Analysis on the Collaboration Between Global Search and Local Search in Memetic Computation

Jih-Yiing Lin and Ying-Ping Chen, *Member, IEEE*

***Abstract*—The synergy between exploration and exploitation has been a prominent issue in optimization. The rise of memetic algorithms, a category of optimization techniques which feature the explicit exploration-exploitation coordination, much accentuates this issue. While memetic algorithms have achieved remarkable success in a wide range of real-world applications, the key to successful exploration-exploitation synergies still remains obscure as conclusions drawn from empirical results or theoretical derivations are usually quite algorithm specific and/or problem dependent. This paper aims to provide a theoretical model that can depict the collaboration between global search and local search in memetic computation on a broad class of objective functions. In the proposed model, the interaction between global search and local search creates a set of local search zones, in which the global optimal points reside, within the search space. Based on such a concept, the quasi-basin class (QBC) which categorizes problems according to the distribution of their local search zones is adopted. The subthreshold seeker, taken as a representative archetype of memetic algorithms, is analyzed on various QBCs to develop a general model for memetic algorithms. As the proposed model not only well describes the expected time for a simple memetic algorithm to find the optimal point on different QBCs but also consists with the observations made in previous studies in the literature, the proposed model may reveal important insights to the design of memetic algorithms in general.**

***Index Terms*—Global search, local search, memetic algorithms, quasi-basin class, subthreshold seeker.**

I. Introduction

OPTIMIZATION, finding the optimal element among a set of feasible ones, is a type of problem commonly encountered in many fields. Many real-world and theoretical problems can be formulated as optimization problems and solved by applying or developing various optimization techniques. Early optimization techniques, such as Newton's method, simplex method, conjugate gradient algorithm, and the like, have been well developed on problems with certain mathematical characteristics. However, as many real-world optimization problems are black-box problems of which *a priori* problem knowledge is not available, the use of meta-heuristics started to prevail. Meta-heuristics are generally population-based algorithms which explore the search space stochastically according to some heuristics. As they are not problem-specific, they have a good chance to perform well on black-box optimization problems. Evolutionary algorithms, particle swarm optimizations, ant colony algorithms, and the like, are some of the renowned meta-heuristics which have been widely adopted.

Manuscript received April 30, 2010; revised September 29, 2010 and March 11, 2011; April 19, 2011. Date of current version September 30, 2011. This work was supported in part by the National Science Council of Taiwan, under Grant NSC 99-2221-E-009-123-MY2.

The authors are with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: jylin@nclab.tw; ypchen@nclab.tw).

Digital Object Identifier 10.1109/TEVC.2011.2150754

The generality of meta-heuristics which provides the wide applicability also limits the efficiency of meta-heuristics. When complicated problems are encountered, without taking advantages of problem-specific information given *a priori* or retrieved during optimization, meta-heuristics can merely deliver mediocre performance. As problem-specific heuristics can generally take advantages of problem-specific information, techniques that hybrid general meta-heuristics and problem-specific heuristics have been developed to provide more efficient optimization techniques for more complicated problems. These techniques which employ general meta-heuristics as global search and problem-specific heuristic as local search are commonly referred to as memetic algorithms (MAs). With an appropriate coordination, memetic algorithms cannot only exhibit a good explorative ability as a population-based global search algorithm does but also deliver a good exploitive performance as a local search algorithm does. As a result, memetic algorithms perform better than pure population-based global search algorithms or stand-alone local search algorithms. As the research interests and activities of memetic algorithms thrive, memetic computing has been evolved from hybridization of global search and local search to hybridization with adaptation and has the potential to be applied to computational intelligence [1]. Manifold of successful memetic algorithms in various application domains, ranging from NP-hard combinatorial problems to non-linear programming problems, have been reported [2]. Besides the various application domains mentioned in [2], recent memetic algorithm applications in Cartesian robot control [3], e-learning systems [4], image segmentation [5], feature selection [6], mission management [7], and portfolio selection [8] also demonstrate the efficacy of memetic algorithms in different application domains.

Among these memetic algorithms, in addition to the selection of the global search component and the local search operator, the synergy between global search and local search has always been one of the key design issues. The design of most memetic algorithms follows the seminal studies on memetic algorithms proposed in [9] and [10]. In these studies, the authors observed that memetic algorithms favor infrequent starts and long running time of local search. They also proposed several renowned strategies for selecting solution candidates on which the local search operator is applied: the fitness based selection and the diversity based selection. However, with the aids of these guidelines, designing a memetic algorithm for a specific problem still requires considerable time as the optimal design is not only algorithm specific but also problem dependent. To cope with this issue, the concept of systematically adjusting the parameters of local search is proposed [11]. Although this technique is robust, it does not guarantee the best performance. Another line of research is regarding the concept of memes [12]–[14]. In these studies, the local search algorithms, encoded as memes, can adapt to the underlying problem and thus improve the efficiency as the memetic algorithm progresses. This framework is robust as well as efficient with the expense of the learning cost of memes.

In spite of the light shed on the design issue of memetic algorithms by the aforementioned studies, the question of how one can achieve the optimal design of memetic algorithms on a specific problem remains. The key to achieve this ultimate goal apparently include a full awareness of the physics behind the algorithm and the problem. As theoretical studies can help to understand the internal mechanism of algorithms, they can provide important insights to the design issue. Compared to the progress of theoretical studies on evolutionary computation, which is still in its infancy [15]–[23], theoretical studies of memetic algorithms are even scarce. Recent studies [24], [25] investigated the behavior of simple memetic algorithms on several classes of functions. The proposed theoretical models on the demonstrative classes of functions reaffirmed that parameterizing memetic evolutionary algorithms can be extremely difficult. As these theoretical models are developed according to different classes of functions, they are capable of depicting the algorithmic behavior from their respective perspectives on the adopted classes of functions instead of providing a unified principle for the design of memetic algorithms.

The concept of basins of attraction [26] provides another perspective and gives an opportunity to conduct general analysis on memetic algorithms. In [27] and [28], the search space is viewed as a union of basins of attraction, and the optimal allowable local search length of simple memetic algorithms is theoretically estimated. A similar concept, quasi-basins defined by the subthreshold seeker, was adopted to prove the searchability of general functions [29] and to investigate the subthreshold seeking behavior [30].

In this paper, we aim to establish a theoretical model that can depict the collaboration between global search and local search in memetic computation on a wide range of problems. To achieve this, we propose the concept of local search zones which are the regions that local search exploits. In this perspective, these local search zones are defined by the landscape of the problem as well as the collaboration between global search and local search. As local search zones are generally not easy to assess, we adopt quasi-basins to estimate local search zones and define the quasi-basin class (QBC) which categorizes problems by their quasi-basin distributions as the basis on which memetic algorithms are investigated. Then, we analyze the performance of the subthreshold seeker, which is regarded as a representative archetype of memetic algorithms, to develop a theoretical model for the global-local search collaboration in memetic computation. The derived theoretical model can describe how the distribution of local search zones and the efficiency of the global search algorithm and the local search algorithm are related to the expected time for a memetic algorithm to find the optimal solution. Because this model, empirically verified, is consistent with the observations made in many previous studies in the literature, it may be considered valid for representing various memetic algorithms on a wide range of problems and may give important insights to the future design of advanced memetic algorithms.

The rest of this paper is arranged in the following manner. Section II gives a survey on the current progress of analysis on memetic algorithms and elaborates the need of a general theoretical model which can describe the collaboration between global search and local search in memetic computation on a broad range of problems. Section III expounds the fundamental concepts on the analysis of memetic algorithms and provides the definitions of our framework to form the basis for further derivation. As a memetic algorithm comprises global search and local search, we first analyze the global search component of the subthreshold seeker and discuss how this analysis is related to the behavior of common global search algorithms in Section IV. Based on the analysis of global search and the concept of QBC, we derive and empirically verify the formula that describes the behavior of the subthreshold seeker working with local search operators of different efficiency on various QBCs in Section V. After the empirically verifying the proposed model, we expound how our model can represent the general behavior of memetic algorithms and discuss possible extensions and future work of the proposed model in Section VI. Finally, we recap the significance of our model and conclude this paper in Section VII.

II. Background

Designing a memetic algorithm requires not only selecting a global search mechanism as well as a local search operators but also establishing a subtle coordination to exhibit the vantage of both ends. Hart [9] in his seminal study for designing efficient memetic algorithms investigated the following four questions on continuous optimization problems.

1) How often should local search be applied?
2) On which solutions should local search be used?
3) How long should local search be run?
4) How efficient does local search need to be?

In his framework, he noted that the memetic algorithms that employ elitism will be most efficient with large population sizes and infrequent local search. He also proposed two strategies, fitness based selection and diversity based selection, for selecting solution candidates to apply local search. He concluded that these two strategies help much. Land [10] extended Hart's study to combinatorial domains. In his study, he adopted steady state genetic algorithms as global search and proposed a local search potential based strategy in selecting local search candidates. The local search potential strategy turned out to be not very useful. Yet, he observed that his steady state memetic algorithm favored smaller rates and longer runtime for local search, consistent with Hart's study.

Although limited to specific problems, the studies of Hart and Land gave some insights to the first three questions and have inspired the successive memetic algorithms in a wide variety of applications. The concepts of selecting the best or some qualified individuals for local search which resemble the fitness based selection have been adopted in [31]–[33]. The steady state memetic algorithm with adaptive local search has been applied in [34]–[36], while other studies exhibit the vantage of utilizing the diversity information in their design of memetic algorithms [37], [38]. Investigations into the balance between global search and local search for some applications available in the literature also accord with Hart's and Land's observations [39], [40]. Despite that the accordance of these results reveals some essential design principles of efficient memetic algorithms, designing a memetic algorithm still requires a considerable amount of effort due to the lack of detailed knowledge on how the key mechanism of memetic algorithms, the synergy between global search and local search, working on the underlying problem. An interesting technique of adapting local search intensity in a simulated annealing way was proposed to cope with the MA parameterizing issue [11]. More robust than the fixed local search intensity setting, this method still requires a range setting and does not guarantee the best performance.

In addition to the parameterizing issue caused by using memetic algorithms to handle different problems, the efficiency of a local search operator is particularly problem dependent. [41] provided a landscape analysis for memetic algorithms. Following this, the concept of memes [12]–[14] was proposed. In these frameworks, the local search component is designed to adapt to the underlying problem as the optimization progresses. These meme evolving or learning memetic algorithms are robust regardless of the underlying problem and efficient. Recent studies [42], [43] have also proposed several metrics to assess the improvement of applying a local search algorithm on a problem.

Furthermore, theoretical analysis has always been a prevalent way to provide clues to the design of algorithms. For continuous problems, convergence analysis is widely adopted in performance assessment for evolutionary computation [16], [19], [22]. For discrete problems, the (1+1) evolutionary algorithm (EA) has been widely adopted in theoretical analysis on evolutionary algorithms [15], [17], [18], [20], [21], [23]. The (1+1)-EA is a rather simple algorithm with one individual and an evolutionary operator flipping each bit of the individual with a uniform probability. Following these studies, the theoretical analysis of memetic algorithms starts from the (1+1)-MA and goes to the (μ+λ)-MA [24], [25]. On three discrete functions, Sudholt investigated the behavior of the (1+1)-MA and the (μ+λ)-MA, and these studies reaffirmed the parameterizing of memetic algorithms is extremely hard.

Theoretical models developed in this way are capable of providing different perspectives, according to the adopted classes of functions, to analyze a memetic algorithm. Reference [44] illustrated that different problems favor different population sizes, while [45] and [46], which investigated the effect of recombination operators, provided counter perspectives. The issue of such an analysis technique is that the derived theoretical behavior is naturally confined and largely determined by the adopted objective functions. As an undesirable result, the different conclusions obtained from various theoretical models cannot form a unified guideline to the design of algorithms.

Another line of analysis involves the concept of basins of attraction. The basin of attraction of a local optimum is the set of points in the search space such that a local search process starts from any member within a basin will eventually find the local optimum in that basin [26]. In this line of research, the search space is a union of basins of attraction. References [27] and [28] adopted this concept to estimate the optimal local search length. In these papers, basins of attraction in the search space are categorized into two types, in which target solutions can or cannot be reached. The optimal local search length is estimated via acquiring the probability of hitting the former basins.

A closely related concept, quasi-basin defined by subthreshold seeker, was introduced by [29] in investigating searchable functions in which the No Free Lunch theorem does not hold. The submedian seeker which starts local search when hitting a point with a submedian value and turns to do random search when hitting a point with a supermedian value was considered. By applying the submedian seeker to functions with a certain degree of self-similarity, that the functions exhibiting self-similarity are searchable was proved. Whiteley and Rowe further proposed the subthreshold seeker, a generalized 1-D submedian seeker, and investigated its seeking behavior [30]. In their work, the subthreshold seeking behavior, the ratio of the sampled subthreshold points to superthreshold points, was used as a performance index. Their theoretical analysis detailed the conditions under which the subthreshold seeker could outperform random search and showed that a higher bit-precision could improve the performance.

Finally, in this paper, we aim to provide a general model for the collaboration between global search and local search in memetic algorithms on a broad class of problems. The proposed model will describe how the expected performance of a memetic algorithm is related to the efficiency of the local search operator, the landscape of the problem, and the collaboration between global search and local search. In order to achieve this goal, we propose the concept of local search zones. Local search zones are the regions which local search prefers and the global optimal point resides in. As generally local search zones are not easy to assess, we adopt the idea of quasi-basins to estimate local search zones and define the QBC to categorize problems according to their quasi-basin distributions. The subthreshold seeker, taken as a representative archetype of memetic algorithms, is analyzed over different QBCs as a general theoretical model for memetic algorithms. Thus, the proposed model can depict the essence of the collaboration between global search and local search in memetic algorithms on various problems and may shed light on the design of memetic algorithms.

III. Quasi-Basin Classes and Subthreshold Seeker

In this section, we introduce the concept of local search zones and give definitions to the fundamental terminologies of our framework. The concept of the local search zones is described based on the formal definitions of the search process of an algorithm on a problem and the search space viewed by a search process. Then, based on the concept of local search zones, we introduce the QBC and the generalized subthreshold seeker on which the theoretical analysis is based.

A. Local Search Zones

The task to handle an optimization problem is to optimize a given objective function $f : \mathcal{X} \to \mathcal{Y}$. For convenience, we specify our optimization goal as to find a point $x^* \in \mathcal{X}$ with the minimum value $y^* \in \mathcal{Y}$. We assume that both $\mathcal{X}$ and $\mathcal{Y}$ are finite sets. Such an assumption makes a practical sense because optimization problems are generally numerically solved on digital computers. In this paper, for simplifying the derivation, we also assume that every function maps different $x \in \mathcal{X}$ to different $y \in \mathcal{Y}$. In order to formally describe a search process of an algorithm on a function, we adopt part of the terminologies defined in [47] as the following definitions.

Definition 1 (Search Process): Given two finite sets $\mathcal{X}$ and $\mathcal{Y}$:

1) A trace of length m is a sequence $T_m := ((x_i, y_i))_1^m = ((x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$ with distinct x_i. "$x \in T_m$" denotes that $x = x_i$ for some $i \in \{1, 2, \dots, m\}$. Let T_0 be the empty sequence and $\mathcal{T}^\ell$ be the set containing all the traces of a length smaller than or equal to ℓ.
2) Let A_T, where $T \in \mathcal{T}^{|\mathcal{X}|-1}$, be a random variable over $\mathcal{X}$ satisfying that $Prob\{A_T = x\} = 0$ for all $x \in T$. An algorithm A is a collection of such random variables, i.e., $A = \{A_T \mid T \in \mathcal{T}^{|\mathcal{X}|-1}\}$.
3) The search process of A on f, $S(A, f)$, is a stochastic process $(X_i, Y_i := f(X_i))$ over $\mathcal{X} \times \mathcal{Y}$ defined by $X_1 \sim A_{T_0}$ and $X_{k+1} \sim A_{(X_i, Y_i)_1^k}$.

For generality, we interpret the search space viewed by a memetic algorithm as a graph. Since a local search algorithm usually starts from a candidate solution and iteratively moves to a neighbor solution, the local search algorithm defines the neighborhood of a candidate solution in the search space viewed by the memetic algorithm which utilizes it. Thus, we define the search space viewed by a memetic algorithm as a graph of which the vertices are the set of points of $\mathcal{X}$ and the edges are the set of pairs of points connected by the local search algorithm of the memetic algorithm as follows.

Definition 2 (Search Space): Given a memetic algorithm MA, a function f, and LS, the local search algorithm adopted by MA. Let $N_{LS}(v)$ denote the neighborhood of a vertex v defined by LS. The search space viewed by MA on f can be represented by a graph $G = (V, E)$, where $V(G) := \mathcal{X}$ and $E(G) := \{\langle v_i, v_j\rangle \mid v_j \in N_{LS}(v_i), \forall i, j\}$.

In the rest of this paper, the terms $\mathcal{X}$ and $V(G)$ are used exchangeably. Now, with all these fundamental terminologies, we can formally define the local search zone as follows.

Definition 3 (Local Search Zone): Given a search process of a memetic algorithm MA on function f, $S(MA, f)$, and LS, the local search algorithm adopted by MA:

1) The local search points of a search space G viewed by $S(MA, f)$ are defined as the set of points $S_{LSZ} = \{v \mid E[Pr(X_{k+1} = u, u \in N_{LS}(v) | v \in T_k, u \notin T_k)] > 0.5, \forall v \in V(G)\}$.[1]
2) A local search zone LSZ is defined as a maximal subset in S_{LSZ} such that there exists a path[2] between all the pairs of vertices.
3) The size of local search zones is denoted by $|S_{LSZ}|$.

By this definition, a local search point is a vertex which if is visited by MA, MA would tend to visit one of its unvisited adjacent vertices in the future, and the local search zones are where the local search points reside. In other words, the local search zones are where local search prefers when a memetic algorithm is applied. In our perspective, the distribution of the local search zones has a great influence on the performance of a memetic algorithm. As the local search of practical memetic algorithms favors the points that have high potential to lead to the optimal point, fitness-relevant and diversity-relevant criteria are adopted. These criteria are somehow dynamic, complicated, and difficult to analyze. Abstracting the exploration behavior of global search and the exploitation behavior of local search, we consider fitness as the prime index of the potential to find the optimal point regardless of the diversity-relevant metrics which are often auxiliary for diversity maintenance. Thus, the local search zones can be estimated by zones consisting of qualified high-fitness points. Based on this way of thinking, we define the QBC to represent different problem classes which possess different local search zone distribution. Then, we take the subthreshold seeker as a representative archetype of memetic algorithms and analyze its behavior on various QBCs to develop a general theoretical model for the core mechanism of memetic algorithms.

B. Quasi-Basin Classes

The QBC conceptually defines problem classes according to the number of local search zones and the size of local search zones. To define QBC, we first define the quasi-basin (QB) as follows.

Definition 4 (QB):

1) For any function f, function value $\beta_m(f)$, defined as $\beta_m(f) := \min\left\{\arg_y \{|\{x \in \mathcal{X} \mid f(x) \leq y\}| = m\}\right\}$, denotes a threshold, and there are $m - 1$ points with an objective value less than $\beta_m(f)$.
2) For any function f, the set that contains all the points with an objective value less than $\beta_m(f)$ is defined as $S_m(f) := \{x \in \mathcal{X} | f(x) \leq \beta_m(f)\}$.
3) Given a graph G, for any function $f : V(G) \to \mathcal{Y}$, a quasi-basin QB is defined as a maximal subset in $S_m(f)$ such that there is a path between all the pairs of vertices.

[1]The $E[]$ notation indicates the expected value of $Pr(X_{k+1} = u, u \in N_{LS}(v) | v \in T_k, u \notin T_k)$ over all $k \in |\mathcal{X}|$ and all possible T_k which containing v and not containing u.

[2]A path in a graph is a sequence of vertices such that from each vertex there exists an edge to the next vertex in the sequence except for the last one.

As generally the points residing in quasi-basins are better than the other points in the search space and are favored by fitness-relevant local search criterion, $S_m(f)$ conceptually estimates the set of points residing in the local search zones with a size m while a quasi-basin QB can be regarded as a local search zone in the search space. Based on the fundamental definitions, we define the QBC and the uniform quasi-basin class (uQBC) in Definitions 5 and 6.

Definition 5 (QBC): Given a graph G and a co-domain $\mathcal{Y}$, the corresponding discrete QBC with b distinct quasi-basins and m subthreshold vertices is defined as

$$\mathcal{Q}(G, \mathcal{Y}, m, b) := \{f : V(G) \to \mathcal{Y} \mid S_m(f) = \bigcup_{i=1}^{b} QB_i, \bigcap_{i=1}^{b} QB_i = \emptyset, |QB_i| \geq 1, 1 \leq i \leq b\} .$$

Definition 6 (uQBC): Given a graph G and a co-domain $\mathcal{Y}$, the corresponding uniform discrete QBC with b distinct quasi-basins and m subthreshold vertices is defined as

$$\mathcal{Q}_u(G, \mathcal{Y}, m, b) := \{f : V(G) \to \mathcal{Y} \mid S_m(f) = \bigcup_{i=1}^{b} QB_i, \bigcap_{i=1}^{b} QB_i = \emptyset, \left\lfloor \frac{m}{b} \right\rfloor \leq |QB_i| \leq \left\lceil \frac{m}{b} \right\rceil, 1 \leq i \leq b\} .$$

The QBC defines a class of problems with the points of m smallest function values distributed among b distinct quasi-basins. Thus, we can categorize problems according to their distribution of quasi-basins conceptually mapping to the distribution of local search zones. The uniform QBC further restricts the sizes of quasi-basins to be uniform. Note that m is naturally restricted to be less than or equal to $|\mathcal{X}|$ and greater than or equal to b. b is a positive integer which is less than the minimum of m and $|\mathcal{X}| - m$.

C. Subthreshold Seeker

Global search and local search of the subthreshold seeker are coordinated by the threshold θ. A subthreshold seeker globally searches by sampling the space uniformly at random (u.a.r.) until it encounters a subthreshold point, a point with a function value lower than the threshold. When this event occurs, the subthreshold seeker starts local search to exploit the quasi-basin, a maximal set of connected subthreshold points, in which the subthreshold point resides. The local search part keeps visiting the neighbors of the current vertex until it could no longer walk on a subthreshold vertex. After local search in the quasi-basin is done, the subthreshold seeker continues global search until another subthreshold pointer is encountered. The subthreshold seeker will continue switching between global search and local search until the stopping criterion is met. Fig. 1 illustrates how the subthreshold seeker proceeds. In Whitley and Rowe's work, their subthreshold seeker was applicable only to 1-D functions. In our present work, we generalize their subthreshold seeker to more dimensions by utilizing the graph representation in the definition of search space for more general applications.

```
procedure SUBTHRESHOLD-SEEKER(X, Y, N : X → 2^X, f : X → Y, θ, s)
    while the stopping criterion is not satisfied do
        if Queue is not empty then
            x ← Queue.pop();
        else
            Select x from X u.a.r.
        end if
        if f(x) ≤ θ then
            Queue.push(N_s(x))
        end if
    end while
end procedure
```

Fig. 1. Generalized subthreshold seeker.

The notation $N_s(x)$ denotes the set of virtual neighbors of vertex x defined by the local search parameter s. When $s = 1$, all the neighbors of vertex x in G defined by the local search operator will be visited. In other words, for $N_1(x)$, all the points in the quasi-basin under local search will be eventually visited. In the rest of this paper, we refer to the exhaustive local search as the local search with the local search parameter $s = 1$. When $s > 1$, only $1/s$ of the points in the quasi-basin will be visited in one local search run. To simulate the effect of this parameter, we manipulate the local search to have a step size s. Thus, the virtual neighbors of vertex x are those vertices who are s distance away from x. Here, s distance refers to the length of a path consisting s edges on the graph. Note that this subthreshold seeker does not sample visited points to avoid the performance declination caused by repeated sampling.

IV. Stochastic Global Search Time

Before we start to analyze the collaboration between global search and local search in the subthreshold seeker, we first investigate the behavior of the global search part employed by the subthreshold seeker. In this section, we theoretically and empirically analyze the behavior of the global search part with respect to the number of subthreshold points m and the number of quasi-basins b. We will derive the expected number of visited points required by the random search, the global search part, to find the first subthreshold point. The expected number of visited points is referred to as the expected first global search time $E(T_\theta)$, where θ is $\beta_m(f)$. Then, we will further approximate the expected kth global search time. The theoretical results will be empirically verified, and the discussion on the implication of the model will be presented.

A. First Global Search Time

In this section, we estimate the expected number of visited points for global search to find the first point x with $f(x) \leq \theta$, referred to as the first global search time T_θ. The first global search time can be interpreted in the following manner. Since $\theta = \beta_m(f)$, the number of points with their function values less than or equal to θ is m. Let N be the size of $\mathcal{X}$. As global search is uniform random sampling without repetition, the search space is of size N and contains m desired points, the probability for q visited points to contain exactly one subthreshold point follows the hypergeometric distribution with parameters N, m, and q

$$P(X = 1; N, m, q) = \frac{\binom{m}{1}\binom{N-m}{q-1}}{\binom{N}{q}} .$$

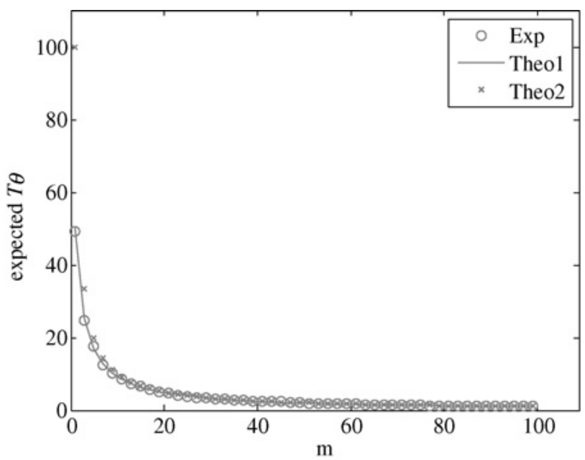

Fig. 2. Expected T_θ with respect to m when $N = 100$. Exp represents the actual average T_θ over 1000 independent simulation runs. Theo1 represents the theoretical expected T_θ of non-repeated uniform random sampling, and Theo2 represents the theoretical expected T_θ of uniform random sampling that allows us to sample visited points.

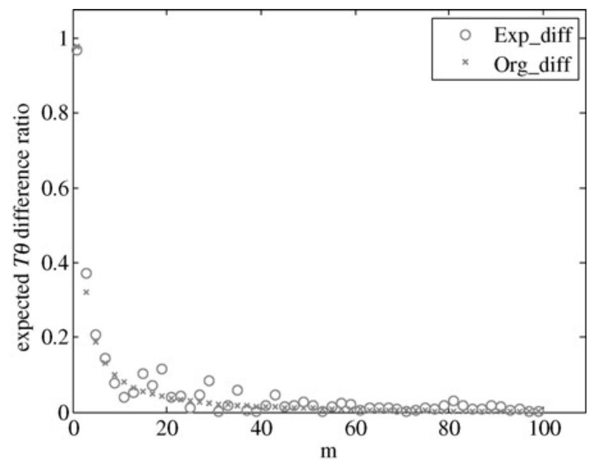

Fig. 3. Difference ratio of the expected T_θ with respect to m when $N = 100$. The Exp_diff represents the difference ratio between the actual average T_θ and N/m. The Org_diff represents the difference ratio between the theoretical expected T_θ and N/m.

The probability to hit a subthreshold point at the qth visited points is therefore

$$\frac{1}{q}P(X = 1; N, m, q) .$$

Let $E(T_\theta)$ be the expected first global search time. We have

$$\begin{aligned} E(T_\theta) &= \sum_{i=1}^{N-m+1} i\,\frac{1}{i} P(X = 1; N, m, i) \\ &= \sum_{i=1}^{N-m+1} \frac{\binom{m}{1}\binom{N-m}{i-1}}{\binom{N}{i}} \\ &= m \sum_{i=1}^{N-m+1} \frac{i}{N} \prod_{j=0}^{i-2} \frac{N-m-j}{N-1-j} . \end{aligned} \tag{1}$$

Fig. 2 illustrates the expected value of T_θ with respect to m when $N = 100$. In this figure, we compare (1) (the solid line, Theo1) with N/m (the crosses, Theo2) and the average first global search time in 1000 independent simulation runs (the circles, Exp). The N/m is the expected T_θ for allowing sampling visited points which is obviously an upper bound of (1). In the figure, we can find that (1) consists of the empirical result perfectly, while the trend of N/m gradually converges toward the other two. For non-repeated random sampling, half of points in the search space are expected to be visited before finding the minimum point. As m increases, indicating that $S_m(f)$ contains more points, time to meet a point in $S_m(f)$ decreases rapidly regardless of whether or not sampling visited points is allowed. It indicates that although finding several specific points in a search space via random search takes a considerable amount of time, finding a point in a small but large enough set can be attained within a relatively shorter time.

Fig. 3 illustrates the differences among the actual T_θ averaged over 1000 runs, and the two theoretical expected T_θ in ratio. The circle represents the difference between the empirical result and N/m in ratio with respect to the empirical result, and the cross represents that between (1) and N/m. From this figure we can find that (1) can be approximated by N/m as it only deviates significantly from (1) when m is rather small. As (1) is a complicated formula and difficult to analyze, we approximate the expected T_θ with N/m.

B. *k*th Global Search Time

In this section, we further measure the expected time for global search to find a subthreshold point after $k - 1$ runs of local search have been executed. In other words, we estimate the time for the kth global search. As the local search frequency and the global search frequency are related to the landscape of the problem, for simplicity, we derive the model on the uniform quasi-basin class $\mathcal{Q}_u(G, \mathcal{Y}, m, b)$. All the problems in this class have their quasi-basin sizes fixed to $\lfloor m/b \rfloor$ or $\lceil m/b \rceil$. Because each local search run in a quasi-basin will eventually visit about $1/s$ of the points in the quasi-basin, each local search run will visit $\lfloor m/bs \rfloor$ or $\lceil m/bs \rceil$ points. For convenience of derivation, we adopt m/bs instead of $\lfloor m/bs \rfloor$ or $\lceil m/bs \rceil$ for the number of points visited by a local search run. For non-revisit search, since the first global search time is approximated as N/m, when in the second global search run, there will be $N - N/m - m/bs$ unvisited points and $m - m/bs$ unvisited subthreshold points, the time required for the second global search run is

$$\frac{N - \frac{N}{m} - \frac{m}{bs}}{m - \frac{m}{bs}} .$$

The ith global search time is denoted as F_i, where i is referred to as the number of global search runs. We have

$$F_1 = \frac{N}{m}$$

$$F_2 = \frac{N - F_1 - \frac{m}{bs}}{m - \frac{m}{bs}}$$

$$F_3 = \frac{N - (F_1 + F_2) - \frac{2m}{bs}}{m - \frac{2m}{bs}}$$

$$F_k = \frac{N - \sum_{i=1}^{k-1} F_i - \dfrac{(k-1)m}{bs}}{m - \dfrac{(k-1)m}{bs}} \ . \tag{2}$$

Fig. 4 illustrates the global search time, estimated by (2), with respect to the number of global search runs for different distributions of quasi-basins. Fig. 4, with $N = 1000$, $m = 10$, $b = 10$, and $s = 1$, represents a case of a scarce small quasi-basin distribution. In this case, the global search time is large and does not change much as the number of global search runs increases. The second case illustrated in Fig. 4 represents a case of a scarce large quasi-basin distribution with $N = 1000$, $m = 900$, $b = 10$, and $s = 1$. The global search time is small and slightly increases as the number of runs increases. The last case illustrated in Fig. 4 represents a case of fully uniform distributed quasi-basins with $N = 1000$, $m = 500$, $b = 500$, and $s = 1$. In this case, the global search time is also small and slightly decreases as the number of runs increases.

As the QBC only defines the number of subthreshold points and the number of quasi-basins, local search of the subthreshold seeker can be considered as stochastic non-repeated sampling in the set of subthreshold points. Since the minimum resides in $S_m(f)$ with m points, for the stochastic non-repeated sampling, it is expected to sample $(m+1)/2$ points before the minimum point can be found. In the uniform QBC, each quasi-basin is about the same size, $\lfloor m/b \rfloor$ or $\lceil m/b \rceil$, and a local search run visits $\lfloor m/bs \rfloor$ or $\lceil m/bs \rceil$ of the points in a quasi-basin. It is expected to require $k = \left\lceil \dfrac{(m+1)/2}{m/bs} \right\rceil \approx \lceil bs/2 \rceil$[3] local search runs to find the minimum, which implies $k = \lceil bs/2 \rceil$ global search runs are required. Thus, the final global search time is

$$F_f = \frac{N - \displaystyle\sum_{i=1}^{\lceil bs/2 \rceil - 1} F_i - \dfrac{(\lceil bs/2 \rceil - 1)m}{bs}}{m - \dfrac{(\lceil bs/2 \rceil - 1)m}{bs}} \ . \tag{3}$$

We are now ready to calculate the upper bound for the last global search run. When bs is even, the last global search run requires

$$\begin{aligned} F_f &< \frac{bsN - (\lceil bs/2 \rceil - 1)m}{bsm - (\lceil bs/2 \rceil - 1)m} \\ &= \left(\frac{2bs}{bs+2}\right)\frac{N}{m} - \left(\frac{bs-2}{bs+2}\right) \ . \end{aligned} \tag{4}$$

When bs is odd, the last global search run requires

$$\begin{aligned} F_f &< \frac{bsN - (\lceil bs/2 \rceil - 1)m}{bsm - (\lceil bs/2 \rceil - 1)m} \\ &= \left(\frac{2bs}{bs+1}\right)\frac{N}{m} - \left(\frac{bs-1}{bs+1}\right) \ . \end{aligned} \tag{5}$$

The lower bound for the last global search run can be derived as follows:

$$\begin{aligned} F_f &> \frac{N - \sum_{i=1}^{k-1} F_i}{m} \\ &> \frac{N - N/2}{m} = \frac{1}{2}\frac{N}{m} \ . \end{aligned} \tag{6}$$

Both (4) and (5) indicate that the final global search time would be no more than twice of the amount of the first global search. On the other hand, (6) shows that the final global

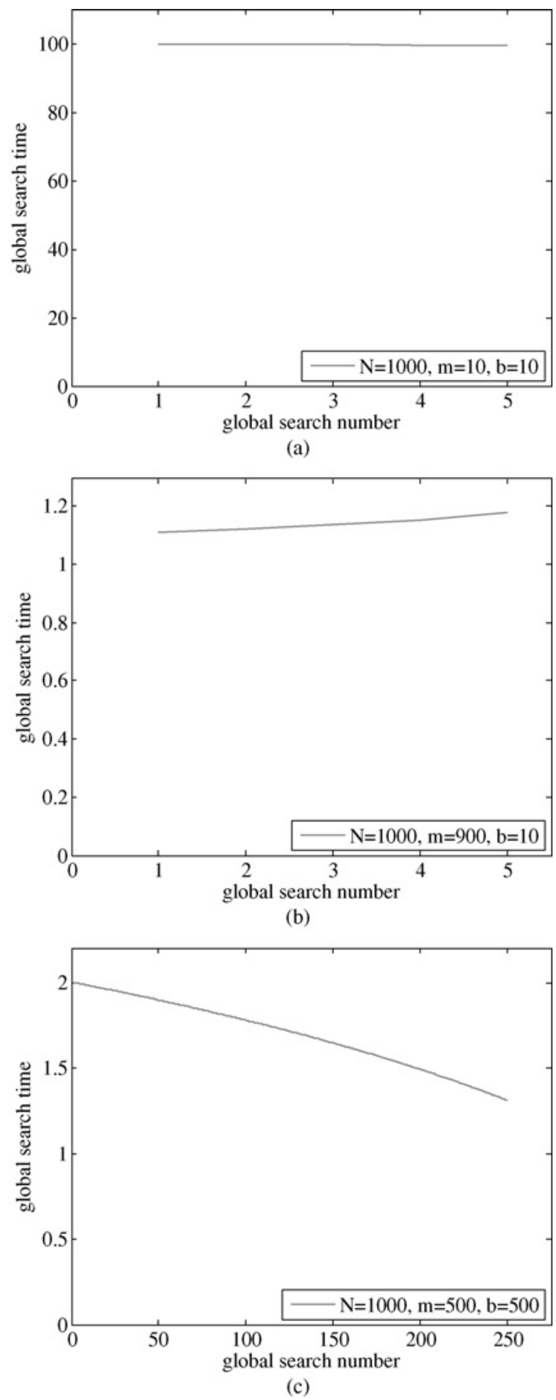

Fig. 4. Global search time with respect to the number of global search runs when the exhaustive local search is applied. (a) Case of a scarce small quasi-basin distribution. (b) Case of a scarce large quasi-basin distribution. (c) Case of fully uniform distributed quasi-basins.

[3]For simplicity, we approximate $(m+1)/2$ with $m/2$.

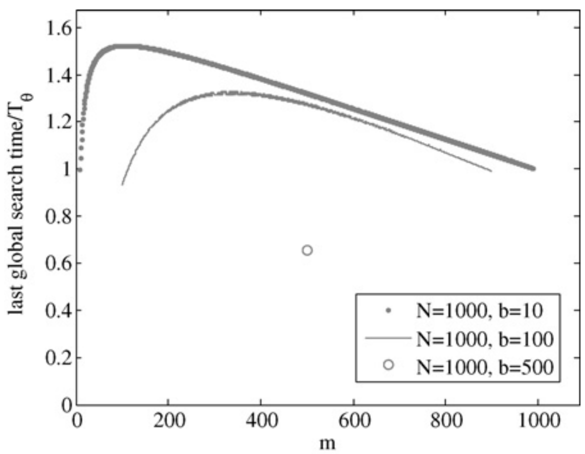

Fig. 5. Last global search time divided by the first global search time, T_θ, with respect to m when the exhaustive local search is applied.

search time would be greater than half of the amount of the first global search. Fig. 5 illustrates the last global search time, F_f, divided by the first global search time, T_θ, with respect to m when the exhaustive local search is applied. The last global search times of $b = 10$ and $b = 100$, initially increase as m increases and reach a peak, followed by gradually degradation. The smaller b is, the smaller m the peak appears at with a greater peak value. Generally, when the number of quasi-basins is considerably large, the smaller the last global search time is. Overall, for $N = 1000$ the last global search time is within 0.6 to 1.6 times of T_θ. As indicated in Fig. 4, the variation of the global search time with respect to the number of global search runs is approximately linear. Thus, we can approximate the average global search time as the average of the first global search time and the last global search time. The resultant upper bound and lower bound of the approximated average global search time are then 0.75 and 1.5 times of T_θ.

C. Discussion

Overall, in this section, we can see that the expected global search time to hit a subthreshold point in local search zones is inversely proportional to the size of local search zones in the search space. Because the uniform random search is employed as the global search component, such results illustrate a baseline behavior of global search in common definitions. It can be observed that when the size ratio between local search zones and the whole search space is very small, the expected global search time will be immensely long because finding a local search zone is very difficult. If the ratio is not very small and permits an acceptable probability to be hit by global search, the expected global search time will drop dramatically. In this case, since the size of local search zones is still small, the local search operator requires a relatively short time to find the optimum solution.

V. Subthreshold Seeker on QBC

In this section, we formulate the expected evaluation time for a subthreshold seeker on a $\mathcal{Q}(G, \mathcal{Y}, m, b)$ as the sum of expected total global search time and the expected total local search time. The expected total global search time is the product of the expected time for global search to enter a local search zone and the expected number of global search runs. The expected total local search time is merely the expected time for local search to find the global optimal points among the local search zones which is proportional to the size of the local search zones in the search space. In this manner, the derived formula can depict how the collaboration between global search and local search influences the performance of memetic algorithms. Then, we propose a sampling test scheme to empirically verify the behavior of the subthreshold seeker on various QBCs. Finally, the empirical results are illustrated to validate the proposed theoretical model.

A. Evaluation Time of Subthreshold Seeker

With the global search time ready, we can now estimate the time to find the minimum point, i.e., the evaluation time T of subthreshold seeker, with the equation

$$T = \frac{cN}{m}\left\lceil\frac{bs}{2}\right\rceil + \frac{m+1}{2}\ . \tag{7}$$

The expected total time over a QBC is considered as the sum of the expected total global search time, the first term, and the expected total local search time, the second term. As discussed in the previous section, it is expected to apply $\lceil bs/2\rceil$ local search runs in order to find the global optima, and thus, $\lceil bs/2\rceil$ global search runs. $\frac{cN}{m}$ represents the average global time with c varies between 0.75 and 1.5, and $\frac{m+1}{2}$ corresponds to the expected time for the local search to find the minimum among subthreshold points. To derive the m that achieves the minimum evaluation time, we solve the following equation with the first derivative of (7)[4] to be zero:

$$T' = -\frac{bscN}{2m^2} + \frac{1}{2} = 0\ . \tag{8}$$

The solution of this equation is $m = \sqrt{bscN}$. Setting m to about $\sqrt{bscN}$ in (7), the subthreshold seeker can achieve the minimum evaluation time T around $\sqrt{bscN}$.[5] Note that the total global search time and the total local search time are near identical when the overall evaluation time is minimum. The following sections verify (7) with the results obtained by our experiments.

B. Sampling Test Scheme

For empirical convenience, we implement the simplest case of QBC, pathwise quasi-basin class (PQBC). PQBC is the class of functions with a simple path spatial structure and a distinct integer value in $\mathcal{Y} = \{1, 2, \cdots, n\}$, where $n = |\mathcal{X}|$, on each vertex. PQBC is formally defined as

Definition 7 (PQBC): Given a finite set $\mathcal{Y} = \{1, 2, \cdots, n\} \subset \mathbb{N}$ and a simple path $G = \overline{v_1 v_2 \ldots v_n}$, the pathwise QBC with b distinct quasi-basins and m subthreshold vertices is defined as $\mathcal{Q}^+(G, \mathcal{Y}, m, b)$.

To investigate the expected subthreshold seeker behavior over a specific PQBC, we sample functions from a specific

[4] For convenience, we omit the ceiling.

[5] The actual value is $\sqrt{bscN}+0.5$, we omit 0.5 as it is a rather small quantity.

```
procedure PATHWISE QBC SAMPLER(v1v2...vn, Y = {1,2,...,n}, m, b, Usize)
    ST ← {1,2,...,m}
    GT ← {m+1, m+2, ..., n}
    i ← 1
    while i ≤ b do
        gt ← UniformPick(GT)
        st ← UniformPick(ST)
        qb_i ← (gt, st)
        i ← i + 1
    end while
    while ST ≠ ∅ do
        st ← UniformPick(ST)
        if Usize then
            i ← i + 1  mod b
        else
            i ← Uniform([1, b])
        end if
        qb_i ← (qb_i, st)
    end while
    QB ← ∪qb_i
    S ← GT ∪ QB
    i ← 1
    while i ≤ n do
        v ← UniformPick(S)
        while length(v) > 1 do
            f(v_i) ← Pop(v)
            i ← i + 1
        end while
        f(v_i) ← v
        i ← i + 1
    end while
    return f
end procedure
```

Fig. 6. Pathwise QBC sampler.

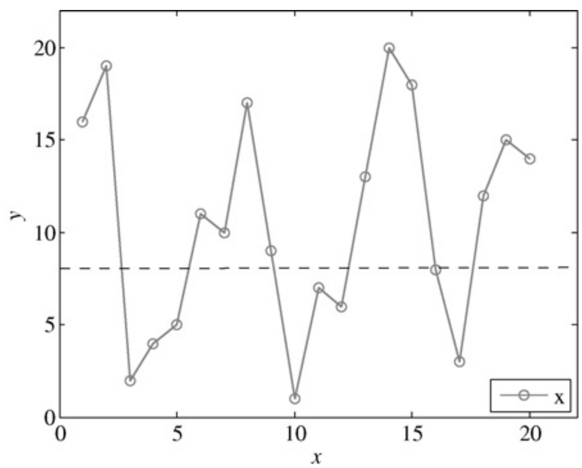

Fig. 7. Example of functions belonging to $\mathcal{Q}^{+}(G, \{1, 2, \cdots, 20\}, 8, 3)$.

PQBC via the PQBC sampler of which the pseudo code is shown in Fig. 6 to generate functions in the pathwise QBC with uniform basins and non-uniform basins. Function *UniformPick* in Fig. 6 samples the input set uniformly at random, returns the sampled value, and removes that value from the input set. Function *Pop* outputs the first value of a sequence and removes the first value from the sequence. The pathwise QBC sampler separates the input values $1, 2, \ldots, n$ into two sets: the set with superthreshold points (GT) and the set with subthreshold points (ST) and then uniformly randomly picks one point from GT and one from ST to construct the basic sequence of a quasi-basin. After every quasi-basin has its basic sequence, the next step is to assign all the subthreshold points to each quasi-basin. When the input boolean parameter *Usize* is set to *True*, the sampler uniformly assigns subthreshold points to every quasi-basin. Otherwise, each subthreshold point is assigned to an arbitrary quasi-basin. We then uniformly randomly pick members in the quasi-basin sequences and GT. When a quasi-basin sequence is picked, this sequence is assigned as the values of next vertices. Fig. 7 illustrates an example of functions in the pathwise QBC which consists 20 points with three quasi-basins containing eight subthreshold points.

In order to empirically verify the time for a subthreshold seeker to find the minimum point of a given PQBC $\mathcal{Q}^{+}(G, \{1, 2, \cdots, n\}, m, b)$, we set the subthreshold seeker's threshold θ to $\beta_m(f)$. In the following sections, we verify (7) with the average time for a subthreshold seeker to find the minimum on various PQBCs. For each PQBC, the performance of the subthreshold seeker is measured by averaging 50 function instances with 20 independent runs on each function instance.

C. Experimental Results

Fig. 8 compares the average evaluation time for a subthreshold seeker with the exhaustive local search ($s = 1$) and the theoretical evaluation time derived by (7) with respect to m on different pathwise uQBCs with $n = 1000$ and $b = 1$, 10, and 250. The solid lines, Ttheo and Ttheo1, indicate the theoretical evaluation time derived from (7) with $c = 1$, while the dashed line Ttheo2 indicates that with $c = 1.5$. The circle (ls) and the cross (gs) represent the average total number of sampling used by local search and global search respectively.

Because there is only one quasi-basin in Fig. 8(a), one global search is required. The proposed model matches the empirical result in this case. In Fig. 8(b), the empirical result matches the proposed model with $c = 1.5$. Such a situation may be caused by the significant global search time growth we observed in Fig. 5. The global search time grows as high as 1.5 when both b and m are quite small. Fig. 8(c) illustrates with large b, the subthreshold seeker performs worse than random search with its evaluation time exceed half of the search space size. Such a result, consisting with the proposed model, indicates that when the number of basins are greater than a quarter of the search space, the problem is unsearchable. Note that in these three cases, the average total local search time and the average total global search time also consist with our theoretical model. The average local search time is about $m/2$ while the average global search time matches (1).

Fig. 9 illustrates the evaluation time of a subthreshold seeker with the exhaustive local search on a non-uniform pathwise QBCs with $n = 1000$ and $b = 10$. Compare the results to that shown in Fig. 8(b), we can observe that although the deviations of the empirical results on non-uniform QBCs is slightly greater than that on uniform QBCs, the two sets of results basically resemble each other. Because the non-uniform pathwise QBCs have every basin's expected size identical, it can be expected that a subthreshold seeker behave statistically similarly on non-uniform and uniform QBCs.

Figs. 10 and 11 compare the theoretical optimal evaluation time with the empirical results with respect to b and n, respectively. In both cases, the exhaustive local search is applied. The solid lines in both figures indicate the optimal theoretical evaluation time predicted by (7) with $c = 1$, and the dashed line indicates that with $c = 1.5$. Both figures demonstrate that the theoretical prediction and the empirical results are in good

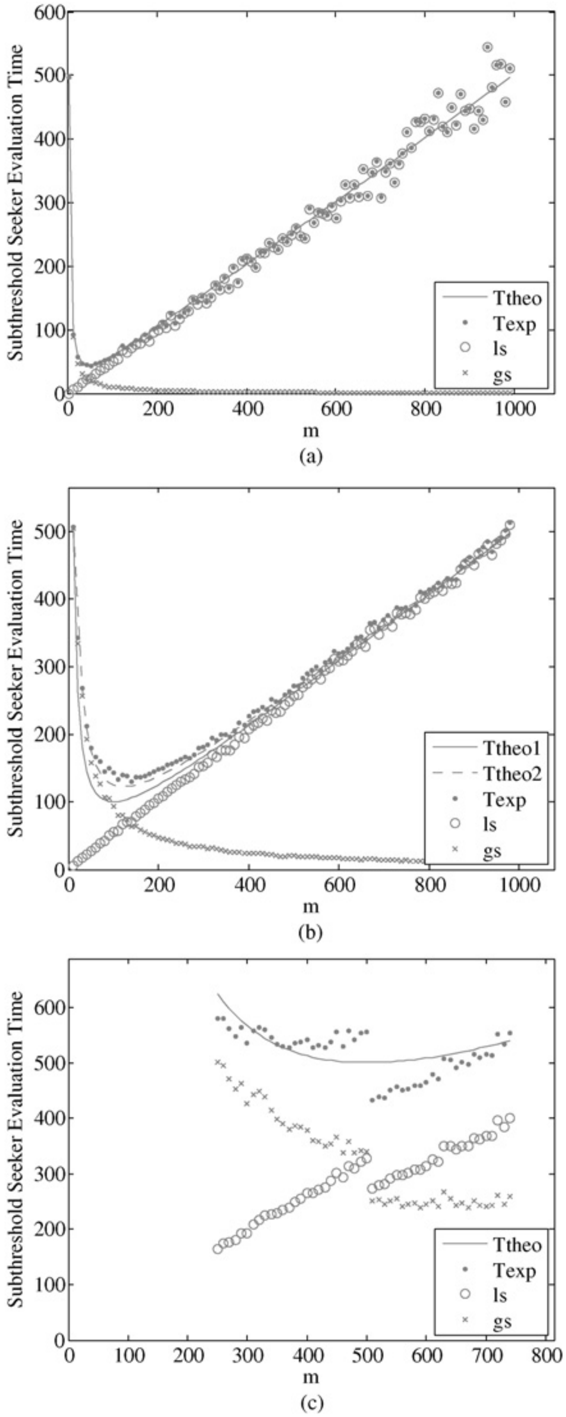

Fig. 8. Time for a subthreshold seeker to find the minimum with respect to m when (a) n = 1000 and b = 1, (b) n = 1000 and b = 10, and (c) n = 1000 and b = 250. The lines, Ttheo, Ttheo1, and Ttheo2, represent the theoretical values derived from (7), and the dot, Texp, represents the average time for a subthreshold seeker to find the minimum on different PQBCs. The average total sampling counts used by local search and global search are also recorded as ls and gs, respectively.

agreement, and therefore, (7) is dimensionally validated for different factors.

Fig. 12 illustrates the evaluation time with respect to m when non-exhaustive local search components, i.e., $s > 1$, are used. In both cases, the solid lines represent the theoretical evaluation time predicted by (7) with $c = 1.5$. These

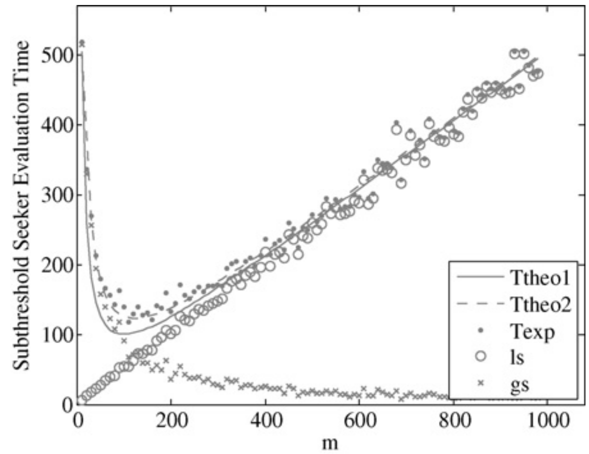

Fig. 9. n = 1000, b = 10, non-uniform quasi-basin.

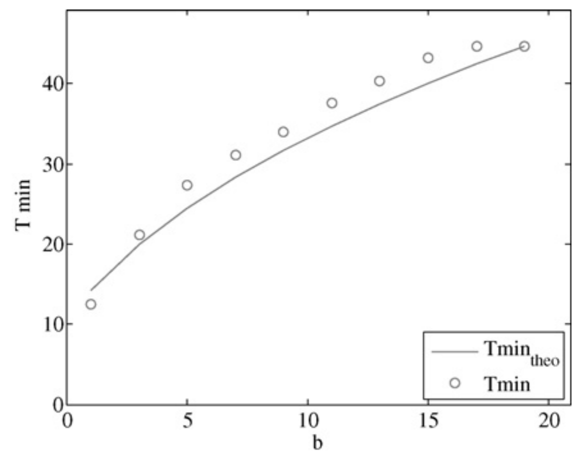

Fig. 10. Optimal evaluation time versus b when n = 100. The solid line indicates the theoretical value predicted by (7) with c = 1, and the dots indicate the empirical results.

empirical results also well match the proposed theoretical model (7).

VI. Discussion

In this section, we first explain how the subthreshold seeker can be regarded as a representative archetype of MAs and how the theoretical model can depict the general behavior of MAs. Then, we connect the proposed model to previous related studies in the literature. Finally, we discuss the extensions and future work of the proposed model.

A. Subthreshold Seeker as a Representative Archetype of MA

Since the proposed model of the subthreshold seeker on different QBCs has been validated by the empirical results in the previous section, in this section, we revisit our framework and discuss how our theoretical model is representative of MAs on a broad range of problems. The origin of our framework is the concept of local search zones. Based on this concept, the search space viewed by a search process can be partitioned into local search zones, which are areas preferred by exploitation, and parts of no interests. Global

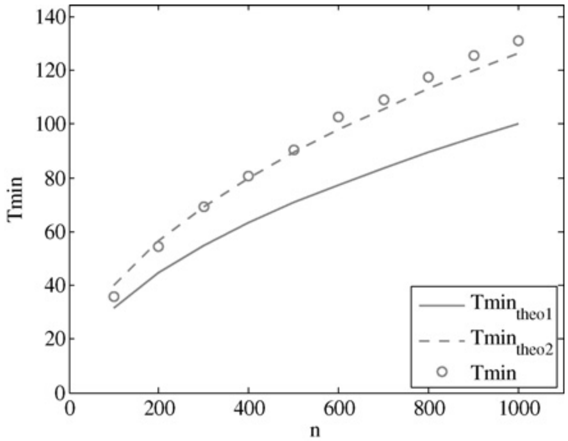

Fig. 11. Optimal evaluation time versus n when b = 10. The solid line indicates the theoretical value predicted by (7) with c = 1, the dashed line indicates that with c = 1.5, and the dots indicate the empirical results.

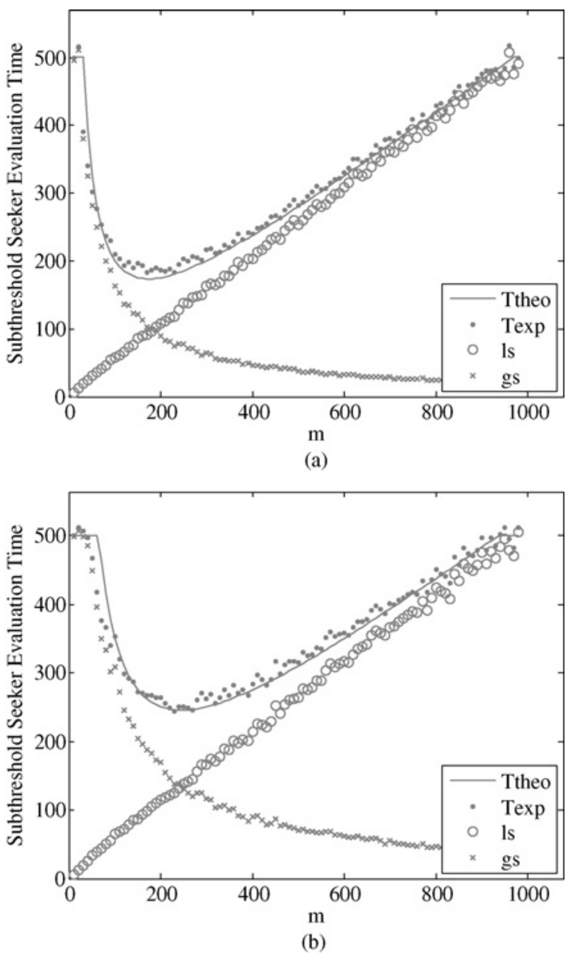

Fig. 12. Evaluation time for subthreshold seekers with a non-exhaustive local search component (s > 1). (a) n = 1000, b = 10, s = 2, uniform quasi-basin. (b) n = 1000, b = 10, s = 4, uniform quasi-basin.

search explores the whole space to find a local search zone for local search to exploit. The size of the local search zones in the search space affects the time for global search to find a point in local search zones and the time for local search to find the optimal solution in local search zones. The number of local search zones and the efficiency of local search further influence the required local search runs and global search runs. The performance of a memetic algorithm is thus determined by the efficiency of global search, the efficiency of local search, and the distribution of the local search zones. As the distribution of local search zones is dictated by the landscape of the search space and the local search criterion, by assessing the impact of the distribution of local search zones on the evaluation time, we can analyze the physics behind the collaboration between global search and local search on various problem classes.

Fitness-relevant and diversity-relevant metrics are common local search criteria in practical memetic algorithms. They form complicated local search zones which are difficult to measure. As we aim to model the pure collaboration between global search and local search, we consider the global search exhibits fair exploration and the local search exhibits fair exploitation. Hence, the diversity-relevant metrics which are commonly used to balance the exploration and the exploitation can be ignored. To build a more comprehensible model, we adopt fitness values as a representative fitness-relevant local search criterion. This criterion forms local search zones that can be referred to as quasi-basins. The QBC, which is accordingly defined, then categorizes all problems according to their quasi-basin distributions. In this way, our model is capable of describing the general behavior of memetic algorithms on a broad range of problems.

Generally, the QBC categorizes all the problems according to their search space landscape and the local search threshold. Besides the number of subthreshold points and the number of the quasi-basins, the QBC does not put any other constraints on the problems belonging to the same class. In other words, except that the subthreshold points may tend to gather according to the number of quasi-basins, both the subthreshold points and the superthreshold points of an instance that belongs to one QBC can be arbitrarily distributed in local search zones and the rest of the search space, respectively. As the expected performance of a local search algorithm on a QBC is calculated over all the possible instances belonging to the QBC, the local search processes on an instance of a QBC can be considered virtually as random sequences of subthreshold points. Thus, the expected performance of a local search generally would resembles that of a random sampling algorithm. On the other hand, as the subthreshold points tend to gather as quasi-basins, a greedy global search may perform worse than random sampling because it is not likely to discover a quasi-basin around a discovered quasi-basin. In fact, a random sampling algorithm may be the perfect explorer on a QBC due to its full diversity. From this point of view, the subthreshold seeker which virtually employs random search as its global search and local search can be considered a representative archetype of memetic algorithms on QBCs.

The proposed theoretical model manifests and gives explanations to the following facts.

1) Memetic algorithms which perform local search to few qualified points perform better.
2) The efficiency of local search greatly influences the evaluation time of memetic algorithms.
3) The physical landscape of a problem greatly influence the evaluation time of memetic algorithm.

We can assume that the average global search time to enter a local search zone is inversely proportional to the size ratio between local search zones and the search space, while the average total local search time is proportional to the size of the local search zones. Putting these two terms together, we can obtain the "V-shaped" curve which resembles those derived from (7). This V-shaped curve implies that a good collaboration between global search and local search should guarantee a short average global search time to hit local zones and sufficiently small sizes of local zones for the local search to exploit. Regarding the influence of the size of local search zones on the average global search time to find local search zones and the average time for local search to find the optimal point, memetic algorithms which have small sized local search zones will perform better. As mentioned in Section III-A, local search zones are generally zones consisting of qualified high-fitness points. A small size of local search zones implies that local search only be applied to few qualified individuals. This observation is consistent with the use of elitism in local search candidate selection and the infrequent local search principle in quite a number of research works [9], [10], [31]–[33]. It is also notable that several studies adopt a local search/global search ratio which is consistent with our theoretical model [36].

In our model, the local search component adopted by the subthreshold seeker exploits a quasi-basin via visiting the neighbors of current search point. The local search parameter of the employed local search operator is connected to how well a quasi-basin is exploited. Recall that when $s = 1$, the exhaustive local search will eventually visit all the points in a quasi-basin. In this case, the local minimum of a quasi-basin, which may be the global minimum, will be visited, and thus, only one local search run for each basin is required. For other local search parameter greater than one, there are chances for one local search run to miss the global minimum in a quasi-basin, and thus, more local search runs on this quasi-basin and more global search runs to hit this quasi-basin are required. The cost will be the extra global search time to enter the quasi-basin again when the algorithm guarantees non-repeated sampling. Fig. 12 and the factor s in (7) demonstrate the effect of the degree of exploitation of a basin. Such an effect implies that a good local search operator ought to fully exploit the given quasi-basin, at least the local minimum resides in the quasi-basin should be found, to guarantee a good local search and global search coordination. This inference is consistent with the empirical results of those studies [10], [39], [40] concluding that longer but not excessive local search lengths are favored in memetic algorithms.

Another notable factor is the number of local search zones in the search space. Our model illustrates that the number of global search runs is proportional to this factor. Recall that we represent the search space viewed by a search process as a graph composed of the neighborhood defined by the employed local search algorithm. The size of local search zones is determined by the local search criterion and the connectivity formed by the local search operator. Given the same local search criterion, the local search operator which forms fewer local search zones will perform better. This suggests that a good local search operator should be able to find local search points regardless of the physical landscape of a problem. This is somehow difficult for naive greedy local search algorithms to achieve and may require landscape knowledge given by the user or learned from the search process. However, operators with this kind of ability to cross the physical landscape of a problem somehow deviate from the traditional definition of local searchers. Thus, for typical local search, the number of local search zones are primarily defined by the physical landscape of a problem and the local search criterion. The physical landscape of a problem is usually connected to the number of niches of a problem. Our model also takes into account this crucial factor and delineates the relationship between this factor and the evaluation time. The proposed theoretical model indicates that for a fair memetic algorithm, the expected evaluation time should scale at most as the square root of the number of niches of a problem.

Despite the aforementioned consistency between the proposed model and the elitism based strategy in local search candidate selection, infrequent local search, long local search length, and local search/global search ratio, some previous studies also show a strong connection to our model. In an investigation on the balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling [39], the authors examined 132 combinations of 11 values of k, which is the maximum number of examined neighbors of the current solution, and 12 values of p_{LS}, which is the local search probability applied to the tournament selected individuals. The former factor k connects to the degree of how well a feasible sub-region can be exploited, and the second factor p_{LS} connects to the threshold that triggers local search. The authors found that the combination of the maximum k value and the minimum nonzero p_{LS} value achieved the best performance, the lowest cost of flowshop scheduling, in their experiments. The V-shaped curve of cost along the axis of the maximum k with respect to p_{LS} in their Fig. 13 resembles our V-shaped curve of the evaluation time in Fig. 8. Because the stop criterion of their experiments is the evaluation of a fixed number of points, the factor combinations that require less evaluation time to find the global minimum will have better solution quality, i.e., lower cost. This agreement implies that the proposed model may be adopted to give a theoretical explanation to the internal working of their multiobjective memetic algorithms.

Another set of intriguing empirical results is presented in the study of parameterizing local search [11]. In that study, the authors applied a hybrid approach to the memory cost minimization problem with various local search parameter settings. The local search parameter refers to the intensity of the local search method, a tractable algorithm called code size dynamic programming post optimization, applied to every individual in the population. The authors depicted in Fig. 13 in their paper that when a fixed runtime is used, the number of generations completed decreases rapidly as the local parameter increases. That the global search time is proportional to the number of generations implies the curve, indicating the global search time, resembles the expected T_θ in our Fig. 2. As the expected global search time is illustrated and the expected local search time will be proportional to the intensity of local search, summing up the expected global search time and the expected local search time, a V-shaped curve of the expected evaluation time with respect to the intensity of local search will be obtained. Fig. 12 in their study illustrates the attained solution quality, lower cost preferred, versus the setting of local search intensity. As previously discussed, lower attained cost in a given fixed time leads to shorter expected time to find the optimal solution. This figure also resembles the V-shaped curve of our theoretical model which confirms that the proposed model is quite applicable to their conclusions.

Although in these two studies, practical memetic algorithms, instead of the subthreshold seeker, are employed and investigated, the trend of their evaluation time resembles the proposed model developed based on the subthreshold seeker. It indicates that our theoretical model is indeed representative of memetic algorithms as we previously inferred. Another interesting study is the optimal bounds on finding fixed points of contraction mappings proved by [48]. In this investigation, the authors presumed that the expected lower bound of a randomized algorithm to find the fixed point of a contraction mapping $f : M \to M$ on a finite metric space (M, d) is $\Omega(\sqrt{|M|})$ and proved this bound is valid. In this fixed point problem, given any point $x \in M$ with the $d(x, f(x))$ the kth largest, one can find the fixed point with k steps via a valid deterministic algorithm. Consider the set exploited by the deterministic algorithm as the subthreshold sub-space which consists only one quasi-basin and the random sampling process to find a starting point for the deterministic algorithm as global search, according to (7), the best size of the subthreshold sub-space should be $\Omega(\sqrt{|M|})$ resulting in an expected optimal evaluation time of $\Omega(\sqrt{|M|})$. Thus, our theoretical model can also provide a reasonable, theoretical interpretation to this presumed value $\Omega(\sqrt{|M|})$.

Although our model is developed based on the subthreshold seeker, the subthreshold seeker can be taken as a representative archetype of memetic algorithms as it employs a fair explorer as its global search and a fair exploiter as its local search. The proposed model delineates the general behavior of memetic algorithms: how the global search behaves with respect to local search criteria, how the local search behaves with respect to local search criteria, how the local search criteria coordinate global search and local search, and how the local search efficiency and the problem landscape influence the evaluation time. The preceding paragraphs illustrated the consistence between our model and various memetic algorithm-problem complexes studied in the literature, either discrete problems or continuous problems, validate that our theoretical model is capable of providing a unified explanation to the physics behind memetic algorithms.

B. Extensions and Future Work

In our model, we propose the concept of local search zones and link it to the QBCs. Then, the subthreshold seeker, which employs random search as global search and local search, clearly illustrates the collaboration of global search and local search on various QBCs. In this presentation, estimating local search zones as quasi-basins, the global search can be considered as a baseline explorer and the local search can be considered as a standard exploiter. Thus, the model can reveal the essential relationship among the performance of memetic algorithms, the problem class categorized by QBCs, and the collaboration of global search and local search. In practical memetic algorithms, not only global searchers are population-based greedy approaches but also local searchers are greedy approaches on continuous problems. The local search criteria constantly depend on the status of the search process and the corresponding local search zones are difficult to measure. Although quasi-basins can roughly estimate local search zones, further studies on local search zones are required if more accurate models are to be developed. Extending our model to an instance of algorithm-problem complexes requires much further investigations into the following scopes.

1) The relationship between local search criteria and local search zones on discrete problems.
2) The behavior of the population-based greedy global search on discrete problems.
3) The behavior of the greedy local search on discrete problems.
4) All the three items in infinite and/or continuous domains.

Here, we discuss the first three scopes via adopting a memetic algorithm in the present framework. The memetic algorithm illustrated in Fig. 13 is a modified version of (μ+λ)-MA adopted in [25]. The algorithm first samples an initial population of size μ from $\mathcal{X}$, and then in each generation, generates λ children via the parent selection, mutation, and local search operations. The mutation operation flips each bit in x independently with probability $1/\ell$ where ℓ is $\lceil \log_2(|\mathcal{X}|) \rceil$. If the mutated offspring x' satisfies the local search criterion, it undergoes the local search operation. The best μ of the μ+λ individuals are selected as the survivors of that generation. The algorithm continues till its stopping criterion is satisfied. The first problem we confront is that for a local search criterion that other than a fitness value threshold, we must find a way to transfer the local search criterion to a fitness value threshold or the size of local search zones in a way that we can link the algorithm-problem complex to a QBC. Though most local search criteria are fitness-relevant and favor elitists, they are dependent on the current population and dynamic along generations. Further investigations are required to be devoted to this issue to provide some proper measurement of the size of local search zones. However, as the dynamics change slightly between generations, approximating the resultant size of local search zones with some statistical techniques may provide good solutions to this issue.

To manifest how population based greedy searchers perform on QBCs, the local search criterion of the (μ+λ)-MA is set the same as the subthreshold seeker. In other words, when the fitness value of the mutated individual is better than the threshold value, local search is applied to the mutated individual. Fig. 14 shows how a (20+20)-MA behaves with different local search operators on QBCs. The greedy local search keeps on moving to a better neighbor until no further move can be made. The exhaustive local search acts identically as the aforementioned subthreshold seeker does. In Fig. 14(a), the evaluation time of this (20+20)-MA can be approximated by the dashed line *Ttheo* which is (7) with $c \cdot s = 3.3$. In

```
procedure (μ+λ)-MEMETIC ALGORITHM(𝒳, 𝒴, f : 𝒳 → 𝒴)
    t ← 1
    Initialize population P_1 with μ individuals
    while the stopping criterion is not satisfied do
        P'_t ← ∅
        i ← 1
        while i ≤ λ do
            Choose x ∈ P_t uniformly at random
            x' ← mutation(x)
            if x' satisfies local search criterion then
                x'' ← localsearch(x')
            else
                x'' ← x'
            end if
            P'_t ← P'_t ∪ x''
            i ← i + 1
        end while
        P_{t+1} ← best μ individuals in P'_t ∪ P_t
        t ← t + 1
    end while
end procedure
```

Fig. 13. (μ+λ)-memetic algorithm.

Fig. 14(b), although the evaluation time could not be properly approximated by (7), the V-shape remains. The triangles, *lscnt*, in these two figures represent the number of local search runs. From these figures, we can find that the applied greedy local search is much less efficient than the exhaustive search with its limited total local search time denoted by the circles, *ls*. The higher *lscnt* than the total local search time in the case of MA with greedy local search suggests revisiting of local searched points. Note also that in both figures, the memetic algorithm resembles the global search behavior of random search with an offset and the greedy local search resembles the exhaustive search with a degraded gradient. In these two memetic algorithms, our theoretical model is still capable of capturing the essence of the collaboration between global search and local search.

To accurately estimate the evaluation time of an instance of MA search process, one needs to take into account the influence of the population size and the exploration ability limited by its greediness to estimate its expected global search time and assess the efficiency of local search which corresponds to the parameter s. Another notable characteristic is that both memetic algorithms in the two cases perform worse than the subthreshold seeker on QBCs. This is consistent with our earlier statement that the random sampling is a better global explorer than any greedy algorithms on QBCs. This may seem contrary to the practical memetic algorithms. However, as the QBC categorizes arbitrary problems according to the quasi-basin distribution, it does not guarantee that all the problems within a QBC exhibit a regularity which a greedy algorithm can take advantages of. To manifest the optimization characteristics of greedy algorithms, fast convergence versus degrading diversity, the QBC framework must be extended to define classes of continuous-like discrete problems. In our opinion, adopting the concept of discrete Lipschitz class (DLC) [47] may be a good choice. In [47], the Lipschitz functions, functions with bounded slope, are transferred to DLC to describe continuous problems in discrete domains. Combining

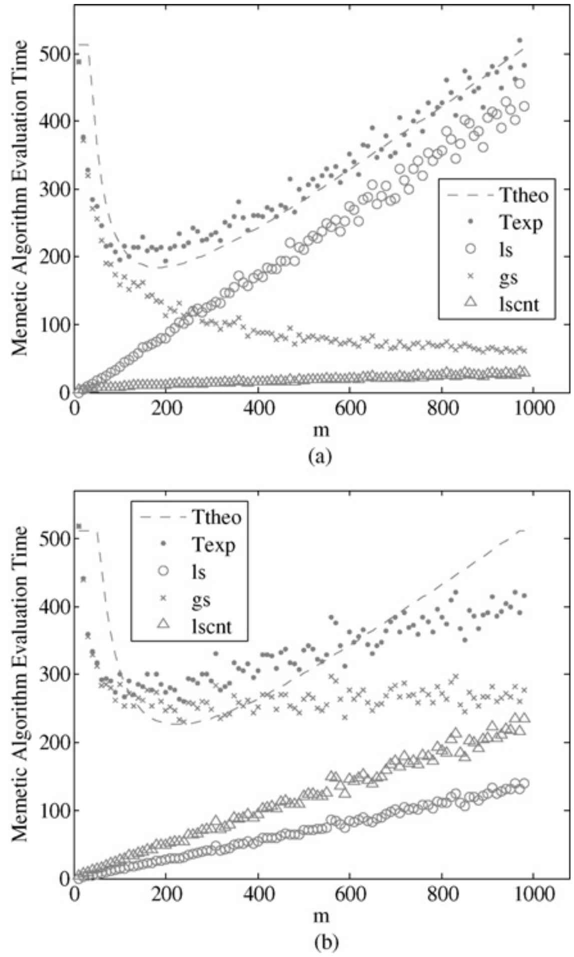

Fig. 14. Evaluation time for (20+20) memetic algorithms. (a) n = 1024, b = 10, MA with exhaustive local search. (b) n = 1024, b = 10, MA with greedy local search.

the DLC and QBC may provide a desired model in discrete domains that exhibits the characteristics of optimization in continuous domains.

For continuous problems, further efforts are required to extend all the analysis from discrete problems to continuous problems. In continuous domains, both $\mathcal{X}$ and $\mathcal{Y}$ are infinite sets. The first question may be how to extend the search space represented by a graph to fit the continuous scenario. If we can define the local search zones in a similar manner in continuous domains, we may start to investigate the behavior of memetic algorithms based on the modified framework. It may be much harder to estimate in continuous domains the expected global search time to enter local search zones, the expected total local search time to find the optimal solution, and the efficiency of local search of a given memetic algorithm. Once all these issues are resolved, an instance of algorithm-problem complex in continuous domains can be successfully delineated as well as the performance of a global search algorithm and a local search algorithm can be measured and compared in continuous domains. Memetic algorithm designers can thus select their global search algorithms and local search algorithms and design the local search criterion by following the guideline provided by the modified framework. As practical problems are generally black-box optimization, a designer with the aforementioned knowledge must dynamically estimate the number of local search zones and the size ratio of the local search zones and the search space to accordingly adjust the local search criterion in order to achieve the best collaboration between global search and local search.

Overall, although our model in this paper depicts the core, general behavior of memetic algorithms, it might potentially be extended to specific instances of memetic algorithm-problem complexes. Based on the concept of local search zones, the expected performance of a memetic algorithm can be assessed by analyzing the following components individually: the expected time for a global search algorithm to find a local search point, the expected time for a local search algorithm to find the optimal point in the local search zones, and the efficiency of a local search algorithm. With all the information available, algorithm designers can compare and select proper global search algorithms and local search algorithms and adopt the optimal local search criterion on the target problem accordingly. As designing an optimal memetic algorithm on a given problem has been the primary goal of memetic algorithms, extending our model to more practical memetic algorithms may provide a feasible way to achieve the goal. This may be an interesting and challenging task in the field of memetic algorithms.

VII. Summary and Conclusion

In this paper, we proposed the concept of local search zones. Based on this concept, we introduced the QBC to estimate the local search zones and adopted the subthreshold seeker as a representative archetype of memetic algorithms in order to analyze the collaboration between global search and local search on various quasi-basin classes. The derived theoretical model was capable of depicting the essence of the collaboration between a baseline global searcher and a standard local searcher. The efficiency of local search algorithms and the niches of problems were also taken into account in the proposed model.

The proposed theoretical model indicates that the global search time to find a point to start local search is inversely proportional to the size ratio between local search zones and the search space. The total local search time is proportional to the size of local search zones. Appropriate settings of local search criteria should guarantee sufficiently small sizes of local search zones. As the theoretical model cannot only well describe the behavior of the subthreshold seeker for the empirical results but can also capture the general behavior of various memetic algorithms proposed and observed in the literature, it can provide a unified explanation to the physics behind memetic algorithms and may reveal important insights to the design of memetic algorithms.

Furthermore, the proposed model is also capable of being extended to describe some specific memetic algorithms. The concept of local search zones provides an alternative way to assess the performance of a memetic algorithm by analyzing individually the performance of global search algorithms and the performance of local search algorithms. In this way, memetic algorithm designers may compare and select their global search algorithms and local search algorithms and adopt appropriate local search criteria for their problem. As the research direction of this paper may be a feasible way to achieve a better memetic algorithm design, along this line, much effort may be worth putting into further investigations.

Acknowledgment

The authors are grateful to the National Center for High-Performance Computing for computer time and facilities.

References

[1] R. Meuth, M.-H. Lim, Y.-S. Ong, and D. Wunsch, "A proposition on memes and meta-memes in computing for higher-order learning," *Memetic Comput.*, vol. 1, no. 2, pp. 85–100, 2009.

[2] Y.-S. Ong, M. H. Lim, and X. Chen, "Memetic computation: Past, present and future," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 24–31, May 2010.

[3] F. Neri and E. Mininno, "Memetic compact differential evolution for Cartesian robot control," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 54–65, May 2010.

[4] G. Acampora, M. Gaeta, and V. Loia, "Exploring e-learning knowledge through ontological memetic agents," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 66–77, May 2010.

[5] L. Jiao, M. Gong, S. Wang, B. Hou, Z. Zheng, and Q. Wu, "Natural and remote sensing image segmentation using memetic computing," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 78–91, May 2010.

[6] Z. Zhu, S. Jia, and Z. Ji, "Toward a memetic feature selection paradigm," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 41–53, May 2010.

[7] R. Meuth, E. Saad, D. Wunsch, and J. Vian, "Memetic mission management," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 32–40, May 2010.

[8] R. Ruiz-Torrubiano and A. Suarez, "Hybrid approaches and dimensionality reduction for portfolio selection with cardinality constraints," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 92–107, May 2010.

[9] W. E. Hart, "Adaptive global optimization with local search," Ph.D. dissertation, Dept. Comput. Sci. Eng., Univ. California, San Diego, 1994.

[10] M. W. S. Land, "Evolutionary algorithms with local search for combinatorial optimization," Ph.D. dissertation, Dept. Comput. Sci. Eng., Univ. California, San Diego, 1998.

[11] N. Bambha, S. Bhattacharyya, J. Teich, and E. Zitzler, "Systematic integration of parameterized local search into evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 137–155, Apr. 2004.

[12] N. Krasnogor, "Studies on the theory and design space of memetic algorithms," Ph.D. dissertation, Faculty Comput. Eng. Math. Sci., Univ. West England, Bristol, U.K., 2002.

[13] Y. S. Ong and A. J. Keane, "Meta-Lamarckian learning in memetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 99–110, Apr. 2004.

[14] J. Smith, "Coevolving memetic algorithms: A review and progress report," *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 37, no. 1, pp. 6–17, Feb. 2007.

[15] H. Mühlenbein, "How genetic algorithms really work: I. Mutation and hillclimbing," in *Proc. 2nd Conf. PPSN*, 1992, pp. 15–25.

[16] G. Rudolph, *Convergence Properties of Evolutionary Algorithms*. Hamburg, Germany: Verlag Dr. Kovač, 1997.

[17] S. Droste, T. Jansen, and I. Wegener, "A rigorous complexity analysis of the (1 + 1) evolutionary algorithm for separable functions with Boolean inputs," *Evol. Comput.*, vol. 6, no. 2, pp. 185–196, 1998.

[18] J. Garnier, L. Kallel, and M. Schoenauer, "Rigorous hitting times for binary mutations," *Evol. Comput.*, vol. 7, no. 2, pp. 173–203, 1999.

[19] H.-G. Beyer, *The Theory of Evolution Strategies* (Natural Computing Series). Berlin, Germany: Springer, 2001.

[20] S. Droste, T. Jansen, and I. Wegener, "On the analysis of the (1+1) evolutionary algorithm," *Theor. Comput. Sci.*, vol. 276, nos. 1–2, pp. 51–81, 2002.

[21] J. He and X. Yao, "Toward an analytic framework for analyzing the computation time of evolutionary algorithms," *Artif. Intell.*, vol. 145, nos. 1–2, pp. 59–97, 2003.

[22] M. Jiang, Y. Luo, and S. Yang, "Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm," *Inform. Process. Lett.*, vol. 102, no. 1, pp. 8–16, 2007.

[23] T. Jansen and I. Wegener, "The analysis of evolutionary algorithms: A proof that crossover really can help," *Algorithmica*, vol. 34, no. 1, pp. 47–66, 2008.
[24] D. Sudholt, "On the analysis of the (1 + 1) memetic algorithm," in *Proc. ACM SIGEVO GECCO*, 2006, pp. 493–500.
[25] D. Sudholt, "The impact of parametrization in memetic evolutionary algorithms," *Theor. Comput. Sci.*, vol. 410, no. 26, pp. 2511–2528, 2009.
[26] L. Kallel, B. Naudts, and C. R. Reeves, "Properties of fitness functions and search landscapes," in *Theoretical Aspects of Evolutionary Computing*. Berlin, Germany: Springer-Verlag, 2001, pp. 175–206.
[27] A. Sinha, Y.-P. Chen, and D. E. Goldberg, "Designing efficient genetic and evolutionary algorithm hybrids," in *Recent Advances in Memetic Algorithms* (Studies in Fuzziness and Soft Computing, vol. 166). Heidelberg, Germany: Physica-Verlag, 2004, pp. 259–288.
[28] Q. H. Nguyen, Y.-S. Ong, and M. H. Lim, "A probabilistic memetic framework," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 604–623, Jun. 2009.
[29] S. Christensen and F. Oppacher, "What can we learn from no free lunch? A first attempt to characterize the concept of a searchable function," in *Proc. GECCO*, 2001, pp. 1219–1226.
[30] D. Whitley and J. Rowe, "Subthreshold-seeking local search," *Theor. Comput. Sci.*, vol. 361, no. 1, pp. 2–17, 2006.
[31] K.-H. Liang, X. Yao, and C. Newton, "Evolutionary search of approximated n-dimensional landscapes," *Int. J. Knowl. Based Intell. Eng. Syst.*, vol. 4, no. 3, pp. 172–183, 2000.
[32] M. Tang and X. Yao, "A memetic algorithm for VLSI floorplanning," *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 37, no. 1, pp. 62–69, Feb. 2007.
[33] Y.-H. Liu, "A hybrid scatter search for the probabilistic traveling salesman problem," *Comput. Oper. Res.*, vol. 34, no. 10, pp. 2949–2963, 2007.
[34] M. Lozano, F. Herrera, N. Krasnogor, and D. Molina, "Real-coded memetic algorithms with crossover hill-climbing," *Evol. Comput.*, vol. 12, no. 3, pp. 273–302, 2004.
[35] S. Garcia, J. R. Cano, and F. Herrera, "A memetic algorithm for evolutionary prototype selection: A scaling up approach," *Pattern Recognit.*, vol. 41, no. 8, pp. 2693–2709, 2008.
[36] D. Molina, M. Lozano, C. Garcia-Martinez, and F. Herrera, "Memetic algorithms for continuous optimization based on local search chains," *Evol. Comput.*, vol. 18, no. 1, pp. 27–63, 2010.
[37] F. Neri, J. Toivanen, G. L. Cascella, and Y.-S. Ong, "An adaptive multimeme algorithm for designing HIV multidrug therapies," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 4, no. 2, pp. 264–278, Apr.–Jun. 2007.
[38] J. Tang, M. H. Lim, and Y. S. Ong, "Diversity-adaptive parallel memetic algorithm for solving large scale combinatorial optimization problems," *Soft Comput.*, vol. 11, no. 9, pp. 873–888, 2007.
[39] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 204–223, Apr. 2003.
[40] Z. Zhu, Y.-S. Ong, and M. Dash, "Wrapper-filter feature selection algorithm using a memetic framework," *IEEE Trans. Syst. Man Cybern. B Cybern.*, vol. 37, no. 1, pp. 70–76, Feb. 2007.
[41] P. Merz and B. Freisleben, "Fitness landscapes and memetic algorithm design," in *New Ideas in Optimization*. New York: McGraw-Hill, 1999, pp. 245–260.
[42] I. Paenke, T. J. Kawecki, and B. Sendhoff, "The influence of learning on evolution: A mathematical framework," *Artif. Life*, vol. 15, no. 2, pp. 227–245, 2009.
[43] M. Le, Y.-S. Ong, Y. Jin, and B. Sendhoff, "Lamarckian memetic algorithms: Local optimum and connectivity structure analysis," *Memetic Comput.*, vol. 1, no. 3, pp. 175–190, 2009.
[44] C. Witt, "Runtime analysis of the (μ + 1) EA on simple pseudo-Boolean functions," *Evol. Comput.*, vol. 14, no. 1, pp. 65–86, 2006.
[45] B. Doerr, N. Hebbinghaus, and F. Neumann, "Speeding up evolutionary algorithms through asymmetric mutation operators," *Evol. Comput.*, vol. 15, no. 4, pp. 401–410, 2007.
[46] A. H. Wright and J. N. Richter, "Strong recombination, weak selection, and mutation," in *Proc. ACM SIGEVO GECCO*, 2006, pp. 1369–1376.
[47] P. Jiang and Y.-P. Chen, "Free lunches on the discrete Lipschitz class," *Theor. Comput. Sci.*, vol. 412, no. 17, pp. 1614–1628, 2011.
[48] C.-L. Chang and Y.-D. Lyuu, "Optimal bounds on finding fixed points of contraction mappings," *Theor. Comput. Sci.*, vol. 411, nos. 16–18, pp. 1742–1749, 2010.

Jih-Yiing Lin received the B.S. and M.S. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2000 and 2002, respectively. She is currently working toward the Ph.D. degree in computer science from the Department of Computer Science, National Chiao Tung University.

She was with Sunplus Technology, Hsinchu, from 2002 to 2007. Her current research interests in the field of genetic and evolutionary computation include theories, working principles, particle swarm optimization, and linkage learning techniques.

Ying-Ping Chen (M'04) received the B.S. and M.S. degrees in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1995 and 1997, respectively, and the Ph.D. degree from the Department of Computer Science, University of Illinois, Urbana, in 2004.

He was an Assistant Professor from 2004 to 2009 and has been an Associate Professor since 2009 with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan. His current research interests in the field of genetic and evolutionary computation include theories, working principles, particle swarm optimization, estimation of distribution algorithms, linkage learning techniques, and dimensional/facet-wise models.