) and never hits the optimum solution.

Fig. 9b shows the fitness progressions in both populations up to 2000 generations; they diverge quickly after approximately 250 generations. The chaotic population ultimately reached an average fitness \( M = 0.98 \) and the optimum solution was found twice. The fixed-size population only reached a mean population fitness of \( M \approx 0.58 \) and never hits the optimum solution.

As a side note, it should be interesting to observe the GA population dynamics (progression) over generation, which is controlled by chaotic dynamic model in this study. Fig. 14 shows the population dynamics (progression) over time (generation) up to 1000 generations when the parameter \( \alpha = 3.57 \), which represents the state of the “onset of chaos” – portion of the bifurcation diagram shown in Fig. 13. Fig. 13 shows the full spectrum of bifurcation process (including chaos) with various parameter values in the interval of \( 1 < \alpha < 4 \) in a single graph.

The dynamic relationships among population mean fitness \( (M) \) and mean fitness crowding \( (M') \) and fitness variance \( (V) \) in the chaotic and fixed-size populations are illustrated in Figs. 10 and 11, respectively. It is noted that both the mean fitness crowding \( (M') \) and variance of fitness measure the heterogeneity (or variability) of the fitness. Besides the difference that the mean fitness \( (M) \) in
Fig. 9. (a). Mean fitness vs. generations for chaotic population with \( (B = 24, \text{MaxPSize} = 2048, \text{Actual PSize} = 1327, \alpha = 3.57) \) and fixed-size population with \( (B = 24, \text{PSize} = 2048) \), maximum generation = 1000, the optimum solution was found once with the chaotic dynamic population, but the fixed-size population failed to find the optimum solution. (b). Mean fitness vs. generations for chaotic population up to 1000 generations with \( (B = 24, \text{MaxPSize} = 2048, \text{Actual PSize} = 1327, \alpha = 3.57) \) and fixed-size population with \( (B = 24, \text{PSize} = 2048) \). The optimum solution was found twice with the dynamic population, but the fixed-size population failed to find the optimum solution.

 chaotic population is three times that of the fixed-size population, the trajectory of fitness in the chaotic population is more like a “bamboo stem” pointing to the “target” of the maximum fitness, but in the fixed size population, the trajectory is more like “a tree with heavy roots”. The heavy roots of the tree might correspond to local optimums, which consumed significant search resources but do not necessarily contribute much to the global optimum search. In contrast, it appears that there is a small jump (the break band in the trajectory) near the “floor” in the chaotic population. The jump itself is small, but sufficient enough to escape a potentially local trap, which might have contributed to the “take-off” of the fitness trajectory.

 The figures demonstrated that the fitness aggregation is necessary for the “climbing” of the fitness and the spectrum of the aggregations measured in both metrics (\( M' \) and \( V \)) in chaotic population are much wider than those in the fixed-size population.

 The reason we observe both metrics (\( M' \) and \( V \)) is due to another conjecture, which is also inspired by their counterparts in biology, that is, \( M' \) seems a better metric for the local crowding, and \( V \) seems to reflect more about the global heterogeneity.

 In the last figure for this second example (Fig. 12), the relationship between mean fitness (\( M \)), population size and the fitness aggregation index \( M'/M \) is drawn for the chaotic population. What is somewhat surprising to us, but not unexpected, is that there are multiple trajectories formed around several population sizes. Five clusters (consisting of at least 8 “stripes”) of trajectories are visible in the graph. This seems to suggest a very interesting explanation for the effects of chaotic populations that Chaos helps the population to explore on “multiple frontlines” or take advantages of local heterogeneities. This is indeed consistent with the effects of Chaos in natural populations, where it is believed that Chaos promotes a species’ fitness by more effectively exploring/exploiting local heterogeneities.

 We do not include the counterpart figure to Fig. 12 for the fixed-size population for obvious reason, since there is only one population regime and the graph can have only one of the trajectories in Fig. 12. From Fig. 9, it is clear that the maximum fitness for the single trajectory will be around 0.2.

 Fig. 12 also bring about a broader point, which is also echoed in the very preliminary experiment of this study (Ma and Kriins [26]), in which only a few simple ad hoc dynamic populations schemes,
without adopting any mathematical models, were tested. The point is that the fluctuating populations may simply push the “bounds” tighter. In other words, the traditional fixed size populations simply waste resources and new dynamic chaotic populations are more efficient by adjusting population sizes. This study with chaotic populations further confirmed the point. Therefore, the population sizing problem is essentially the optimization of resource allocation; obviously, dynamic populations are more efficient in allocating resources.

This study also shows two other critical issues associated with population sizing problem: one is the control of the generations iterated in GA, and the other is the relationship between population sizing and the controlling of the number of generations. The former problem is illustrated with the first example in this section with very preliminary results. Both problems should be explored further in future studies.

6. Summary

This topic has accomplished two objectives: (1) emulating chaotic natural populations in GA populations by utilizing the Logistic Chaos map model; (2) analyzing the population fitness distribution by utilizing insect spatial distribution theory.

Inspired by the fact that natural populations are dynamic systems with complex nonlinear regulation (control) mechanisms, we have introduced the Logistic Chaos map model to GA populations. The Logistic map model is of remarkably rich behaviors but the model structure is very simple, which makes it ideal to emulate natural populations for GA algorithms. We have tested the hypothesis whether the dynamic GA populations controlled by Logistic chaos map would carry an advantage over the traditional fixed-size populations. We have tested key threshold values of the Logistic Chaos model, and found that dynamic GA populations controlled by the Chaos model indeed outperform the traditional populations in terms of fitness evaluation indexes as well as the frequency (hits) to detect the optimum solution. It is interesting to note that the so-called “onset of chaos” threshold \( \alpha = 3.57 \) seems to perform best overall. This suggests that further studies on the relationship between the Logistic Chaos parameter and key GA features as well as problem complexity should be rewarding.

Inspired by the spatial distribution pattern theory of insect populations, we envision an analogical mapping between the fitness distributions (aggregation) in GA populations to the spatial distributions (aggregation) of insect populations. The spatial distribution of insect population is an emergent behavior (at population level) that is fine-tuned by natural selection to facilitate the maximum adaptation to the environment heterogeneity. Individuals in a population constantly redistribute themselves in their habitat space as a response to the stimuli from environment or its peers and this lead to the generally aggregated distribution, which is known to follow Power Law since 1960s. We have borrowed the models from insect spatial distribution theory and applied them to the analysis of GA fitness distributions. The results from this analysis show that:

(i) Power Law and to a lesser extent, the meaning fitness crowding model, universally fit to the fitness distributions in GA extremely well. (ii) Although there seems no apparent difference between the chaotic and fixed-size populations with regard to the fitting of Power Law and mean crowding models, which suggests that fitness aggregation is a universal property of GA populations just like in natural populations, we do not exclude the possibility that Chaos may affect the fitness distribution in GA in more dynamic ways. Our argument is based on the fact that Power Law and mean crowding model parameters are largely the summary statistics of the GA fitness distribution in aftermath of the GA search completion. Although the models are fitted with the same GA program, it is essentially an offline-analysis of the fitness distribution. We have therefore suggested investigating the dynamics of fitness aggregation across generations during the GA search. (iii) We have adopted two aggregation index \( M/M^* \) and \( V/M \) to characterize the dynamics of fitness aggregation in GA. Due to the potentially extensive experiments involved, we only used two examples to illustrate the follow-up research problems. With two examples, we show the potential of the following three research problems: (1) Both the dynamics of fitness aggregation \( M/M^* \) and fitness \( M \) follow the Logistic growth curve and both Logistic curves seem synchronized. It should be possible to estimate the upper bounds of reflection points from the problem definition before the GA search start and/or from the feedback information in the early stage of the GA search. The minimum generations corresponding to the reflection points seem to be the minimum generations associated with the detection of the optimum solution. This implies that it is possible to predict the optimum (minimum) generations required to obtain the optimum solution. (2) The population sizing (per generation) and the number of generations in a GA search are most likely dependent. The capturing of this dependent relationship is obviously of more practical significance. (3) The third interesting research problem should be the phenomenon revealed by Fig. 12, the multiple clusters of fitness trajectories surrounding various population sizes in chaotic populations. This may not only explain why chaotic populations are advantageous over the fixed size populations, but also offer insights for determining the most appropriate values of the Logistic map parameter for a specific problem.

Acknowledgements

This research has received funding from the following funding sources: National Science Foundation of China (Grant No: 61175071), “The One Hundred-Pl (Principal Investigator) Plan for the Exceptional Talents in Science & Technology” of the Chinese Academy of Sciences, the startup fund from the State Key Laboratory of Genetic Resources and Evolution of China, and “Funding for the Top Talents Program in Science and Technology of Yunnan Province.”
Appendix A.

The following Figs. 13 and 14 and Table 11 are referred to in Section II: Ecological Theory.

A. Logistic Chaos map. Figs. 13 and 14.

Fig. 13. The logistic map bifurcation diagram, displaying chaotic behavior past various thresholds. In particular, \( a = 3.57 \) signals the “Onset of Chaos”.

Adapted from Wikipedia with minor changes, reuse permission granted.

Fig. 14. Chaotic population dynamics over GA generation with Logistic chaos model (\( \alpha = 3.57 \)).

B. Spatial distribution patterns. Fig. 15

Fig. 15. Distribution patterns of animal populations. Regular (a), Random (b), and Aggregated (c).

C. Reinterpreted Taylor’s Power Law. Table 11

Table 11: Reinterpreted Taylor’s Power Law by Ma [21,22,24].

<table>
<thead>
<tr>
<th>Parameters, ( a, b, m_0 )</th>
<th>Spatial distributions</th>
<th>Aggregation mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b = 1 )</td>
<td>( 1 &lt; a &lt; 1 )</td>
<td>Regular</td>
</tr>
<tr>
<td></td>
<td>( a = 1 )</td>
<td>Random</td>
</tr>
<tr>
<td></td>
<td>( a &gt; 1 )</td>
<td>Aggregated</td>
</tr>
<tr>
<td>( m &gt; m_0 )</td>
<td></td>
<td>Independent (ADI)</td>
</tr>
<tr>
<td>( b &gt; 1 )</td>
<td>( m = m_0 )</td>
<td>Random</td>
</tr>
<tr>
<td></td>
<td>( m &lt; m_0 )</td>
<td>Regular</td>
</tr>
<tr>
<td></td>
<td>( m = m_0 )</td>
<td>Aggregated</td>
</tr>
<tr>
<td>( b &lt; 1 )</td>
<td>( m &gt; m_0 )</td>
<td>Inversely density-</td>
</tr>
<tr>
<td></td>
<td>( m = m_0 )</td>
<td>dependent (ADID)</td>
</tr>
<tr>
<td></td>
<td>( m &lt; m_0 )</td>
<td></td>
</tr>
</tbody>
</table>

References