

Dynamics of Local Search Heuristics for the Traveling Salesman Problem

Waiqi Li and Bahram Alidaee

Abstract—This paper experimentally investigates the dynamical behavior of a search process in a local heuristic search system for a combinatorial optimization problem. Or-opt heuristic algorithm is chosen as the study subject, and the well-known traveling salesman problem (TSP) serves as a problem testbed. This study constructs the search trajectory by using the time-delay method, evaluates the dynamics of the local search system by estimating the correlation dimension for the search trajectory, and illustrates the transition of the local search process from high-dimensional stochastic to low-dimensional chaotic behavior.

Index Terms—Combinatorial optimization, data visualization, dynamical complexity, heuristics.

I. INTRODUCTION

HEURISTIC search techniques have been very popular as a means of finding reasonable solutions to hard combinatorial optimization problems. However, despite the simplicity of the techniques and their widespread use, little theory is available as a guide for the design of effective heuristic search algorithms. The design has been, and remains, very much an art [1]. People usually study empirically the performances of such techniques, simply by trying a heuristic, comparing with others, and observing the relative quality of results. Identifying and encoding the geometric structure of the solution space and the dynamical behavior of a heuristic search process is a difficult task. As the number and complexity of heuristic algorithms grow, it becomes increasingly difficult for a heuristic search system builder to imagine how the search process behaves. Although the topic of studying heuristic search behavior has attracted some attention in the literature [2], [3], there are, indeed, few theoretical publications and experimental works that address the dynamical nature of heuristic search processes. The dynamics of heuristic search processes is not well understood because the complexity of the search process makes the phenomenon not amenable to simple analysis and leads to differing views on modeling techniques and analytical methods. Certainly, it is unsatisfactory that, so far, we do not have an effective technique to study the behavior of heuristic search processes. Not even a useful search path analysis can be conducted. It is this very situation that motivates this paper to cover one of the important topics in the research and application of heuristic algorithms: reconstructing

the trajectory of a heuristic search process and investigating its dynamical behavior.

In order to discover how such a heuristic search process behaves when it searches for solutions to a particular combinatorial optimization problem, and in order to acquire a better understanding of the heuristic search mechanism, this paper reconstructs the search trajectory of Or-opt algorithm for the traveling salesman problem (TSP). Then, based on the reconstructed trajectory, the dynamical property of the search system is defined quantitatively.

II. COMBINATORIAL OPTIMIZATION AND HEURISTICS

Combinatorial optimization studies hard optimization problem in which there can be many possible solutions. In general, the primary task is to find the best element in a finite solution set S , with respect to some objective function. Some representative examples of combinatorial optimization problems are the TSP, the assignment problem, the set covering problem, and the vehicle routing problem. These problems have both theoretical and practical significance.

An instance of a combinatorial problem is usually formalized as a pair (S, f) , where S is the finite set of feasible configurations, called *configuration space* or *solution space*, and $f : s \rightarrow \mathcal{R}$ is a real-valued score function (also called *objective function*) that assigns a value to each configuration. We wish to find a solution $s^* \in S$ with the optimal value

$$\forall s \in S, f(s^*) \leq f(s) \quad \text{or} \quad f(s^*) \geq f(s). \quad (1)$$

Such a point s^* is called a globally optimal solution to the given problem instance.

People have used many strategies to attack combinatorial optimization problems. However, the NP-completeness and intractability of combinatorial optimization problems have led many researchers to employ heuristic methods to solve those hard problems. A *heuristic* is a technique that seeks good solutions at a reasonable computational cost without being able to guarantee either feasibility or optimality, or even, in many cases, to state how close to optimality a particular feasible solution is [4]. Many heuristic algorithms have been developed to solve the combinatorial optimization problems. Useful references to heuristic techniques include [4]–[6].

There are three important aspects in the analysis of heuristics:

- 1) search performance;
- 2) time or space complexity;
- 3) dynamical complexity.

The first two aspects have occupied a fair amount of attention. A large body of literature has been focused on measurement

Manuscript received March 19, 1999; revised November 9, 2001 and February 5, 2002. This paper was recommended by Associate Editor W. T. Scherer. W. Li is with the School of Management, University of Michigan, Flint, MI 48502 USA (e-mail: weli@umflint.edu).

B. Alidaee is with the School of Business, University of Mississippi, University, MS 38677 USA (e-mail: balidaee@bus.olemiss.edu).

Publisher Item Identifier S 1083-4427(02)06008-3.

of performance and determination of time complexity. The third aspect is related to the *behavior* of the heuristic search process. There have been some works in studying search mechanisms and problem structures over the past years to touch on this aspect in one way or another. Elmaphraby [7] introduced the idea of representing graph-theoretically the structure of a generating mechanism in relation to the state space. Evans [8] examined the neighborhood structures for a problem class from the viewpoint of state-space graphs in artificial intelligence, and showed that state-space analysis can provide useful insights into the nature of local search algorithms. Grover [9] studied the neighborhood structures of several NP-complete problems and showed that these problems satisfy a linear equation that is similar to the wave equation of mathematical physics. Gent and Walsh [10] studied a local search heuristic for the satisfiability problem, plotted the solution values against the iteration number, and found that the plot is almost identical from problem to problem.

Since many heuristics are based upon intuition and involve randomness, the behavior of a heuristic search process looks erratic, like the behavior of a system strongly influenced by random “noise” or the complicated behavior of a system with many degrees of freedom. The technical approach of this paper views the heuristic search process as a *nonlinear dynamical system*. Fig. 1 illustrates a conceptual framework for describing a heuristic search system.

In general, an instance of a problem consists of all the inputs that are needed to compute a solution to the problem and satisfies whatever constraints imposed in the problem statement. The problem instance input is the object of search, which can influence the solution output in some manner through the dynamical behavior of the search process. In this paper, the TSP is chosen to serve as a problem testbed. This paper generates a symmetric TSP instance, which consists of 1000 cities with distance $d(i, j)$ independently drawn from a uniform distribution of the integers over the interval [1, 1000].

The search process in a heuristic search system is carried out by a heuristic search algorithm, called a *search operator*. Thus, the behavior of a search process is determined by the properties of the heuristic operator used in the search system. Many heuristics are problem-specific, so that a method that works well for one problem cannot be used to solve a different one. This paper chooses a general-purpose heuristic search algorithm, namely, the Or-opt algorithm [11].

Or-opt heuristic is a local search technique, which explores a subset of feasible solutions by repeatedly moving from the current solution to a neighboring solution. For each solution $s_i \in S$, a neighborhood $N(s_i) \subset S$ is defined, consisting of all configurations that can be reached from s_i in one transition. The search operator searches the defined neighborhood $N(s_i)$ of s_i for a better solution. Having found one, the process restarts from the new configuration. It is repeated until no further improvement can be found in the neighborhood of the current configuration. Then the system terminates at a local optimum that may deviate from a global optimum.

The output of a search system is the best solution value found during the search. All local search algorithms have the iterative descent property. A local search process generates a series of

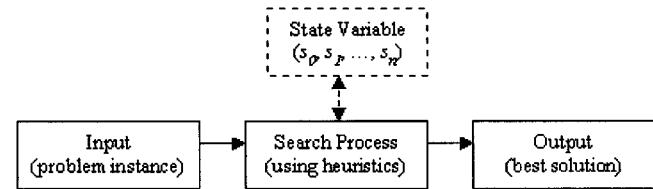


Fig. 1. Heuristic search system takes a problem instance as input, uses a heuristic algorithm or hybrid algorithm to search for solutions, and outputs the best solution found in the search process. The state variable characterizes the internal dynamics of the search system and is usually used to control the search process and to approach the final solution.

solution points, each new points being calculated on the basis of the current configuration. This property makes a local search system locally convergent, that is, it generates a sequence that converges to a point that may be not an optimal solution.

A state variable is the characteristic variable that directly measures the internal dynamics of a system. The search operator determines the possible changes in the position of solution space in a heuristic search system. For the TSP, the length associated with a set of connected edges forms the state variable. The state variable is used to measure the current search result, control the search process, and represent the dynamics of the search system by the positions of the system in the solution space.

III. RECONSTRUCTION OF LOCAL SEARCH TRAJECTORY

A. The Time Series Data

Typically, a local heuristic search system selects an initial solution vector s_0 in solution space S and then generates a new vector s_1 from s_0 . If s_1 is an improved vector, the search operator keeps it. A new solution vector s_2 is generated from s_1 . If s_1 is not an improved vector, the operator discards it. A new vector s_2 is still generated from s_0 . The process is repeated and a new solution vector s_3 is generated and tested. Continuing in this fashion, a sequence of vector points $A = \{s_0, s_1, \dots, s_i, \dots, s_n\}$ is generated, and a sequence of ever-improved points $B = \{s_0, \dots, s_j, \dots, s_m\}$ approaches a final solution vector x^* . It is obvious that B is a subset of A . In fact, the set A defines the search path and can be served as the representation of solution space for the problem.

The dynamical information of an experimental process is often contained in an observed time series. A *time series* is a sequence of data representing the evolution of an observable quantity of the process, indexed by time. For any system, if we set an initial state point $s_0 = (t_0)$ at time t_0 , and record the state variable $s_1 = s(t_1)$ at time t_1 , $s_2 = s(t_2)$ at time t_2 , and so on, the state variable should form a time series

$$s_i = s(t_i) \quad i = 0, 1, 2, \dots \quad (2)$$

Nonlinear dynamics theory shows that a set of sampled values of just *one* variable can be sufficient to capture the features of a system’s dynamics [12]. In other words, we can reconstruct the information given in the time series to present information about the whole system. In fact, if the time series are used cleverly, the essential features of the dynamics in state space can be reconstructed. From the single-variable time series, we can also

determine the number of state variables needed to specify the state of the system. For the local search system in this paper, the values of tour length recorded at each iteration yield the following sequence of numbers:

$$s_0 = s(t_0), s_1 = s(t_1), \dots, s_i = s(t_i), \dots, s_n = s(t_n) \quad (3)$$

where s_0 is some initial solution value and s_i is a solution value after the i th iteration. If we consider the solution value as the state variable and one iteration as a time interval, this sequence forms our experimental time series. This paper collects 10 000 data points (first 10 000 iterations) from the Or-opt search system, running on the TSP instance. For convenience, let D denote the data set of 10 000 points, and let $D1, D2$, and $D3$ denote the data subset of the first 1000 points, 4000–5000 points, and 8000–9000 points, respectively.

Fig. 2 plots the time series D . In the horizontal axis, the number of iteration (time) is marked, while on the vertical axis, the values of solution are given for each iteration. The plot shows an irregular curve with a certain amount of structure. The search pattern of the heuristic search system shows a movement from a hill-climbing phase to a plateau phase, indicating that the performance of the heuristic search slows down as a local optimal point is approached.

B. Time-Delay Coordinates

In scientific experiments, the behavior of an experimental process is usually ruled by k -coupled autonomous differential equations, thus the state space for the experimental process is \mathbb{R}^k . Each vector v_i represents a possible state of the process. Of course, fundamental understanding of a dynamical process can be achieved by building a dynamical model from the whole set of differential equations. However, tracking the differential equations is not easy. In an experimental setting, it is seldom the case that all relevant dynamical variables can be measured. The set of coupled differential equations that governs the behavior of the process would be unknown. The state space dimension k would be very large [13], [14]. Thus, it is very difficult to model fully the dynamics of the process.

Usually, the first step to analyze the dynamics of a system is to reconstruct its trajectories in a state space. A trajectory of a dynamical process is the path in state space followed by the process over time. The philosophy of reconstructing trajectories is to extract “physical sense” from an experimental data. One useful graphical device to reconstruct a trajectory from a time series is the *time-delay coordinates* [15], [16]. We plot each value of the time series versus its time-delay version, by plotting $\{s_t, s_{t+T}\}$ for a fixed time-delay T . This procedure is called *time-delay coordinates reconstruction* [15]. This technique is to reproduce the set of dynamical states of a process using vectors derived from a time series measured from the process. An interesting feature of the reconstruction is that the reconstructed trajectory in m dimensions is topologically equivalent to the actual trajectory of the process. This method had been applied to a wide range of dynamical systems and has gained acceptance as a powerful analysis tool for investigating dynamical processes in many scientific and research communities. In fact, this time-delay technique is currently the most widely used choice for state space

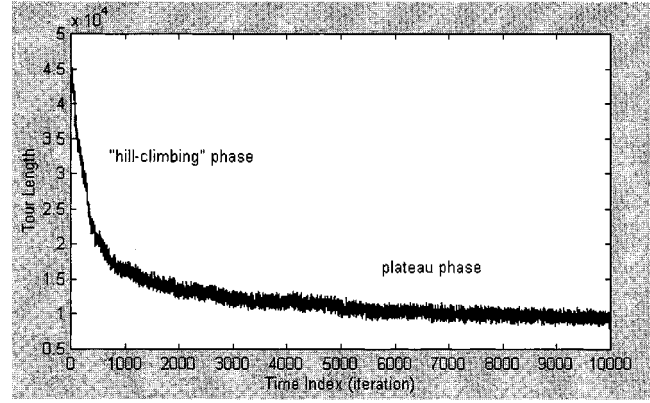


Fig. 2. Time series plot for the Or-opt heuristic search system on the TSP.

reconstruction, which enables us to retrieve the geometric structure of any nonlinear dynamical process [17]–[19].

The theoretical basis for the time-delay reconstruction is the embedding theory [14], [16], [20], which describes the connections between a system’s full state space, the measurements that comprise a time series, and the reconstructed state space. The embedding theory shows that points on the dynamical trajectory in the full system state space have one-to-one correspondence with measurements of a limited number of variables. A one-to-one correspondence means that the state space can be identified by the measurements. Thus, we do not need the derivatives to form a coordinate system in which to capture the geometric structure of a trajectory in state space for a dynamical process. Instead, we can directly use the time-lagged variables $s(t+T) = s(t_0 + (t+T)\tau)$ from a single-variable time series to reconstruct a state space to view the dynamics of the process. Specifically, using a collection of time lags to create a sequence of vectors in m dimensions

$$\begin{aligned} &(s(0), s(T), s(2T), \dots, s((m-1)T)) \\ &(s(\tau), s(\tau+T), s(\tau+2T), \dots, s(\tau+(m-1)T)) \\ &(s(2\tau), s(2\tau+T), s(2\tau+2T), \dots, s(2\tau+(m-1)T)) \\ &\dots \\ &(s(t\tau), s(t\tau+T), s(t\tau+2T), \dots, s(t\tau+(m-1)T)) \end{aligned} \quad (4)$$

where T is a time-delay, and τ is the time interval, we can provide the required coordinates to reconstruct the trajectory for a dynamical process in m dimension [18]. In general, rather than considering $s(t)$ as a sequence of single values, we group together m successive values for each time t in the time series, thus the vector

$$v(t) = [s(t), s(t+T), s(t+2T), \dots, s(t+(m-1)T)] \quad (5)$$

represents a point in an m -dimensional space the coordinates of which are just these m values. Here, T denotes a time-delay, which must be a multiple of time interval τ , and m , called the *embedding dimension*, denotes the dimension of the coordinates for a reconstructed trajectory.

Theoretically, trajectories of a dynamical process can always be faithfully reconstructed using this procedure. In practice, what time-delay T to choose and what embedding m to use are the central issues of the reconstruction.

C. Choosing Time-Delay T

Most of the research conducted on the problem of state space reconstruction has centered on theoretical approaches to choosing T and m in an optimal fashion for time-delay coordinates. The proposed approaches are related to either information-theoretic property [21]–[23] or geometrical property [24], [25].

Takens' study implies that any time-delay T may be acceptable [16], [20]. This means that the choice of the time-delay T is almost arbitrary. However, in practice, if T is too small, the coordinates $s(t)$ and $s(t+T)$ that we wish to use in the reconstruction of data vector $v(t)$ will not be independent enough. The resulting trajectory will be close to the "diagonal" of the state space. On the other hand, if T is very large, any connection between the measurement $s(t)$ and $s(t+T)$ will be numerically subject to being random with respect to each other. The reconstructed trajectory will appear to wander all around space such that the structure of the trajectories is hard to detect. Thus, in practice, the quality of the reconstruction depends on the value chosen for T .

Therefore, some prescription is needed to identify a proper time-delay T that is large enough that $s(t)$ and $s(t+T)$ are quite independent, but not so large that they are completely independent in a statistical sense. There are a number of suggested techniques for choosing T [26]. The *average mutual information* approach can be used to obtain a proper, but not necessarily optimal, time-delay T [23], [27].

Mutual information measures the general dependence of two variables. Suppose a process consisting of a set A and a set B of possible measurements, the *mutual information* between measurement a_i drawn from set $A = \{a_1, a_2, \dots, a_i, \dots, a_n\}$ and measurement b_j drawn from set $B = \{b_1, b_2, \dots, b_j, \dots, b_m\}$ is the amount learned by the measurement of b_j . Let P map measurements to probabilities. The amount one learns about a measurement of a_i from a measurement of b_j is given by the arguments of information theory as

$$I_{AB}(a_i, b_j) = \log_2 [P_{AB}(a_i, b_j) / P_A(a_i)P_B(b_j)] \quad (6)$$

where

$$\begin{aligned} P_A(a_i) & \text{probability of observing } a_i \text{ out of the set } A; \\ P_B(b_j) & \text{probability of finding } b_j \text{ in the set } B; \\ P_{AB}(a_i, b_j) & \text{joint probability of measurements } A \text{ and } B \text{ resulting in values } a_i \text{ and } b_j \text{ [28], [29].} \end{aligned}$$

If the measurement a_i drawn from A is completely independent of the measurement b_j from B , then $P_{AB}(a_i, b_j) = P_A(a_i)P_B(b_j)$ and thus the mutual information is zero.

Since a mutual information function is a numerical function of the elements in the sample space, it can be treated as a kind of random variable. In particular, mutual information has a mean value and a variance. The mean value, called the *average mutual information* between measurements A and B , is the average over all possible measurements of $I_{AB}(a_i, b_j)$

$$I_{AB} = \sum_{i=1}^n \sum_{j=1}^m P_{AB}(a_i, b_j) \log_2 [P_{AB}(a_i, b_j) / P_A(a_i)P_B(b_j)]. \quad (7)$$

To place this abstract definition into the context of time-delay in which measurements $s(t)$ at time t are connected in a theoretic information fashion to measurements $s(t+T)$ at time $t+T$, the values of the observed $s(t)$ are taken as the set A , and the values of $s(t+T)$, as the set B . Then, the average mutual information between $s(t)$ and $s(t+T)$ measurements is

$$I(T) = \sum_{t=1}^N P(s(t), s(t+T)) \log_2 \left[\frac{P(s(t), s(t+T))}{P(s(t))P(s(t+T))} \right] \quad (8)$$

and $I(T) \geq 0$ [17], [28]. The implication of function $I(T)$ is that, from the one-dimensional (1-D) time series, we want to determine a characteristic time T after which the system has lost an essential part of its information about its previous state. In other words, the quantity T tells us how much additional information about the system we gain through a measurement of $s(t)$ given that we know the results of measurement of $s(t+T)$.

When plotting $I(T)$ against T , the curve for the average mutual information $I(T)$ has a large peak around $T = 0$ due to self-correlation. For larger T , this curve generally decays toward zero. However, very often this decay may be oscillatory and may occur over a long range, indicating a nontrivial correlation. In this case, Frazer and Swinney [23] have shown that the "optimal" choice of T corresponds to the first local minimum in the average mutual information function $I(T)$, at which point, the values of $s(t)$ and $s(t+T)$ are independent of each other enough to be useful as coordinates in a time-delay plot but not so independent as to have no connection with each other at all. With the condition of minimal mutual information between $s(t)$ and $s(t+T)$, the time-delay T for reconstruction provides maximal independence between adjacent coordinate axes $s(t)$ and $s(t+T)$. The first minimum of mutual information is a fair criterion for an efficient reconstruction of state space dynamics.

The average mutual information statistic is a useful and practical tool to build up reconstructed state space vectors. It is easy to evaluate the average mutual information between measurement $s(t)$ and time-lagged measurement $s(t+T)$ directly from a time series. However, this approach is only a prescription and provides a general rule of thumb as a guide to choose a time-delay T that is workable. It has not been proven to be a reliable indicator of optimal time-delay. If the average mutual information function $I(T)$ has no local minimum, Abarbanel *et al.* [18] suggest using $T = 1$ or 2 , or choosing T such that $I(T)/I(0) \approx 1/5$.

This study evaluates the average mutual information $I(T)$ from time-delay $T = 0$ to $T = 20$. Fig. 3 illustrates the curves of average mutual information $I(T)$ for the data sets of D , $D1$, $D2$, and $D3$. According to these curves, the proper values of time-delay for D , $D1$, $D2$, and $D3$ can be determined as $T = 12$, $T = 6$, $T = 9$, and $T = 14$, respectively.

D. Choosing the Embedding Dimension m

The embedding technique is based on the notion that even for a multivariate dynamical process, the single-variable time series is often sufficient to determine many of the properties of full dynamics of the process. The basic scheme is using the time series

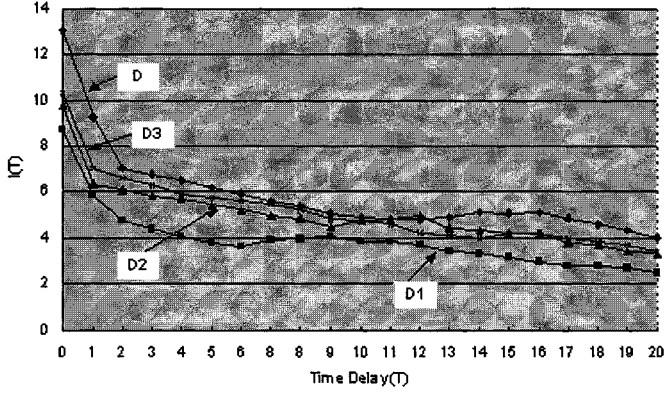


Fig. 3. Average mutual information $I(T)$ for data sets D , $D1$, $D2$, and $D3$. The first local minimum values for D , $D1$, $D2$, and $D3$ can be found at time-delay $T = 12$, $T = 6$, $T = 9$, and $T = 14$, respectively.

to create a multidimensional *embedding space*. If the embedding space is generated properly, the behavior of dynamical trajectories in the embedding space will have the same geometric and dynamical properties that characterize the actual trajectories in the full multivariate state space for the dynamical process. In a sense, the evolution of the trajectories in the embedding space mimics the behavior of the actual trajectories in the full state space.

The number of embedding dimension m is determined by asking when projection of the geometrical structure of state space has been completely unfolded. *Unfolding* means that points lying close to one another in the state space of $v(t)$ vectors do so because of the dynamics and not because of the projection. The purpose of time-delay reconstruction embedding is to unfold the projection back to a multivariate state space that is representative of the original dynamical process. Thus, the embedding dimension m is defined as the lowest integer dimension that unfold an observed trajectory from self-crossings arising from projection of higher dimensional process into lower dimensional space [17], [30].

There are a few approaches to determine an embedding dimension m [18]. This paper uses the *global false nearest neighbors (GFNNs)* method proposed by Kennel *et al.* [30]. In an embedding dimension that is too small to unfold the trajectory, there are some points that lie close to one another by virtue of having projected the geometric structure of the trajectory down into a smaller space. Suppose we have made a state space reconstruction in an embedding dimension d with vectors

$$v(t) = [s(t), s(t+T), \dots, s(t+(d-1)T)] \quad (9)$$

using the time-delay T suggested by the average mutual information. Each vector $v(t)$ has a nearest neighbor $v^{\text{NN}}(t)$

$$v^{\text{NN}}(t) = [s^{\text{NN}}(t), s^{\text{NN}}(t+T), \dots, s^{\text{NN}}(t+(d-1)T)]. \quad (10)$$

If the vector $v^{\text{NN}}(t)$ lies close to $v(t)$ because of the dynamics, $v^{\text{NN}}(t)$ is a *true* neighbor of $v(t)$. If $v^{\text{NN}}(t)$ is actually far from $v(t)$ but simply appears as a neighbor of $v(t)$ by projection from a higher dimension into too low a dimension, the vector $v^{\text{NN}}(t)$ is a *false* neighbor of $v(t)$. By examining this false neighbor in dimension $d+1$, then going to dimension $d+2$, etc., we will

eventually move it out of the neighborhood of $v(t)$. For every data point $v(t)$, if we examine false neighbors in dimension one, then in dimension two, and so on, until all false neighbors are eliminated, at that juncture we will have identified the necessary embedding dimension m for unfolding. In going from dimension d to $d+1$ by time-delay embedding scheme, we compare the distance between the vectors $v(t)$ and $v^{\text{NN}}(t)$ in dimension $d+1$ with the distance between the same vectors in dimension d . If the increase in distance between $v(t)$ and $v^{\text{NN}}(t)$ is larger than some defined threshold when going from dimension d to $d+1$, we have a false neighbor; otherwise, we have a true neighbor.

The square of the Euclidean distance between the point $v(t)$ and its nearest neighbor $v^{\text{NN}}(t)$ in dimension d is

$$R_d(t)^2 = \sum_{n=1}^d [s(t+(n-1)T) - s^{\text{NN}}(t+(n-1)T)]^2 \quad (11)$$

while the distance between $v(t)$ and the same nearest neighbor $v^{\text{NN}}(t)$ in dimension $d+1$ is

$$\begin{aligned} R_{d+1}(t)^2 &= \sum_{n=1}^{d+1} [s(t+(n-1)T) - s^{\text{NN}}(t+(n-1)T)]^2 \\ &= R_d(t)^2 + [s(t+dT) - s^{\text{NN}}(t+dT)]^2. \end{aligned} \quad (12)$$

The criterion for deciding whether the neighbor is true or false can be represented by

$$\sqrt{\frac{R_{d+1}(t)^2 - R_d(t)^2}{R_d(t)^2}} = \frac{|s(t+dT) - s^{\text{NN}}(t+dT)|}{R_d(t)} > R_T \quad (13)$$

where R_T is some threshold. This criterion says that, when the difference between the distance in dimension $d+1$ and the distance in dimension d relative to the distance in dimension d is larger than the threshold R_T , the neighbor is a false neighbor in dimension d .

The generic decision basically involves the threshold R_T . R_T should be defined such that the false neighbors are clearly identified. In practice, for values of threshold R_T in the range $10 \leq R_T \leq 50$, the number of false neighbors identified by the criterion is almost constant [18]. Abarbanel [17] suggested that, for a large variety of systems, a workable threshold R_T to define a false neighbor is a number of approximately 15. We examine each of total N points on the trajectory to determine how many of the N nearest neighbors are false. We record the results of the computations as the proportion of all data points that have a false nearest neighbor. If a clean (without “noise”) time series is presented, it is expected that the percentage of false nearest neighbors will drop from nearly 100% in dimension one to strictly zero when the embedding dimension m is reached, and remains zero from then on. In practice, however, the observed data is often with “noise.” Furthermore, as one increases the number of data points analyzed, the embedding dimension can systematically increase [18], [30]. To account for this, a second criterion for falseness of a nearest neighbor is added to the previous one. That is, if

$$|s(t+dT) - s^{\text{NN}}(t+dT)|/R_A \geq 2 \quad (14)$$

then the point $v(t)$ and its nearest neighbor $v^{NN}(t)$ would be labeled as “false nearest neighbor.” Here, R_A is the size of a trajectory and can be calculated from

$$R_A^2 = \frac{1}{N} \sum_{t=1}^N \left[s(t) - \frac{1}{N} \sum_{t=1}^N s(t) \right]^2 \quad (15)$$

and N is the number of data points.

This paper applies these two criteria jointly: meeting either of these two criteria indicates that the change of distance between points $v(t)$ and $v^{NN}(t)$ in going from dimension d to dimension $d + 1$ is too large and the neighbor $v^{NN}(t)$ would be declared false. For the computation, this paper uses the value $R_T = 15.0$ and the dimension going from one to ten. Using the time-delay T determined in the previous section, the values of the percentage of GFNNs for data sets D , $D1$, $D2$, and $D3$ are calculated and then plotted in Fig. 4. The value of GFNNs for the data D drops to zero at $m = 4$, and remains zero after that. This means that the corresponding search trajectory of data D can be reconstructed in an embedding dimension $m = 4$ to represent faithfully its geometric structure.

Let us look at the time series $D1$, $D2$, and $D3$ more closely. From the curve of global false neighbors for $D1$, we can see that, as the dimension increases, the percentage of false neighbors drops and then rises, which indicates that this data comes from a dynamical process with very high dimension. However, the curves for $D2$ and $D3$ shows that the embedding dimension for the corresponding trajectories of $D2$ and $D3$ are $m = 6$ and $m = 5$, respectively. This fact indicates that the dynamics of the heuristic search process changes over time. More specifically, the dimensionality of the local heuristic search process decreases over time, from high-dimensional to low-dimensional dynamics.

E. Reconstruction Results

The reconstruction of a trajectory can be interpreted as a process of coordinate change. Often the trajectory of a dynamical process is originally defined in some high-dimensional space. The reconstruction amounts to a projection of the original to a smaller dimensional Euclidean space. The chosen dimension m for the embedding space guarantees that each point in the projected trajectory corresponds to one and only one point in the original trajectory. In other words, the reconstruction is a truthful representation of the original trajectory. There is no image where parts of the trajectory are collapsed onto each other. Thus, we should be able to extract full information about the geometric structure from a single-variable time series, using the time-delay reconstruction.

One of the problems of state space representation is the deficiency in our brains that prevents us from visualizing n dimensions for n greater than three. Because of this limitation, two-dimensional (2-D) or three-dimensional (3-D) space is still a common medium for communication about n -dimensional problems worked out in our heads. This paper reconstructs the heuristic search trajectory on a 3-D plot, using the time-delay T suggested by the average mutual information; even the calculated values of the embedding dimension m are larger than three. The 3-D reconstructed trajectory is not identical to the

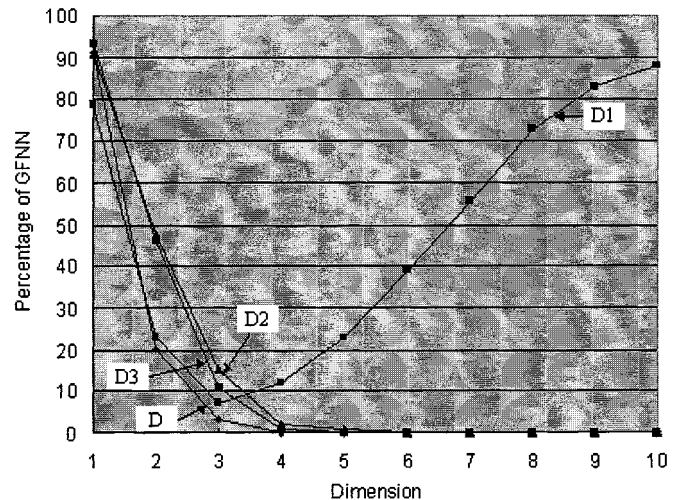


Fig. 4. Curves of global false nearest neighbors (GFNNs) for D , $D1$, $D2$, and $D3$. The embedding dimensions for D , $D2$, and $D3$ are $m = 4$, 6 , and 5 , respectively. The embedding dimension for $D1$ is very high.

original but more or less a distorted copy. This kind of dimensional reduction is forced on us by our presentation instrument and also is employed as a way of reducing complexity since our brains are so limited. Certainly, some information about the trajectory is lost in this dimension-reduced representation. However, just because some information is lost does not mean that the visual representation is useless. The reconstructed trajectory still provides information about the dynamical structure of the process.

Fig. 5 portrays the reconstructed trajectories for the data sets D , $D1$, $D2$, and $D3$. These portraits provide a geometric representation of complex dynamics of the local search system. The search trajectory has a loose structure at beginning [part (b)], which means the search process has more opportunities to find better solutions and thus has fast convergent speed. The middle part of the trajectory [part (c)] shows that the structure becomes tight, which means that the process has to take longer time searching neighbors to find a better solution. The lower part ([part (d)] reveals even a tighter structure, like a thread ball, which means that the search activity is restricted in an even smaller domain.

Data visualization is a process of transformation of data into a picture, which engages the interaction of human visual system with the spatial organization of the data. The result is a simple, effective, and eye-pleasing graphic medium for communicating complex information. The time series generated by the Or-opt system can be used to visualize the solution space explored by the search process. This time series has three characteristics. First, it is discrete. The implication of this characteristic is that the character of the data between discrete points unknown. However, we often need information at positions other than the supplied points for graphic rendering. One of important visualization activities is to determine data value at an arbitrary position. Interpolation is a widely used technique. This method interpolates data from points to some intermediate point using interpolation functions that presume a relationship between neighboring data values. Often this is a linear function, but it can be a quadratic, cubic, spline, or other function.

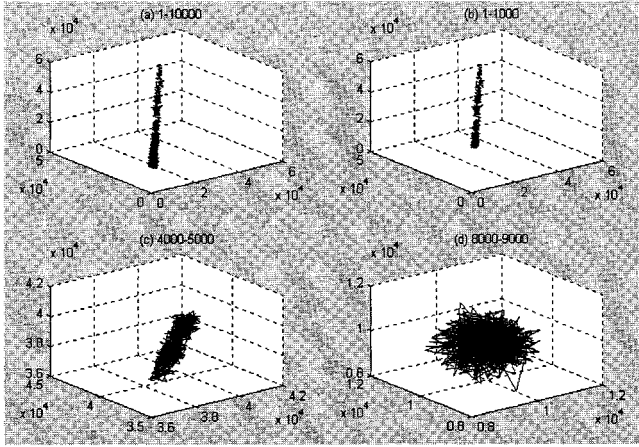


Fig. 5. Search trajectory of the Or-opt heuristic system. (a) shows the search trajectory for data D , (b) shows the trajectory for data $D1$, which is a subset of D , and (c) and (d) show the trajectories for data $D2$ and $D3$, respectively.

Another characteristic of the data is its irregularity. The points in an irregular data set are irregularly located in space and have no inherent structure. The geometry is unstructured. An irregular data set must be explicitly represented. This requires greater memory and computational resources. However, the advantage of irregular data is that irregular data gives us more freedom in representing it. Data can be represented in such a way that no regular patterns can be found in topology or geometry. However, for the purpose of visualization, an irregular data is typically transformed into a structured form.

Finally, the third important characteristic of the data is its dimension. For example, as calculated in the previous section, the embedding dimension for data $D1$ is very high. However, people usually use a surface in three dimensions to represent a state space.

Data visualization deals with the issues of data transformation and representation. Transformation is the process of converting data from its original form into computer image. Representation is both the internal data structure used to depict the data and the graphics used to display the data. In this paper, the structural representation of the data is a matrix of solution values, while the graphical representation of the data is a surface in 3-D space. Thus, the solution space is created in a 3-D Cartesian space from a time series that is actually embedded in a higher dimension. Each point is expressed as a triplet of values (x, y, z) along the x -, y -, and z -axes. The primary task of visualization in this paper is to transform the data from scalars into vectors and to specify data geometry (i.e., point coordinate). The time series data are sampled on a regular $k \times k$ matrix. Suppose the size of the time series is N , the $k \times k$ matrix is created such that $k^2 \leq N$. One matrix defines the values along only one axis. Therefore, three $k \times k$ matrices, X, Y , and Z , are created to specify the solution surface. These three matrices provide a structured frame for defining the position of the vectors. Specifically, matrices X, Y , and Z together return a list of triples specifying vector points in 3-D space, which in turn defines rectangular grids on the surface. Each point in the rectangular grid can be thought of as connected to its three nearest neighbors. This underlying rectangular grid induces four-sided patches on the surface. Then,

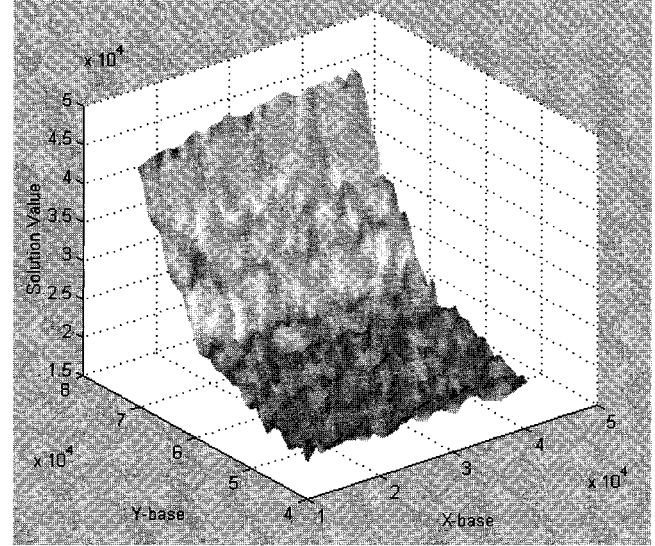


Fig. 6. Solution surface space explored by the Or-opt heuristic search system during its first 1000 searches.

the patches are filled with color. This paper uses the ready-made 3-D surface visualization tools in MATLAB. Fig. 6 shows the solution surface built from the data $D1$, which can be considered as the local solution space searched by the Or-opt search system during its first 1000 iterations.

IV. THE DYNAMICS OF THE SEARCH SYSTEM

A. The Correlation Dimension

Generally speaking, the analysis of dynamical behavior is extremely difficult. The goal of trajectory reconstruction is to capture the geometric pattern of points on a trajectory of a dynamical process [15]. There is also a need to define quantitatively dynamical properties of heuristic search system.

In characterizing a dynamical process quantitatively, the emphasis is on the time-dependent dynamical behavior of the system and the geometric nature of the trajectories in the state space. One approach to characterize the temporal dynamics of a system is to calculate the dimension of the process trajectory. The dimensionality of the trajectory is closely related to dynamics of the system [31], since the dimensionality of a process gives us an estimate of the number of active degrees of freedom in the system. This paper uses the *correlation dimension* technique to investigate the dynamical structure of the Or-opt search system, and shows how to extract a multidimensional description of dynamics of a solution space from a single-variable time series.

The determination of the fractal dimension of a trajectory has become a standard diagnostic tool in the analysis of a dynamical system. The dynamical complexity of a system can be described by the calculated dimension of its trajectory. The most recent work in characterizing dynamical process has focused on the evaluation of fractal dimension by calculating the correlation dimension. The correlation dimension is originally due to Grassberger and Procaccia [32]. It becomes a popular technique because of its simplicity, computational speed, and minimal storage requirement.

Correlations between points on a trajectory are defined in terms of spatial correlation that can be formally measured by the Euclidean distance. The correlation dimension considers the statistics of distances between pairs of points [33]. The procedure of estimating the correlation dimension from a single-variable time series is as follows.

- 1) The time series are embedded in successively increasing dimensions as prescribed by Takens [34]. The series of scalars is converted into a series of vectors in each embedding dimension d_E .
- 2) For each embedding dimension, the correlation integral $C(r)$ is computed and then the correlation exponent k is estimated.
- 3) The procedure is repeated until the estimates of k converge.
- 4) Whatever value at which k remains stable over a number of embedding dimensions is taken as the estimate of the correlation dimension D_C for the underlying trajectory.

The correlation integral $C(r)$ measures the spatial correlation among the points on a trajectory. Specifically, for each $r > 0$, the correlation function $C(r)$ is defined as the average number of pairs of data vector (v_i, v_j) that are separated by a distance less than r

$$C(r) = \lim_{N \rightarrow \infty} \frac{1}{N} \left\{ \text{the number of pairs } (v_i, v_j) \text{ such that } \|v_i - v_j\| < r \right\} \quad (16)$$

where N denotes the number of points on the trajectory. In principle, N should go to infinity, but in practice N is limited by the availability of the time series points. The $\|\cdot\|$ denotes the distance induced by the selected norm. According to Theorem 2.4 of Brock [35], the correlation dimension is independent of the choice of norm. Thus, the Euclidean distance is used in this paper.

Intuitively, $C(r)$ measures the probability that any particular pair in the time series is close. Grassberger and Procaccia [32] have established that the true spatial correlation grows according to the power law, that is, for small values of r , $C(r)$, grows exponentially at the rate of k

$$C(r) \approx r^k \quad (17)$$

where k is the correlation exponent. If k tends to be a constant as embedding dimension d_E increases, then k yields an estimate of the correlation dimension of the trajectory. In this case, the data are consistent with deterministic behavior. If k increases without bound as the embedding dimension d_E increases, this suggests that the data is stochastic as the dimension of the dynamical process approaches infinity.

Therefore, the correlation dimension D_C is defined to be the number that satisfies

$$C(r) = \lim_{r \rightarrow 0} r^{D_C}. \quad (18)$$

Then, the correlation dimension D_C is obtained by

$$D_C = \lim_{r \rightarrow 0} \ln C(r) / \ln r \quad (19)$$

if the limit exists.

In (16), $C(r)$ is defined as a function of r such that $C(r)$ increases from zero to one as r increases from zero to ∞ . In other words, if all the data points are within the distance r of each

other, then $C(r) = 1$. If r is smaller than the smallest distance between trajectory points, then $C(r) = 0$. Since any real data set consists of a finite number of points, it is not possible to take the limits $N \rightarrow \infty$ and $r \rightarrow 0$. Hence, in practice, the correlation dimension is found using some range of r values and plotting $\ln C(r)$ as a function of $\ln r$. In this way, a plot of $\ln C(r)$ versus $\ln r$ is produced. We can observe that only in some intermediate r region does $C(r)$ obey the power law expressed in (18) [36]. This region is called the *scaling region*, and the curve in this region is approximately a straight line. The slope for the curve determines the correlation dimension D_C from (19) without the limit $r \rightarrow 0$.

Since in practice the presence of noise in a data is likely to result in a choppy appearance in the curve of $\ln C(r)$ versus $\ln r$, we often need to run a linear regression on the curve. The slope estimate from this regression is the correlation dimension estimate for this value of d_E . Therefore, there are some uncertainties involved in calculating D_C , which can cause the computed dimension to differ slightly from the “true” value.

This procedure is repeated for a sequence of increasing values of d_E . If the time series are purely random, then in infinite data sample, the correlation dimension will equal d_E . In practice, with a finite data set, the correlation dimension estimate may rise with d_E at less than a one-for-one rate [37]. If the time series are deterministic, the correlation dimension D_C should initially increase with increasing values of d_E . However, at some point, d_E, D_C should level off and remain constant for all further values of d_E . One should observe a “plateau” on the plot of the correlation dimension $C(r)$ in the function of embedding dimension d_E . This “saturation” value of the slope is the correlation dimension estimate for the underlying system that generates the data.

In practice, the slope estimate may never completely stop rising. Thus, the process of determining whether saturation has occurred can, in general, be rather judgmental. Ding *et al.* [38] reported that, for a large enough data set, the saturation begins when the number of embedding dimension d_E first exceeds the correlation D_C . However, a lack of sufficient data will delay the saturation onset. Taken’s theorem [34] provides a practical clue. The embedding theorem shows that if the true system that generated the time series is in d dimension, the embedding space needs to have $d_E \geq 2d + 1$ dimension to capture completely the dynamics of the underlying system. Since d is unknown, an increasing sequence of d_E values must be tried. Usually, d_E has to be increased to nearly $2d + 1$ before the value deduced for d become independent of d_E .

B. Empirical Results

Fig. 7 illustrates a plot of the natural logarithm of the correlation integral as a function of $\ln r$ for different values of d_E for the data D . The curves correspond to computation of the correlation function in embedding dimension $d_E = 1, 2, \dots, 10$. The values of r are experimentally determined in the following manner. Let R_A represent the size of the trajectory under consideration, which can be obtained from (15), then r is calculated as

$$r = R_A(\alpha)^k \quad (20)$$

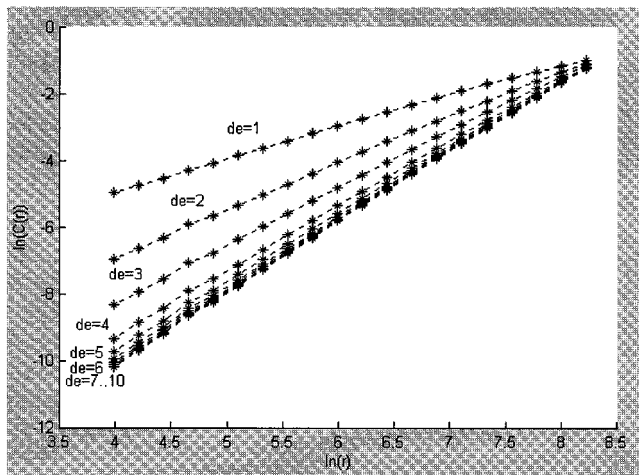


Fig. 7. Correlation integers in different embedding dimension $d_E = 1, 2, \dots, 10$ for the data D .

TABLE I
THE SLOPE OF EACH CURVE IN FIG. 7

d_E	Slope (D_C)
1	0.5911
2	1.3940
3	1.7012
4	1.9243
5	2.0145
6	2.0617
7	2.0863
8	2.1007
9	2.1110
10	2.1164

where the values of α and k are chosen experimentally to provide a range of r values that would encompass a nearly linear region of the curve. This study selects $\alpha = 0.8$ by using a trial-and-error program. The proportion that lie within the range is counted for $r = R_A(0.8)^{0.2}, R_A(0.8)^{0.4}, R_A(0.8)^{0.6}, \dots, R_A(0.8)^4$, which make 20 points on the curve. In this way, a series of correlation dimensions is calculated over ten embedding dimensions and plotted. Then, the slope for each curve is found by the least squares regression and shown in Table I.

As a practical matter, we search to see whether the values of correlation dimension stabilize at some value D_C as embedding dimension d_E increases. If so, then the value D_C is the correlation dimension estimate. If D_C is a small value, then the process is substantially deterministic even it is complicated. If, however, as d_E increases, the correlation dimension continues to increase at the same rate, then the process is taken to be stochastic. In principle, an independently and identically distributed stochastic system is infinite dimensional. Each time one increases one available degree of freedom, the random system utilizes that extra freedom. Therefore, it is desirable to calculate the correlation dimension for a “random” system to provide a basis for comparison. Fig. 8 compares the correlation dimensions of the time series D with the correlation dimensions of a random data, which consists of 10 000 data points that are

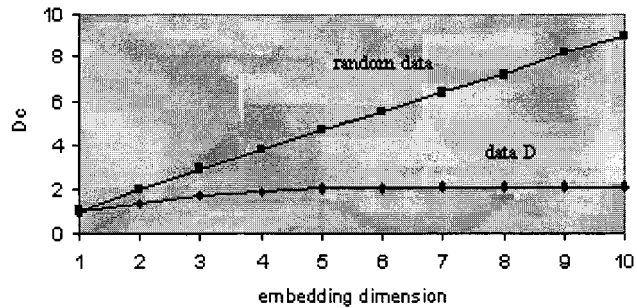


Fig. 8. Comparison of correlation dimensions between the heuristic time series D and the random data.

uniformly distributed over the interval $[1, 1000]$. As shown in Fig. 8, the correlation dimension D_C for the data D stabilizes when the embedding dimension is greater than about twice the correlation dimension of the data, indicating that the correlation dimension becomes independent of d_E . The saturated correlation dimension value $D_C = 2.0145$ for the data D is determined at the corresponding $d_E = 5$. For the random-number data, no saturated correlation dimension can be found.

The correlation dimension estimate for the data D contains information about the dynamics of the heuristic search process during its first 10 000 searches. This value describes an averaged property of the search process. In other words, this correlation dimension is an average measure over the entire search trajectory (10 000 points). However, Fig. 5 reveals that, in the heuristic search process, the hill-climbing phase has a different geometric structure from the plateau phase’s, indicating different levels of dynamics of search motions. The local contributions from different parts of the search trajectory to the average dynamics are, in general, different. Just an average value does not acknowledge all of the complexity of the search process. We ask what happens to the dynamics of the search trajectory if we examine it at different parts. Thus, this study divides the 10000-points trajectory into ten parts, each consisting of 1000 points. It is expected that different parts of the search trajectory to be characterized by different values of the correlation dimension. The heuristic search trajectory should be an object with a multiplicity of fractal dimension. Fig. 9 shows a schematic graph of the correlation dimension as a function of the embedding dimension for five partitions of the heuristic time series. For comparison, the correlation dimension for a 1000-points random data is also plotted.

We can see that, although the curve of correlation dimension for the first 1000 data points are below the curve for the random data, up to $d_E = 10$, there is no sign of saturation, indicating that the time series for this partition are indeed of high, if not infinite, dimension. It indicates the presence of a stochastic search process during the first 1000 searches. However, the correlation dimension of search process decreases as the search continues, indicating that the complexity of search dynamics appears to decrease as the heuristic search is approaching to a local optimal point. In summary, in the course of searching for an optimal solution on the solution space, the dynamics of the heuristic search process changes from a stochastic process to a process with low-dimensional structure.

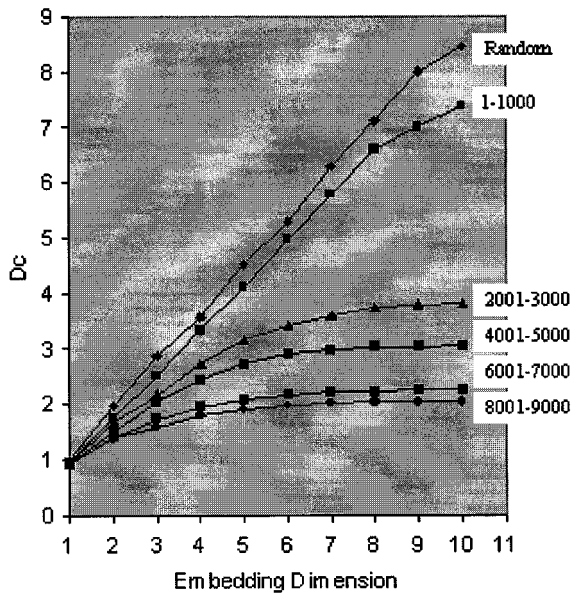


Fig. 9. Correlation dimensions for the partitions of heuristic data and random data.

V. CONCLUSION AND DISCUSSION

This paper reconstructs a search trajectory for the Or-opt heuristic search system on the TSP. The trajectory illustrates the graphical representation of the dynamical process of the search system in its solution space. The critical aspect of the time-delay coordinates is that the dynamical behavior of a local search process can be reconstructed from a time series data. The dynamics of the search process can be seen by the picture of the data. In fact, a collection of many such search trajectories originating from different initial points can portray the entire solution space for the search system.

This paper also uses the correlation dimension technique to quantitatively characterize the complexity of the dynamics for the search system. The results reveal that the heuristic search process exhibits the transition from high-dimensional stochastic to low-dimensional chaotic behavior. The initial phase of search process shows more complexity in dynamics than the later phases of the search. The heuristic search process starts with a high-dimensional searchable space. However, the dimension of the searchable space is rapidly reduced. As the search is approaching to a local optimal point, the searchable space becomes smaller and smaller. This decomplexification of search dynamics appears to be a generic response to perturbation associated with the solution configuration. A local heuristic search process searches for an optimal configuration by repeatedly moving from the current configuration to a neighboring configuration. In the early stage of search, the frequency of this configuration perturbation is very high. However, the possibility of finding a better configuration is less and less as the search goes on. Thus, the loss of complexity, as assessed by the measurement of correlation dimension, can be related with this decreasing possibility. The search process looks very much like passing through many levels of laminar. The search begins with a large laminar with many corridors. It is very easy for the process to find a corridor to pass through to next laminar, which

TABLE II
ITERATIONS AND BETTER SOLUTIONS

Iteration	1000-City	2000-City	4000-City
100	62	71	79
1,000	206	359	466
10,000	242	444	659
100,000	250	471	776
1,000,000	250	476	830

is characterized by smaller size and less corridors. Every time the search process gets into a new laminar, it will take a longer time to find a corridor to pass through to the next laminar. The closer to a local optimal point, the longer the average laminar time will take. One issue related with the loss of complexity is the size of solution space that can be effectively searched. Many degrees of freedom N on the first laminar are reduced to a few degrees of freedom n ($n \ll N$) on the later laminars. Average over the residual degrees of freedom ($N - n$) is increased by the restriction of neighborhoods. The size of solution space that can be reached may decrease exponentially.

This decreasing size of the solution space perhaps can be modeled to describe the search effectiveness for a heuristic search process. Usually, the more iterations we perform the better final solution we can get. But how many iterations are enough? It is perhaps somewhat surprising that there does not yet exit a useful convergence theory for this question. Let us assume that after, say 175 321 iterations, we found another better solution. However, is it the last one we will find it if we continue to iterate? There is no way to tell. However, still we must make a decision. For example, we can propose to perform another 10 000 iterations, and if we do not find any more new better solution, we may be content and stop the iteration. However, probably only the next iteration beyond the 10 000 may prove that we are wrong in this assumption. Table II illustrates the results of an experiment. This experiment applies the Or-opt algorithm on three uniformly distributed TSP instances with 1000, 2000, and 4000 cities, respectively. One million iterations are performed on each instance. The number of better solutions found (i.e., the times of solution configuration perturbations) during 10^k ($k = 2, 3, \dots, 6$) iterations is recorded.

We can see that, for example, for the 1000-city instance, 242 better solutions are found in the first 10 000 iterations and another eight better solutions in the next 90 000 iterations, but no more better solutions found in the next 900 000 iterations. In the 2000-city case, 471 better solutions are found in the first 100 000 iterations and another five better solutions in the next 900 000. How many more iterations should we perform to produce one more better solution? Even if the case of simple local search iterations are very cheap, for a very large problem instance, the number of iterations needed for a satisfied solution may be typically in the order of tens of millions. For many other more sophisticated search heuristics, one million iterations may be well above any reasonable computational time frame. Thus, the problem outlined above really exists in practice, and a solution should be sought. We have known that this problem may be associated with the loss of complexity in the search process. Correlation dimension technique may help us to capture the relevant frequencies or time scales for a particular heuristic search process. Then, an expected number of better

solutions (N) found during n iterations, given the size of the problem (s), may be predicted by a map $N = f(n, s)$. Such a search effectiveness model may answer the question whether we are finding an adequate solution within a time acceptable to the situation, for this size of problem instance, when using this particular search heuristic. Thus, if a heuristic user can expect how many better solutions will be produced during the next, say 100 000 iterations, one has great flexibility in deciding whether or not to trade solution quality for time.

Although this paper investigates the dynamical behavior of the Or-opt heuristic in the TSP, the results appear to be behavioral archetypes for the class of local search heuristics in combinatorial optimization problems generally. There is no strict mathematical definition to describe the behavior of the heuristic search process concisely. The process exhibits complicated patterns in both space and time. Suppose that, for a heuristic search system, the initial conditions are confined to some volume V in solution space bounded by a “surface” of all initial points. As the process evolves, the volume V will move through solution space, which looks like the motion of compressible object. The shape of the volume will be distorted, and the size will be decreased. The final volume occupied by the set of local optimal points is very small. This small volume is called a *solution attractor*. All search trajectories will be attracted to this volume. In fact, there are two competing effects present in the heuristic search process. In one, due to the sensitive dependence on initial conditions, two different search trajectories will diverge. On the other hand, the trajectories must converge to a finite region of the solution space. It is clear that the initial segment of search trajectory is a kind of “coherent randomness.” This randomness is subdued by some modulation. After certain search iterations, this randomness is quickly replaced by a low-dimensional mode of chaos. Then, the search process continues to be suppressed slowly by the modulation over a long period of time. Finally, when the system reaches some local optimal configuration, the chaos is replaced by a quasi-periodic mode of motion.

In the classical picture of statistical mechanics, it is believed that a large number of degrees of freedom are necessary for ergodicity [39]. When a heuristic search process is in the stochastic motion stage, it has the ability to explore wide regions of the solution space, and thus to utilize information about a large collection of potential solutions of the process. When the process gets into the chaotic stage with a few degrees of freedom, it explores a very limited domain of solution space. The search activity now is rather restrictive. The distribution of solution space being searched is far from “equipartition” and is replete with high-probable regions. In summary, a local search process can start from different initial points in solution space and finally wind up at a small attracting region in the solution space. No wonder Boese *et al.* [40], [41] found a “big valley” structure governing the set of local minimums. Their research shows the evidence that “very good solutions are located near other good solutions.” In fact, it is the attracting region that creates a globally convex structure for the set of local minimums in the optimization solution space. This kind of asymptotic behavior of the heuristic search process can have distinct implications for a wide range of theoretical and practical problems in heuristic theory, design, and analysis.

REFERENCES

- [1] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
- [2] L. K. Grover, “Local search and the local structure of NP-complete problems,” *Oper. Res. Lett.*, vol. 12, pp. 235–243, 1992.
- [3] C. H. Papadimitriou and K. Steiglitz, “On the complexity of local search for the traveling salesman problem,” *SIAM J. Comput.*, vol. 6, pp. 76–83, 1977.
- [4] C. R. Reeves, *Modern Heuristic Techniques for Combinatorial Problems*. New York: Wiley, 1993.
- [5] “The design, analysis, and implementation of heuristics,” *Manage. Sci.*, vol. 34, no. 3, pp. 363–429, 1988.
- [6] S. H. Zanakis, J. R. Evans, and A. A. Vazacopoulos, “Heuristic methods and applications: A categorized survey,” *Eur. J. Oper. Res.*, vol. 43, no. 1, pp. 88–110, 1989.
- [7] S. E. Elmaphraby, “A graph theoretic interpretation of the sufficient conditions for the contiguous binary switching (CBS) rule,” *Nav. Res. Logist.*, vol. 18, no. 3, pp. 339–344, 1971.
- [8] J. R. Evans, “Structural analysis of local search heuristics in combinatorial optimization,” *Comput. Oper. Res.*, vol. 14, no. 3, pp. 465–477, 1987.
- [9] L. K. Grover, “Local search and the local structure of NP-complete problems,” *Oper. Res. Lett.*, vol. 12, no. 4, pp. 235–243, 1992.
- [10] P. Gent and T. Walsh, “An empirical analysis of search in GAST,” Univ. of Edinburgh, Artif. Intell. Dept., Edinburgh, U.K., 1993.
- [11] I. Or, “Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking,” Ph.D. dissertation, Northwestern Univ., Evanston, IL, 1976.
- [12] H. Tong and K. S. Lim, “Threshold, autoregression, limit cycles, and cyclical data,” *J. Roy. Stat. Soc.*, vol. 42, no. 3, pp. 245–292, 1980.
- [13] K. T. Alligood, T. D. Sauer, and J. A. Yorke, *Chaos: An Introduction to Dynamical System*. New York: Springer-Verlag, 1997.
- [14] E. Ott, T. Sauer, and J. A. Yorke, Eds., *Coping with Chaos: Analysis of Chaotic Data and the Exploitation of Chaotic Systems*. New York: Wiley, 1994.
- [15] N. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, “Geometry from a time series,” *Phys. Rev. Lett.*, vol. 45, no. 9, pp. 712–716, 1980.
- [16] F. Takens, “Detecting strange attractors in turbulence,” in *Dynamical Systems and Turbulence, Warwick 1980, Lecture Notes in Mathematics*, 898, D. Rand and L. S. Young, Eds. New York: Springer-Verlag, 1981, pp. 366–382.
- [17] H. D. I. Abarbanel, *Analysis of Observed Chaotic Data*. New York: Springer-Verlag, 1996.
- [18] H. D. I. Abarbanel, R. Brow, J. J. Sidorowich, and L. S. Tsimring, “The analysis of observed chaotic data in physical systems,” *Rev. Mod. Phys.*, vol. 65, no. 2, pp. 1331–1392, 1993.
- [19] D. S. Broomhead and G. P. King, “Extracting qualitative dynamics from experimental data,” *Phys. D*, vol. 20, no. 1, pp. 217–221, 1987.
- [20] M. Casdagli, T. Sauer, and J. A. Yorke, “Embedology,” *J. Stat. Phys.*, vol. 65, no. 3/4, pp. 579–616, 1991.
- [21] H. D. I. Abarbanel and J. B. Kadtko, *Information Theoretic Methods for Choosing the Minimum Embedding Dimension for Strange Attractors*. San Diego: Inst. Nonlinear Sci., Univ California, 1989.
- [22] A. M. Frazer, “Information and entropy in strange attractors,” *IEEE Trans. Inform. Theory*, vol. 35, no. 2, pp. 245–262, 1989.
- [23] A. M. Frazer and H. L. Swinney, “Independent coordinates for strange attractors from mutual information,” *Phys. Rev. A, Gen. Phys.*, vol. 33, no. 2, pp. 1134–1140, 1986.
- [24] A. Cenys and K. Pyragas, “Estimation of the number of degrees of freedom from chaotic time series,” *Phys. Lett. A*, vol. 129, no. 4, pp. 227–230, 1988.
- [25] W. Liebert and H. G. Schuster, “Proper choice of the time-delay for the analysis of chaotic time series,” *Phys. Lett. A*, vol. 142, no. 2/3, pp. 107–111, 1989.
- [26] T. Buzug, T. Reimers, and G. Pfister, “Optimal reconstruction of strange attractors from purely geometrical arguments,” *Europhys. Lett.*, vol. 13, no. 7, pp. 605–610, 1990.
- [27] R. S. Shaw, *The Dripping Faucet as a Model Chaotic System*. Santa Cruz, CA: Aerial, 1985.
- [28] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.
- [29] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL: Univ. Illinois Press, 1949.

- [30] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Phys. Rev. A, Gen. Phys.*, vol. 45, no. 6, pp. 3403–3411, 1992.
- [31] R. C. Hilborn, *Chaos and Nonlinear Dynamics: An Introduction for Science and Engineers*. London, U.K.: Oxford Univ. Press, 1994.
- [32] P. Grassberger and I. Procaccia, "Characterization of strange attractors," *Phys. Rev. Lett.*, vol. 50, no. 5, pp. 346–349, 1983.
- [33] T. S. Parker and L. O. Chua, *Practical Numerical Algorithms for Chaotic Systems*. New York: Springer-Verlag, 1990.
- [34] F. Taken, "On the numerical determination of the dimension of an attractor," in *Dynamical Systems and Bifurcations*, N. Breaksma, H. Broer, and F. Takens, Eds. New York: Springer-Verlag, 1985, pp. 99–106.
- [35] W. Brock, "Distinguishing random and deterministic systems: Abridged version," *J. Econ. Theory*, vol. 40, no. 1, pp. 168–195, 1986.
- [36] J. Holzfuss and G. Mayer-Kress, "An approach to error-estimation in the application of dimension algorithm," in *Dimensions and Entropies in Chaotic Systems: Quantification of Complex Behavior*, G. Mayer-Kress, Ed. New York: Springer-Verlag, 1986, pp. 114–122.
- [37] J. B. Ramsey and H. J. Yuan, "The statistical properties of dimension calculations using small data sets," C. V. Starr Center, New York Univ., New York, Res. Rep. 87-20, 1987.
- [38] M. Ding, C. Grebogi, E. Ott, T. Sauer, and J. A. Yorke, "Plateau onset for correlation dimension: When does it occur?," *Phys. Rev. Lett.*, vol. 70, no. 25, pp. 3972–3975, 1993.
- [39] Y. Kuramoto, Ed., *Chaos and Statistical Methods*. New York: Springer-Verlag, 1984.
- [40] K. D. Boese, A. B. Kahng, and S. Muddu, "On the big valley and adaptive multi-start for discrete global optimizations," Comput. Sci. Dept., Univ. California, Los Angeles, Tech. Rep. TR-930 015, 1993.
- [41] —, "A new adaptive multistart technique for combinatorial global optimizations," *Oper. Res. Lett.*, vol. 16, no. 2, pp. 101–113, 1994.



Bahram Alidaee received the B.S. degree from the University of Tehran, Iran, the M.B.A. degree from the University of North Texas, Denton, and the Ph.D. degree in mathematical sciences from University of Texas at Arlington, in 1992.

He is currently an Associate Professor of operations management at the School of Business Administration and the Hearing Center for Enterprise Science at the University of Mississippi, University. His current research interests include: design and implementation of metaheuristics for the combinatorial optimization problems, and surrogate/Lagrangian relaxation applications to very-large-scale problems in optimization problems. He has published in various journals including *Management Science*, *Journal of Production and Operations Management*, *Journal of Operational Research Society*, *European Journal of Operational Research*, *Journal of Optimization Theory and Applications*, *Computers and Operations Research*, etc.



Weiqi Li received the B.S. degree in mechanical engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1982, the B.A. degree in international trade from University of International Business and Economics, Beijing, China, in 1986, and the M.B.A. and M.S. degrees in computer science and the Ph.D. degree in management information system from the University of Mississippi, University, in 1994, 1996, and 1998, respectively.

He served as Visiting Assistant Professor in the School of Business at University of Mississippi in 1999 and as Assistant Professor in the School of Business at Michigan Technological University, Houghton, from 1999 to 2001. In 2001 he joined the faculty of the School of Management of the University of Michigan, where he is currently an Assistant Professor of management information systems. His research interests include machine learning, heuristics for the combinatorial optimization problems, and e-learning.