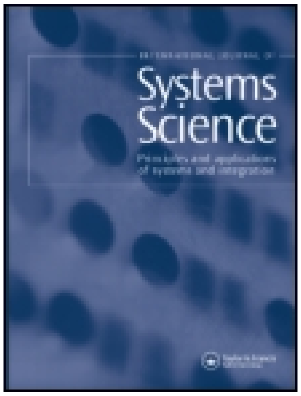

International Journal of Systems Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tsys20

# Population sizing for inductive linkage identification

Jih-Yiing Lin [a] & Ying-ping Chen [a]
[a] Department of Computer Science , National Chiao Tung University , Hsinchu , Taiwan

Published online: 13 May 2011.

*International Journal of Systems Science*
Vol. 44, No. 1, January 2013, 1–13

# Population sizing for inductive linkage identification

Jih-Yiing Lin and Ying-ping Chen*

*Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan*

(*Received 10 August 2010; final version received 9 March 2011*)

Variable interdependency, referred to as linkage in genetic algorithms (GAs), has been among the most useful information in evolutionary optimisation. With the aid of linkage information, efficient evolution can be attained by GAs. Among variants of advanced GAs, linkages are either explicitly identified, as in perturbation-based methods, or implicitly extracted, as in estimation of distribution algorithms (EDAs). As linkage discovery can be considered a matter of information extraction, Shannon's entropy, a renowned metric, has been widely adopted in modern GAs. Despite the validation of theoretical bounds, which is not algorithm-specific, on evaluation complexity of linkage problems, a representative population sizing model for discrete EDAs has been developed based on the distribution of entropy measurement. On the other hand, though entropy metrics have been adopted in recent perturbation-based methods, relevant complexity analysis on these methods is still absent. In this article, we propose a population sizing model for a recently developed linkage identification method, called *inductive linkage identification* (ILI). The proposed model takes the entropy-based classification algorithm into account and is capable of providing an accurate estimation of population requirement. The adopted modelling approach is different than that for discrete EDAs and may give researchers insights into entropy-based linkage discovery approaches.

**Keywords:** inductive linkage identification; perturbation-based methods; building blocks; population sizing; decision trees; genetic algorithms

## 1. Introduction

Genetic algorithms (GAs), as optimisation methods, are popular for their simplicity and applicability. Inspired by natural evolution, a GA commonly starts with initialising a population of chromosomes which represent candidate solutions to the optimisation problem. Then, evolutionary operations such as parent selection, recombination, mutation and survivor selection are applied to the population to generate offspring and to evolve the population towards better solutions with higher fitness. These operations are usually easy to implement. In addition to the implementation simplicity, GAs also require little or limited problem domain knowledge and exhibit a high level of global exploration ability. All these features enable GAs to be widely applied in many areas, including scheduling problems, telecommunication, robotics, engineering design, chemistry, finance and others.

One underlying mechanism that leads to their success is stated in the building block hypothesis (Goldberg 1989): GAs can implicitly decompose a problem into sub-problems via recombining promising partial solutions, which are referred to as building blocks. When a problem is decomposable into low correlated sub-problems, solving each sub-problem is more efficient than solving the whole problem. Acquiring the information of variable interdependency and thus separating the problem into proper sub-problems are hence crucial to the GA performance. To cope with this issue, early studies aimed on evolving the recombination operators and representation of solutions to capture the data interdependency, or linkage, during the optimisation (Goldberg, Korb, and Deb 1989; Kargupta 1996; Harik 1997). However, these evolutionary processes capture linkage in a much slower pace than the selection process and consequently leads to premature convergence. The succeeding popular algorithms can be roughly classified into two categories: estimation of distribution algorithms and perturbation based methods (Munetomo and Goldberg 1998).

These two categories of techniques solve the linkage problem in different ways. The estimation of distribution algorithms (EDAs) implicitly identify linkages by building probability models on promising solutions and generating new solutions accordingly, while the perturbation based methods explicitly identify linkage by perturbing variables and observing the induced fitness difference. Nevertheless, they both utilise entropy related metrics to select models or

*Corresponding author. Email: ypchen@nclab.tw

ISSN 0020–7721 print/ISSN 1464–5319 online

http://dx.doi.org/10.1080/00207721.2011.577246
http://www.tandfonline.com

identify linkages. As entropy related metrics assess the impurity of a collection, the population size of these techniques is required to be sufficient to guarantee the correctness of the probability model or the identified linkage. The population size has been a crucial parameter relevant to the performance of a GA. Along the research line of EDAs, Yu, Sastry, Goldberg, and Pelikan (2007) proposed a general population sizing model for entropy based discrete estimation of distribution algorithms. Their model was developed based on the distribution of entropy measurement and took into account the selection pressure and the variance of fitness. They proved that utilising EDAs to solve a problem with $m$ sets of interdependent variables requires a $\Theta(m \log m)$ population to guarantee $(1 - 1/m)$ model accuracy. Along the research line of perturbation based methods, Tsuji, Munetomo, and Akama (2006) proposed the *dependency detection for distribution derived from fitness differences* ($D^5$) and provided a recent population sizing model for perturbation based methods. Though $D^5$ adopts a clustering technique and entropy metric in linkage identification, its population model was developed based on the statistics of fitness difference in the population regardless of the influence of entropy. The population sizing model states that $D^5$ requires a $\mathcal{O}(2^k \log \ell)$ population to handle an order-$k$ additively separable function defined on strings of length $\ell$.

In this article, we propose a population sizing model for the *inductive linkage identification* (ILI) (Chuang and Chen 2007; Chen, Chuang, and Huang 2011). ILI is a perturbation-based linkage identification method which utilises ID3 decision trees to identify sets of dependent variables. It identifies dependent variables by constructing a decision tree based on the fitness differences caused by perturbing one selected variable. The variables used as nodes in the constructed decision tree are considered interdependent. As ILI is a concise method that consists of only the decision tree technique, the proposed population sizing model for ILI reveals a pure relationship between the number of training instances and the error rate of a decision tree. This pure relationship is further extended to provide a population sizing model for perturbation based methods. In contrast to the population sizing model of $D^5$, our model takes the entropy metric into account and is capable of accurately delineating the relationship between the linkage identification error rate and the population size in a broad range of problems. In contrast to the general population sizing model of EDAs, a complex analysis of implicit linkage discovery and selection pressure with the presumption of a building block fitness distribution, our model considers only the linkage discovery and provides an alternative population sizing approach to entropy based linkage discovery methods. In our point of view, analysing the solely explicit linkage discovery behaviour may provide some insight into the implicit linkage discovery algorithm which often mixes several optimisation factors.

Our model can provide not only a concrete guide to the population size setting of ILI but also, for its clear delineation of a pure relationship between the number of training instances and the error rate of a decision tree, some insights into the population sizing of other entropy based mechanisms, varying from EDAs to data mining approaches. With guidance of our population sizing model, applying linkage discovery algorithms to the problems which require finding relevant decision variables to the outputs (Saridakis and Dentsoras 2009; Wang 2009; Kim, de Silva, and Park 2010) may further improve the performance obtained with traditional GAs. Furthermore, the proposed model may also provide an analysis approach to the learning curve which is potentially useful to various data mining applications.

The rest of this article is organised as follows. Section 2 gives the background of this study. Section 3 briefly introduces ILI. It first gives the definition of linkage and explains how perturbation methods is related to linkage discovery, then reviews the ID3 decision tree learning algorithm, and finally describes the algorithm of ILI. The proposed population sizing model for ILI is delineated, verified, and discussed in Sections 4 and 5. In Section 4, the error probability of the ID3 decision tree algorithm is analysed and applied to derive a formula that depicts the relationship among the population size, sub-problem size, number of subproblem and linkage identification correct probability, followed by empirical verification and discussion in Section 5. Finally, Section 6 concludes this article.

## 2. Background

Recent methods that deal with the linkage problem can be classified into two categories: The EDAs which implicitly discover linkage and the perturbation based algorithms which explicitly identify linkage. The EDAs construct probability models on promising solutions and generate new solutions accordingly. Though early EDAs assume no interactions between variables (Baluja 1994; Harik, Lobo, and Goldberg 1999), subsequent studies model pairwise interactions as well as multivariate interactions to provide better performance (Baluja and Davies 1997; de Bonet, Isbell, and Viola 1997; Harik 1999; Mühlenbein and Mahnig 1999; Pelikan and Mühlenbein 1999; Pelikan, Goldberg, and Cantú-Paz 1999; Mühlenbein and Höns 2005; Gámez, Mateo, and Puerta 2007; Jiang, Wang, and Yang 2009;

Santana, Larrañaga, and Lozano 2010). The modelling of multivariate interactions captures the distributions of sets of dependent variables and involves selection among numerous possible probability models to attain appropriate ones. Various metrics have been adopted in model construction. The most common metrics are Bayesian metrics (Cooper and Herskovits 1992; Heckerman, Geiger, and Chickering 1995) and minimum description length (MDL) metrics (Rissanen 1978, 1989, 1996). The Bayesian metrics approximate the likelihood of a probability model given the data, and the MDL metrics measure the code length of the model and the code length of the modelled data. Though distinguished, these two types of metrics are strongly connected. In the view of some researchers, the code length of the model and the code length of modelled data in MDL do correspond to the prior probability and marginal likelihood, respectively, in the Bayesian framework (MacKay 2003). As the MDL principle is a formalisation of Occam's Razor, either adopting Bayesian metrics or MDL metrics, EDAs embrace the fundamental concept of information theory.

The term *perturbation* has been widely adopted in GAs. Among relevant interesting studies is Solteiro Pires, Tenreiro Machado, and de Moura Oliveira (2006)'s work investigating the effect of perturbing mutation probability on GA dynamics from the viewpoint of signal propagation. The perturbation methods referred to here explicitly identify variable interdependencies via examining the fitness differences caused by variable perturbations. These methods basically assume that non-linearity (Munetomo and Goldberg 1998) or non-monotonicity (Munetomo and Goldberg 1999) exists within interdependent variables. Following this concept, recent studies (Tsuji et al. 2006; Chuang and Chen 2007; Ting, Zeng, and Lin 2010) have introduced data mining mechanisms into fitness difference analysis to efficiently identify variable interdependencies to replace the simple linearity or monotonicity check. The *dependency detection for distribution derived from fitness differences* ($D^5$) clusters individuals into subpopulations according to fitness differences and identifies interdependent variables by finding the set of variables that can achieve the lowest entropy (Tsuji et al. 2006). The *inductive linkage identification* (ILI) constructs ID3 decision trees according to the fitness differences to identify interdependencies (Chuang and Chen 2007). The latest perturbation based method adopted the Apriori algorithm, a well-known mining technique, to identify linkage (Ting et al. 2010). These studies, inspired by data mining methods and adopting concepts of information theory, provide efficient ways to identify interdependencies given the entire population's fitness differences.

While the trend of adopting concepts of information theory in these algorithms is evident, some generic theoretical analysis on the evaluation time of decomposable problems have been proposed to delineate the performance limitation inherent in these problems and formed a basis for performance comparison. In Streeter's study, an efficient algorithm was proposed to illustrate that the upper bound of optimising an order-$k$ additively separable function defined on strings of length $\ell$ is $\mathcal{O}(2^k \ell \ln \ell)$ function evaluations (Streeter 2004). Choi, Jung, and Moon (2009) further provided the theoretical lower and upper bounds for data interdependency discovery. Given a problem of $n$ variables with $m = \mathcal{O}(n^{k-\delta})$ sets of dependent variables with each set limited to $k$ variables, their deduction validated that discovering all the interdependencies requires $\Omega(m \log n / \log m)$ function evaluations. They also proved that $O(n^2 \log n)$ is enough to bound the evaluation time of bounded problems with $\mathcal{O}(n)$ interdependencies. As these rigorous analysis on decomposable problems are generic and do not depend on any specific algorithms, they can well formulate the generalised bounds and provide a baseline for performance comparison.

However developing an efficient algorithm capable of reaching the lower bound of evaluation time requires delving into the complexity analysis of manifolds of algorithms. In evolutionary computation, complexity analysis usually comprises estimations of both function evaluations and population sizes. Though both of them are fundamental performance indexes, the analysis on population sizes is more critical than that on function evaluations because population sizing is considered critical to the success and efficiency of GAs. For the aforementioned two categories of algorithms which adopt the information theory concept, the size of a population is required to be sufficient not only to satisfy the need of initial building block supply but also to guarantee small error rate for linkage discovery.

In the line of performance analysis on EDAs Pelikan, Sastry, and Goldberg (2002) estimated that the number of evaluations of the *Bayesian optimisation algorithm* (BOA) until reliable convergence to optimum grows as $\mathcal{O}(n^{1.55})$ or $\mathcal{O}(n^2)$, where $n$ is the number of variables in the problem, depending on the scaling of the sub-problem in a proper problem decomposition. They also assured that the population size and the selection pressure have non-negligible impact on the BOA performance. A population size between $\mathcal{O}(n^{1.05})$ and $\mathcal{O}(n^{2.1})$ is required by BOA to build an accurate model. These bounds are consistent with the empirical results of other EDAs where the function evaluation and the population size roughly scale as $\Theta(n^{1.4})$ (Sastry and Goldberg 2004). As both metrics for EDAs, MDL metrics and Bayesian metrics, can be considered as entropy-related metrics, Yu et al. (2007) further derived a general population sizing model for entropy-based discrete estimation of distribution algorithms. Based on the distribution of entropy measurement and taking into account the selection pressure and the variance of the fitness of sup-problems, the population size must be $\Theta(m \log m)$ to guarantee $(1 - 1/m)$ model accuracy, where $m$ is the number of sets of interdependent variables and is proportional to the problem size. In the population sizing of perturbation based methods Tsuji et al. (2006) estimated the population size for $D^5$ solely based on the statistics of the sub-population clustered according to fitness differences. They provided a bound of $\mathcal{O}(2^k \log \ell)$ for $D^5$ to handle an order-$k$ additively separable function defined on strings of length $\ell$.

The general population sizing model proposed by Yu et al. (2007) illustrates a complex relationship among the selection pressure, probability model error rate, and population size with the presumption of some certain building block fitness information is given. On the other hand, though adopting clustering technique and the entropy metric, the population sizing model of $D^5$ proposed by Tsuji et al. (2006) was developed regardless of these matters. As entropy metric or the like have been a trend in linkage discovery methods, a clear relationship between the population size and the linkage error rate of entropy based methods is in order. In this article, we propose a population sizing model for inductive linkage identification (ILI). As a perturbation-based linkage identification algorithm, ILI is a concise, newly developed algorithm which utilises ID3 decision trees in the process of linkage identification. The proposed analysis approach firstly depicts the essential relationship between the number of training instance and the accuracy of the corresponding decision tree and then extends the relationship to provide a population sizing model for ILI. The proposed population sizing model can accurately approximate the sufficient population size for ILI given an error rate requirement. As it is also a simple approach which can clearly delineate the pure relationship between the population size and the error rate in entropy-based methods, the model may provide some insights into the population sizing of other entropy-based mechanisms, varying from EDAs to data mining approaches.

## 3. Inductive linkage identification

In this section, we first give the definition of linkage and describe how perturbation methods detect linkage. Then, we briefly review the ID3 decision tree learning algorithm and relate it to the linkage identification of ILI. Finally, we describe ILI in detail.

### 3.1. *Linkage definition and perturbation method*

For convenience, in this article, we adopt additively decomposable functions (ADF) as the problem model as Chuang and Chen (2007) did in their study. Let $s = s_1 s_2 \cdots s_\ell$, for $\ell$ variables, represent a string $s$ of length $\ell$. The fitness of string $s$ is defined as

$$f(\mathbf{s}) = \sum_{i=1}^{m} f_i(\mathbf{s}_{v_i}),$$

where $m$ is the number of subfunctions $f_i$, and $s_{v_i}$ is the subset variables of $s$ that corresponds to $f_i$. Each subfunction $f_i$ is a nonlinear function and $v_i$ here is a vector of indexes that specifies the corresponding subset variables $s_{v_i}$. For example, if $v_i = (1, 3, 5, 8)$, $\mathbf{s}_{v_i} = s_1 s_3 s_5 s_8$. Let $V_i$ be the set that contains all the elements of $v_i$, and we refer to $V_i$ as a linkage set. In ILI, binary variables are assumed, and subfunctions that do not share decision variables, i.e. *non-overlapping* subfunctions, are considered. That is, $V_i \cap V_j = \emptyset$ if $i \neq j$.

Without lose of generality, we assume that $V_i = \{1, 2, \ldots, k\}$ and a perturbation is applied to $s_1$. The corresponding fitness difference $df_1$ is expressed as

$$\begin{aligned} df_1(\mathbf{s}) &= f(s_1 s_2 \cdots s_\ell) - f(\overline{s_1} s_2 \cdots s_\ell) \\ &= \left[ f_1(s_1 s_2 \cdots s_k) + \sum_{i=2}^{m} f_i(\mathbf{s}_{v_i}) \right] \\ &\quad - \left[ f_1(\overline{s_1} s_2 \cdots s_k) + \sum_{i=2}^{m} f_i(\mathbf{s}_{v_i}) \right] \\ &= f_1(s_1 s_2 \cdots s_k) - f_1(\overline{s_1} s_2 \cdots s_k). \end{aligned} \tag{1}$$

As each subfunction is nonlinear, we can find from the derivation that the fitness difference obtained from perturbing a variable $s_j$ is a function which solely depends on all the variables in $V_i$ containing $s_j$. In other words, by means of perturbation, one can turn the fitness function dependent on all variables into a fitness difference function dependent on only the corresponding linkage set.

### 3.2. *ID3 decision tree*

Decision tree learning is a widely adopted method in data mining. Given a set of training instances with their attribute values and target values, a decision tree can be constructed to predict the target value of an instance given its attribute values. A constructed decision tree has its nodes corresponding to attribute variables and its leaves corresponding to target values. Each node has as many descendants as the number of the possible values of the attribute variable it represents. Given a new instance's attribute values, one can trace the corresponding attribute values in the tree to predict the target value.

For the purpose of analysis, we introduce the ID3 decision tree learning algorithm (Quinlan 1993) adopted by ILI. Given a set of training instance, categorising them according to different attribute variables can result in different target value distributions. As the goal of the decision tree is to construct a concise tree that can predict well the target value of a new instance, a statistical property, *information gain*, is adopted to assess how well the training instances are categorised by selecting an attribute variable as a tree node. After an attribute variable is selected as a node, the training instances are categorised into groups according to its value. Each group of training instances further undergoes the attribute variable selection to categorise the training instances into subgroups. This process iterates until all training instances in their categorised group have the same target value or have all the attribute values assigned.

*Information gain* assesses the impurity of instances reduced by selecting an attribute variable to categorise them. In information theory, the impurity of an arbitrary collection of instances is defined as *entropy*. Given a collection $D$, containing instances of $c$ different target values, the entropy of $D$ relative to this $c$-wise classification is defined as

$$\mathrm{Entropy}(D) \equiv \sum_{i=1}^{c} -p_i \log_2 p_i \, , \tag{2}$$

where $p_i$ is the proportion of $D$ belonging to class $i$. In all the calculations related to entropy, we define $0 \log_2 0$ to be 0. If an attribute $A$ is selected to categorise the collection $D$, the resulting entropy of the categorised collection $D$ is

$$\mathrm{Entropy}(D|A) \equiv \sum_{v \in Val(A)} \mathrm{Prob}(Val(A) = v)\mathrm{Entropy}(D_v), \tag{3}$$

where $Val(A)$ is the set of all possible values for attribute $A$, $\mathrm{Prob}(Val(A) = v)$ is the probability of an instance with its attribute $A$ of value $v$, and $D_v$ is the subset of $D$ in which attribute $A$ has value $v$. As $\mathrm{Prob}(Val(A) = v)$ can be approximated by the proportion of $D_v$ to $D$, we can further have

$$\mathrm{Entropy}(D|A) \equiv \sum_{v \in Val(A)} \frac{|D_v|}{|D|} \mathrm{Entropy}(D_v). \tag{4}$$

Thus, in terms of entropy, the information gain of selecting attribute $A$ to categorise the collection $D$ can be defined as the difference between Entropy($D$) and Entropy($D|A$). Then the information gain, Gain($D, A$), of an attribute $A$ relative to a collection of instances $D$ is

$$\mathrm{Gain}(D, A) \equiv \mathrm{Entropy}(D) - \sum_{v \in Val(A)} \frac{|D_v|}{|D|} \mathrm{Entropy}(D_v). \tag{5}$$

All the attribute variables map to the tree nodes are obviously the variables responsible for the different values of the target variable. In this perspective, a constructed tree illustrates a function which maps relevant attribute variables to the target variable. Thus, treating the solution string as the list of attribute values and the fitness difference caused by perturbation as the target values, ILI can identify linkages by applying the ID3 decision tree learning algorithm.

### 3.3. *ILI algorithm*

Integrating the ideas illustrated in previous sections, ILI performs a perturbation operation to the whole population and constructs an ID3 decision tree according to the fitness difference caused by the perturbation. The resultant decision tree then consists of a linkage set that corresponds to the perturbation. This procedure repeats until all variables are divided into their linkage sets. The pseudo code of the overall ILI procedure is depicted in Algorithm 1. The first step of ILI is to initialise a population of strings. Then, ILI identifies one linkage set at a time using the following procedure: (1) a variable is randomly selected to be perturbed; (2) an ID3 decision tree with the perturbed variable as root is constructed according to the fitness differences caused by perturbations; (3) by inspecting the constructed tree, the variables used in the decision tree are collected and considered as a linkage set.

From Algorithm 1, we can find that the evaluation time of ILI is of $\mathcal{O}(mn)$, where $m$ is the number of linkage sets, and $n$ is the population size. As the population size must be sufficiently large to guarantee small linkage identification error rate, an accurate population sizing model is necessary for both complexity analysis and an appropriate population size setting.

## 4. Population sizing for inductive linkage identification

Since ILI adopts the ID3 decision tree to identify linkage, we first investigate the relationship between the population size, $n$ and the probability of selecting a wrong decision variable in the decision tree. As aforementioned, we assume that the objective function consists of non-overlapping subfunctions and the sampling of objective function is noise free. Consider the following additively decomposable function:

$$f(\mathbf{s}) = f_{\text{trap}_k}(s_1 \cdots s_k) + \sum_{i=2}^{m} f_i(s_{v_i}).$$

**Algorithm 1:** Inductive Linkage Identification

```
procedure IdentifyLinkage($f$, $l$)
  Initialize a population $P$ with $n$ string of length $\ell$.
  Evaluate the fitness of strings in $P$ using $f$.
  $V \leftarrow \{1, \ldots, \ell\}$
  $m \leftarrow 0$
  while $V \neq \emptyset$ do
    $m \leftarrow m + 1$
    Select $v$ in $V$ at random.
    $V_m \leftarrow \{v\}$
    $V \leftarrow V - \{v\}$
    for each string $\mathbf{s^i} = s^i_1 s^i_2 \cdots s^i_\ell$ in $P$ do
      Perturb $s^i_v$.
      $df^i \leftarrow$ fitness difference caused by perturbation.
    end for
    Construct an ID3 decision tree using $(P, df)$ with $v$ as root node.
    for each decision variable $s_j$ in the tree do
      $V_m \leftarrow V_m \cup \{j\}$
      $V \leftarrow V - \{j\}$
    end for
  end while
  return the linkage sets: $V_1, V_2, \ldots, V_m$
end procedure
```

The trap$_k$ function is a $k$-bit trap function, a function of unitation which can be expressed as

$$f_{\text{trap}_k}(s_1 s_2 \cdots s_k) = \begin{cases} k, & \text{if } u = k \\ k - 1 - u, & \text{otherwise} \end{cases},$$

where $u$ is the number of ones in the binary string $s_1 s_2 \cdots s_k$. To identify the linkage set, $V = \{1, 2, \ldots, k\}$, $s_1$ is perturbed. In the case of alternating a bit from one to zero, one is added to the fitness of each individual except those who originally have $f_{\text{trap}_k}(s_1 \cdots s_k) = k$. These exceptions have a new value of zero, and hence the fitness difference $-k$. In the case of alternating a bit from zero to one, this reduces one from the fitness of each individual except those who originally have $f_{\text{trap}_k}(s_1 \cdots s_k) = 0$. These exceptions have a new value of $k$, and hence fitness difference $k$. Thus, after perturbation, only those $s_2 s_3 \cdots s_k = 11 \cdots 1$ individuals have fitness difference $\pm k$. Figure 1(a) illustrates the fitness differences of individuals categorised by $s_1 s_2 s_3 s_{3+}$ values. For example, the first row indicates the fitness difference of any individual with its $s_1 s_2 s_3 s_{3+} = \overline{0}000$ is 1. The $f_o$ column denotes the original value of $f_{\text{trap}_3}(s_1 s_2 s_3)$, and the $f_n$ column denotes the new value of $f_{\text{trap}_3}(s_1 s_2 s_3)$ after perturbation. The $\overline{0}$ denotes that the corresponding variable has been perturbed from 1 to 0, and $s_{3+}$ denotes a specified variable other than $s_1$, $s_2$ and $s_3$. From this table, we can see that only those $s_2 s_3 = 11$ individuals have fitness difference $\pm 3$.

Since every variable of each individual is generated at random, the subpopulation size of individuals with a same substring $s_1 s_2 \cdots s_k = a_1 a_2 \cdots a_k$, denoted as $n_{s^k}$, is a binomial distribution with $n =$ population size and $p = 2^{-k}$. When $n$ is sufficiently large, an excellent approximation of such a binomial distribution can be obtained via the normal distribution:

$$N(np, np(1 - p)).$$

Let $\overline{s}_i$ denote that $s_i$ is perturbed, $b$-$df$ denote a fitness difference of $b$, $P_b$ denote the population of the individuals with $b$-$df$, and $P_{(s_{i_1} s_{i_2} \cdots s_{i_j} = a_1 a_2 \cdots a_j)}$ denote the population of the individuals with their $s_{i_1} = a_1, s_{i_2} = a_2, \ldots, s_{i_j} = a_j$. Accordingly, the size of $P_{(\overline{s}_1 s_2 \cdots s_k = 11 \cdots 1)}$ and the size of $P_{(\overline{s}_1 s_2 \cdots s_k = 01 \cdots 1)}$ behave as the normal distribution described above. Then both $P_k$ and $P_{-k}$ have $n_{s^k}$ individuals while both $P_1$ and $P_{-1}$ have $(2^{k-1} - 1) n_{s^k}$ individuals.

Since ILI introduces the perturbed bit as the root of decision tree, the decision tree then divides the population into two portions, $P_{(\overline{s}_1 = 0)}$ and $P_{(\overline{s}_1 = 1)}$. The fitness difference distributions of these two subpopulations are similar. For simplicity, we omit the analysis of $P_{(\overline{s}_1 = 1)}$. As $P_{(\overline{s}_1 = 0)}$ consists of only two distinct subpopulations, $P_{-k}$ and $P_1$, the portion of $P_{-k}$ is critical to the correctness of linkage identification. And with this least portion of unique fitness difference, a size of $n_{s^k}$, the trap function has the largest error probability in perturbation methods. Note also that the *nith* function behaves the same as the trap function after perturbation, and thus has the same error probability as the trap function in perturbation methods.

In $P_{(\overline{s}_1 = 0)}$, selecting any variable in the linkage set $V$ as a decision node would separate evenly the population into two subpopulations, $P_{(\overline{s}_1 s_{i \in V} = 00)}$ and $P_{(\overline{s}_1 s_{i \in V} = 01)}$. $P_{(\overline{s}_1 s_{i \in V} = 00)}$ consists of only individuals with 1-$df$ while $P_{(\overline{s}_1 s_{i \in V} = 01)}$ consists of 1-$df$ individuals and the whole $P_{-k}$. One the other hand, selecting a variable $s_{k+}$ outside the linkage set as a decision node would separate the population into two equivalent subpopulation, each with roughly half of $P_{-k}$. Let $T_{s_{i \in V}}$ and $T_{s_{k+}}$ denote these two kinds of tree correspondingly. The fitness difference distribution symmetry inherent in $T_{s_{k+}}$ trees grounds a

(a)

| $s_1$ | $s_2$ | $s_3$ | $s_{3+}$ | $f_o$ | $f_n$ | $df_1$ |
|---|---|---|---|---|---|---|
| $\bar{0}$ | 0 | 0 | 0 | 1 | 2 | 1 |
| $\bar{0}$ | 0 | 0 | 1 | 1 | 2 | 1 |
| $\bar{0}$ | 0 | 1 | 0 | 0 | 1 | 1 |
| $\bar{0}$ | 0 | 1 | 1 | 0 | 1 | 1 |
| $\bar{0}$ | 1 | 0 | 0 | 0 | 1 | 1 |
| $\bar{0}$ | 1 | 0 | 1 | 0 | 1 | 1 |
| $\bar{0}$ | 1 | 1 | 0 | 3 | 0 | -3 |
| $\bar{0}$ | 1 | 1 | 1 | 3 | 0 | -3 |
| $\bar{1}$ | 0 | 0 | 0 | 2 | 1 | -1 |
| $\bar{1}$ | 0 | 0 | 1 | 2 | 1 | -1 |
| $\bar{1}$ | 0 | 1 | 0 | 1 | 0 | -1 |
| $\bar{1}$ | 0 | 1 | 1 | 1 | 0 | -1 |
| $\bar{1}$ | 1 | 0 | 0 | 1 | 0 | -1 |
| $\bar{1}$ | 1 | 0 | 1 | 1 | 0 | -1 |
| $\bar{1}$ | 1 | 1 | 0 | 0 | 3 | 3 |
| $\bar{1}$ | 1 | 1 | 1 | 0 | 3 | 3 |

(b)

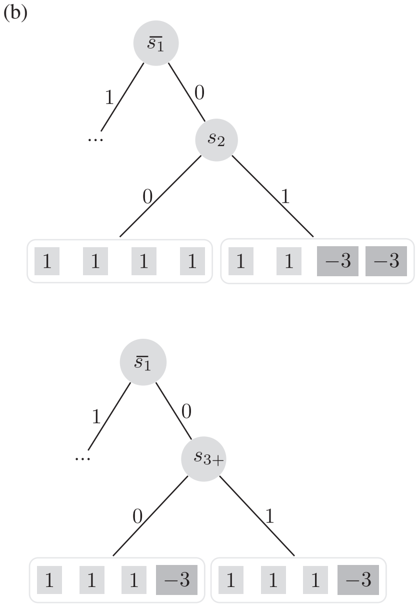

Figure 1. Fitness differences and decision tree construction for scenario I.

higher entropy state than that of the asymmetric $T_{s_{i\in V}}$ trees. According to the expected fitness difference distribution and the information gain defined in Equation (5), taking $s_{i\in V}$ as a tree node can achieve a lower entropy state than taking $s_{k+}$ can. Thus, selecting wrong variables during the decision tree construction does not often occur. Figure 1(b) illustrates a $k=3$ example of $T_{s_{i\in V}}$, on the above, and $T_{s_{3+}}$, on the bottom.

Of course, there are still chances for a decision tree to take a wrong variable as an internal node. Since $P_{-k}$ can be decomposed into $P_{(\bar{s}_1 s_2 \cdots s_k s_{k+}=01\cdots10)}$ and $P_{(\bar{s}_1 s_2 \cdots s_k s_{k+}=01\cdots11)}$, once either one of the subpopulation is absent, the $s_{k+}$ may be taken as a decision node. In the scenario where $P_{(\bar{s}_1 s_2 \cdots s_k s_{k+}=01\cdots10)}$ is absent, $T_{s_{i\in V}}$ and $T_{s_{k+}}$ would both have a subpopulation consisting of only 1-*df* individuals and the other subpopulation consisting of both 1-*df* and $-k$-*df* individuals. Accordingly, $P_{(\bar{s}_1 s_{i\in V}=01)}$ in $T_{s_{i\in V}}$ and $P_{(\bar{s}_1 s_{k+}=01)}$ in $T_{s_{i_{k+}}}$ are the subpopulations that contain $-k$-*df* individuals. As the $-k$-*df* individuals are exactly the same in these two trees, once the size of $P_{(\bar{s}_1 s_{i\in V}=01)}$ is larger than that of $P_{(\bar{s}_1 s_{k+}=01)}$, $T_{s_{k+}}$ can achieve a lower entropy state than $T_{s_{i\in V}}$ and thus introduce a wrong variable into the linkage set. Figure 2 illustrates a $k=3$ example of this scenario.

In the scenario when $P_{(\bar{s}_1 s_2 \cdots s_k s_{k+}=01\cdots10)}$ is absent, since there are around half of the population in $P_{(\bar{s}_1=0)}$, the size of $P_{(\bar{s}_1 s_{k+}=01)}$, $n_{s_{k+}}$, can be approximated with a normal distribution as

$$N\left(\frac{2^{k-1}}{2(2^k-1)}n, \frac{2^{k-1}(2^{k-1}-1)}{2(2^k-1)^2}n\right).$$

The size of $P_{(\bar{s}_1 s_{i\in V}=01)}$, $n_{s_{i\in V}}$, can also be approximated with a normal distribution as

$$N\left(\frac{(2^{k-1}-1)}{2(2^k-1)}n, \frac{2^{k-1}(2^{k-1}-1)}{2(2^k-1)^2}n\right). \tag{6}$$

when $n_{s_{k+}}$ is less than the largest among $n_{s_{i\in V}}$, it is possible for $s_{k+}$ to be selected as a decision tree node. Moreover, when $n_{s_{k+}}$ is less than the smallest among $n_{s_{i\in V}}$, $s_{k+}$ would definitely be taken as a decision variable. Since the distribution of each $n_{s_{i\in V}}$ is identical, the largest among $n_{s_{i\in V}}$ can be considered as the largest number sampled from the normal distribution described by Equation (6). In other words, it is a $(k-1)$-th order statistic. The smallest among $n_{s_{i\in V}}$ is a first order statistic. Therefore, $p_{\text{terr}}$, the error probability of the $P_{(\bar{s}_1=0)}$, when $P_{(\bar{s}_1 s_2 \cdots s_k s_{k+}=01\cdots10)}$ is absent, can be estimated as

$$\Phi\left(\frac{\mu_{X_{(1:k-1)}} - \mu_{s_{k+}}}{\sigma_{s_{k+}}}\right) \le p_{\text{terr}} \le \Phi\left(\frac{\mu_{X_{(k-1:k-1)}} - \mu_{s_{k+}}}{\sigma_{s_{k+}}}\right), \tag{7}$$

where $\Phi$ denotes the cumulative distribution function of the standard normal function, $\mu_{X_{(1:k-1)}}$ denotes the mean of the first order statistic in a sample of size $k-1$ and $\mu_{X_{(k-1:k-1)}}$ denotes the mean of the $(k-1)$-th order statistic. The mean and the standard deviation of $n_{s_{k+}}$ are denoted as $\mu_{s_{k+}}$ and $\sigma_{s_{k+}}$ correspondingly. In the following sections, for verification purpose, we will use the terms

$$\Phi\left(\frac{\mu_{X_{(1:k-1)}} - \mu_{s_{k+}}}{\sigma_{s_{k+}}}\right) \tag{8}$$

(a)

| $s_1$ | $s_2$ | $s_3$ | $s_{3+}$ | $df_1$ |
|---|---|---|---|---|
| $\bar{0}$ | 0 | 0 | 0 | 1 |
| $\bar{0}$ | 0 | 0 | 1 | 1 |
| $\bar{0}$ | 0 | 1 | 0 | 1 |
| $\bar{0}$ | 0 | 1 | 1 | 1 |
| $\bar{0}$ | 1 | 0 | 0 | 1 |
| $\bar{0}$ | 1 | 0 | 1 | 1 |
| ~~$\bar{0}$~~ | ~~1~~ | ~~1~~ | ~~0~~ | ~~–3~~ |
| $\bar{0}$ | 1 | 1 | 1 | –3 |
| $\bar{1}$ | 0 | 0 | 0 | –1 |
| $\bar{1}$ | 0 | 0 | 1 | –1 |
| $\bar{1}$ | 0 | 1 | 0 | –1 |
| $\bar{1}$ | 0 | 1 | 1 | –1 |
| $\bar{1}$ | 1 | 0 | 0 | –1 |
| $\bar{1}$ | 1 | 0 | 1 | –1 |
| $\bar{1}$ | 1 | 1 | 0 | 3 |
| $\bar{1}$ | 1 | 1 | 1 | 3 |

(b)

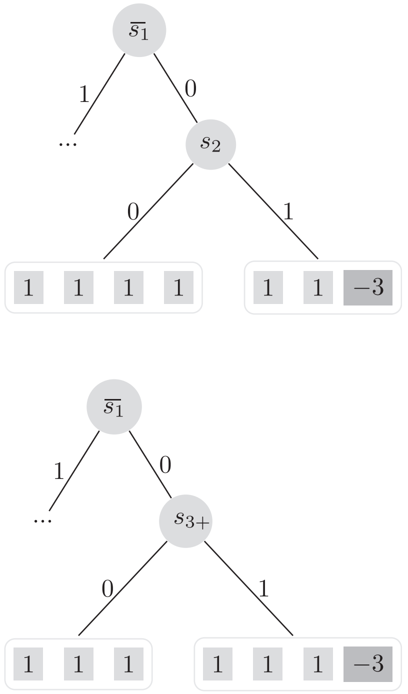

Figure 2. Fitness differences and decision tree construction for scenario II.

and

$$\Phi\left(\frac{\mu_{X_{(k-1:k-1)}} - \mu_{s_{k+}}}{\sigma_{s_{k+}}}\right), \tag{9}$$

as the lower bound and the upper bound of $p_{\text{terr}}$ to compute the lower bound and the upper bound of the population size.

Since the two absent scenarios are symmetric, their $p_{\text{terr}}$ are identical. For each scenario to occur, all the individuals in $P_{(\bar{s}_1=0)}$ should not contain the absent substring. Because $P_{(\bar{s}_1=0)}$ is about half of the population and the probability for an individual to contain the absent substring is $2^{-k}$, we can estimate the probability of one of the scenarios to occur as $(1-2^{-k})^{n/2}$. Thus, the total probability for selecting a wrong decision variable $s_{k+}$ in $P_{(\bar{s}_1=0)}$ is

$$2 \cdot (1-2^{-k})^{n/2} \cdot p_{\text{terr}}.$$

The probability for $P_{(\bar{s}_1=0)}$ to identify linkage correctly is

$$1 - 2 \cdot (1-2^{-k})^{n/2} \cdot p_{\text{terr}}.$$

Since there are two subpopulations from the root, $(\ell-k)s_{k+}$ candidates, the probability for ILI to correctly identify a linkage set of $k$ variables among $\ell$ variables is

$$[1 - 2 \cdot (1-2^{-k})^{n/2} \cdot p_{\text{terr}}]^{2(\ell-k)}.$$

The probability, $p_\alpha$, for ILI to correctly identify $m$ linkage sets among $\ell$ variables is then larger than

$$\prod_{i=1}^{m}[1 - 2 \cdot (1-2^{-k_i})^{n/2} \cdot p_{\text{terr}}(k_i)]^{2(\ell-k_i)}. \tag{10}$$

Given the value of $p_\alpha$, one can obtain an approximated upper bound of the required population size via solving Equation (10).

## 5. Empirical verification and discussion

In this section, we empirically verify our population sizing model and briefly discuss the population size requirements of EDAs and perturbation based methods as well as their relationship with the performance of optimisation.

### 5.1. *Empirical verification*

In the empirical verification, we adopted the following fitness function

$$f(\mathbf{s}) = \sum_{i=1}^{m} f_{\text{trap}_k}(s_{k\cdot(i-1)+1} \cdots s_{k\cdot(i-1)+k}).$$

When the fitness function is of this form, the probability for ILI to correctly identify all the linkage sets is

$$p_\alpha = [1 - 2 \cdot (1-2^{-k})^{n/2} \cdot p_{\text{terr}}]^{2m(\ell-k)}. \tag{11}$$

In order to conduct a thorough verification, we empirically determine the population sizes required for $k$ from 3 to 6 with various numbers of subproblems. The overall problem sizes, $\ell$, are $60, 120, 180, \ldots, 600$ bits. And $m$, the number of subproblems, is calculated by $m = \ell/k$. For each problem instance, we apply a bisection method to find the minimum population size required for ILI to correctly identify all the linkage sets. The judgement criterion applied is 30 consecutive and independent successful runs. Accordingly, the 95% confidence interval of $p_\alpha$ is 0.884 to 1.0 (Zwillinger and Kokoska 2000). Thus, we select $p_\alpha = 0.942$, the middle point of the interval, for the proposed model described by Equation (11) to estimate the population sizes. Table 1 lists the means of normal order statistics for different $k$'s (Arnold, Balakrishnan, and Nagaraja 1993). The mean of the $x$-th order statistic in a sample of size $n$ from the normal distribution is denoted as $\mu_{x:n}$.

Hence, the mean values of the $i$-th order statistics in a sample of size $k-1$ from the distribution of $n_{s_{i\in V_i}}$ can be calculated as $\mu_{X_{i:k-1}} = \mu_{s_{i\in V_i}} + \mu_{i:k-1}\sigma_{s_{i\in V_i}}$, where $\mu_{s_{i\in V_i}}$ and $\sigma_{s_{i\in V_i}}$ are the mean and standard deviation of Equation (6), respectively. Applying the aforementioned setting of $p_\alpha$ to Equation (11), the estimated population sizes and the corresponding empirical results are illustrated in Figure 3. The circle marks represent the empirically obtained population sizes,

Table 1. Mean values of the normal order statistics for different $k$.

| $\mu_{2:2}$ | $\mu_{3:3}$ | $\mu_{4:4}$ | $\mu_{5:5}$ | $\mu_{1:2}$ | $\mu_{1:3}$ | $\mu_{1:4}$ | $\mu_{1:5}$ |
|---|---|---|---|---|---|---|---|
| 0.564 | 0.846 | 1.029 | 1.163 | −0.564 | −0.846 | −1.029 | −1.163 |

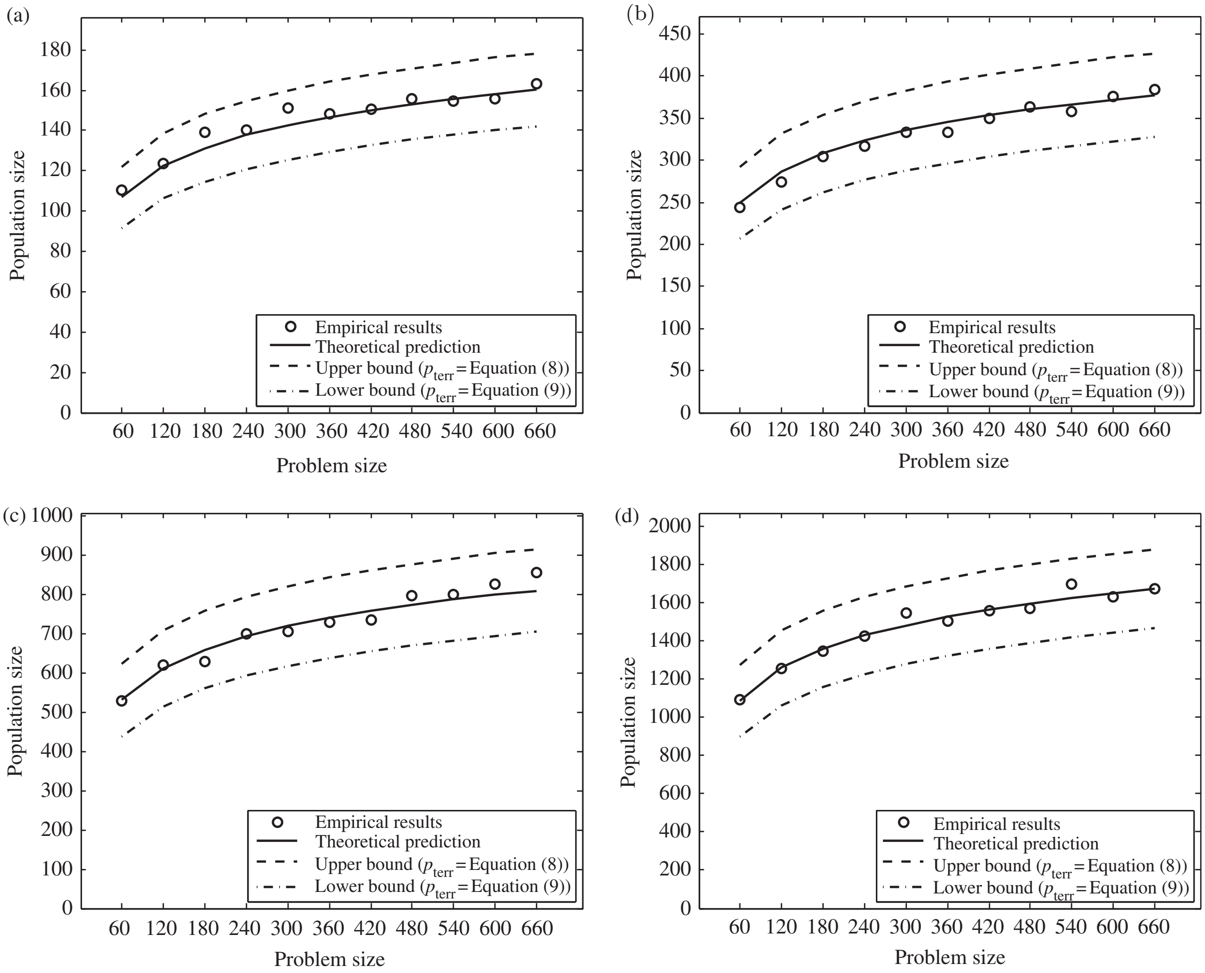

Figure 3. Population sizing model: theoretical prediction versus empirical results (a) $k = 3$, (b) $k = 4$, (c) $k = 5$ and (d) $k = 6$.

and the solid lines are predicted population sizes according to the proposed population sizing model Equation (11). Figure 4 further illustrates the population requirement of ILI for problems of different subproblem complexity. In Figure 4, the marks represent the obtained population sizes for problems of lengths 120, 360 and 600 bits, and the lines are predicted population sizes. Figures 3 and 4 indicate that the proposed population sizing model is able to provide a very good approximation for the population sizes required by ILI to identify the linkage sets in trap functions for different overall problem sizes as well as subproblem sizes.

Figure 5 shows the population sizes required by ILI on problems composed of different subproblem types ($k = 4$) and the computed theoretical bounds. The corresponding fitness difference distributions of different subproblem types are listed in Table 2. As $\text{trap}_4$ and $\text{nith}_4$ have identical fitness difference distribution, their population sizes present consistence in Figure 5. The test functions $\text{valley}_4$ and $\text{tmmp}_4$ also illustrate that test functions with more distinct fitness values requires smaller population size. All these empirical results support our statement in Section 4: the population sizing model based on the trap function provides an approximated population size upper bound for all test functions.

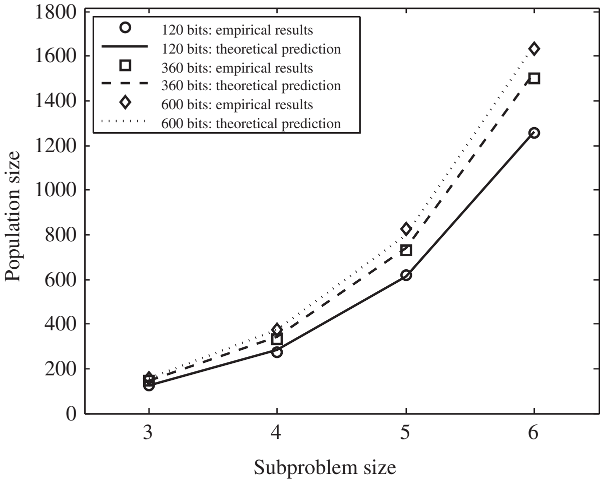

Figure 4. Population sizing model: theoretical prediction versus empirical results for different $k$'s.

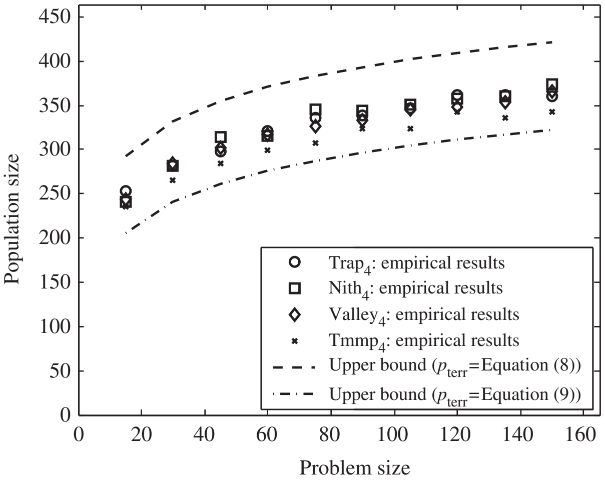

Figure 5. Population sizing model: theoretical prediction versus empirical results for different types of subproblems.

### 5.2. *Discussion*

Since the proposed population sizing model has been validated by the empirical results in the previous section, it may be a representative model for entropy-based perturbation linkage identification methods. Observing closely to the population size required by ILI, we can find that the population size is bounded by $\mathcal{O}(\log m)$ which leads to the requirement of $\mathcal{O}(m \log m)$ evaluation time. Comparing the population size to other methods's, $\mathcal{O}(m \log m)$ with $(1 - 1/m)$ accuracy for EDAs and $\mathcal{O}(\log m)$ for $D^5$, the perturbation based method requires much less population size than EDAs do. This implies that the perturbation operations, transforming the all-variable relevant fitness function into a fitness difference function that depends on only fewer variables, do help to reduce the population size required for linkage discovery by $\mathcal{O}(m)$. Comparing the numerical results, it can be observed that the accuracy of linkage discovery does not seem to guarantee efficient optimisation. To the best of our limited knowledge, the relationship between optimisation efficiency and the accuracy of linkage discovery is still elusive and requires further investigations. Nevertheless, linkage discovery is indeed helpful to optimisation. The question is how accurate is enough for efficient optimisation and how large the population size is required for the demanded linkage discovery accuracy.

## 6. Conclusions

In this article, we proposed a population sizing model for ILI. This model, assessing the information gain of selecting a variable as dependent via the statistics of sub-population sizes that are categorised by the candidate variables, is simple and can accurately approximate the population size which is sufficiently large for ILI to meet a given linkage error rate. The kernel of our population sizing model is the relationship between the number of training instances and the decision tree error rate. This kernel may potentially help analysing

Table 2. Fitness. differences of different sub-functions.

| Substring | | | | trap$_4$ | | | nith$_4$ | | | valley$_4$ | | | tmmp$_4$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $s_1$ | $s_2$ | $s_3$ | $s_4$ | $f_o$ | $f_n$ | $df_1$ | $f_o$ | $f_n$ | $df_1$ | $f_o$ | $f_n$ | $df_1$ | $f_o$ | $f_n$ | $df_1$ |
| $\overline{0}$ | 0 | 0 | 0 | 2 | 3 | 1 | 0 | 0 | 0 | 2 | 4 | 2 | 1 | 0 | −1 |
| $\overline{0}$ | 0 | 0 | 1 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 1 | −1 |
| $\overline{0}$ | 0 | 1 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 1 | −1 |
| $\overline{0}$ | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 2 | 0 | −2 | 1 | 2 | 1 |
| $\overline{0}$ | 1 | 0 | 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 1 | −1 |
| $\overline{0}$ | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 2 | 0 | −2 | 1 | 2 | 1 |
| $\overline{0}$ | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 2 | 0 | −2 | 1 | 2 | 1 |
| $\overline{0}$ | 1 | 1 | 1 | 4 | 0 | −4 | 4 | 0 | −4 | 4 | 2 | −2 | 4 | 1 | −3 |
| $\overline{1}$ | 0 | 0 | 0 | 3 | 2 | −1 | 0 | 0 | 0 | 4 | 2 | −2 | 0 | 1 | 1 |
| $\overline{1}$ | 0 | 0 | 1 | 2 | 1 | −1 | 0 | 0 | 0 | 2 | 0 | −2 | 1 | 2 | 1 |
| $\overline{1}$ | 0 | 1 | 0 | 2 | 1 | −1 | 0 | 0 | 0 | 2 | 0 | −2 | 1 | 2 | 1 |
| $\overline{1}$ | 0 | 1 | 1 | 1 | 0 | −1 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 1 | −1 |
| $\overline{1}$ | 1 | 0 | 0 | 2 | 1 | −1 | 0 | 0 | 0 | 2 | 0 | −2 | 1 | 2 | 1 |
| $\overline{1}$ | 1 | 0 | 1 | 1 | 0 | −1 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 1 | −1 |
| $\overline{1}$ | 1 | 1 | 0 | 1 | 0 | −1 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 1 | −1 |
| $\overline{1}$ | 1 | 1 | 1 | 0 | 4 | 4 | 0 | 4 | 4 | 2 | 4 | 2 | 1 | 4 | 3 |

other entropy based linkage discovery methods as well and thus help linkage discovery algorithms to improve the performance obtained with traditional GAs in problems which require finding interdependent decision variables to the desired outputs.

Furthermore, the population sizing analysis approach proposed in this study may also be applied to approximate the learning curve of a decision tree, given the maximum possible number of relevant attribute variables, the number of total attribute variables, and the number of possible values of the target variable. Predicting a learning algorithm's learning curve, which illustrates the relationship between the accuracy of the learned model and the number of training instances, has been one of the critical issues in machine learning. Currently, most studies approximate the learning curve of an algorithm with mathematical models (Gu, Hu, and Liu 2001; Morgan, Daugherty, Hilchie, and Carey 2003) such as power law model, logarithm model, and the like, or empirically predict the learning curve (Leite and Brazdil 2007). For this facet of research in machine learning, the proposed approach may provide some insights into the prediction of an learning algorithm's learning curve. It may be helpful in the analysis of data mining aided algorithms and thus improves the algorithmic performance in applications.

## Acknowledgements

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 grateful to the National Center for High-performance Computing for computer time and facilities.

## Notes on contributors

***Jih-Yiing Lin*** received her BS and MS degrees in Electronics Engineering from National Chiao Tung University, Taiwan, in 2000 and 2002, respectively. She had been with Sunplus Technology, Hsinchu, during 2002–2007. She is currently working towards the PhD degree in Computer Science at National Chiao Tung University, Taiwan. Her research interests in the field of genetic and evolutionary computation include theories, working principles, particle swarm optimisation and linkage learning techniques.

***Ying-ping Chen*** received his BS and MS degrees in Computer Science and Information Engineering from National Taiwan University, Taiwan, in 1995 and 1997, respectively, and the PhD degree in 2004 from the Department of Computer Science, University of Illinois at Urbana-Champaign, Illinois, USA. He has been an Assistant Professor from 2004 to 2009 and an Associate Professor since 2009 in the Department of Computer Science, National Chiao Tung University, Taiwan. His research interests in the field of genetic and evolutionary computation include theories, working principles, particle swarm optimisation, estimation of distribution algorithms, linkage learning techniques and dimensional/facet-wise models.

## References

Arnold, B.C., Balakrishnan, N., and Nagaraja, H.N. (1993), *A First Course in Order Statistics*, NY, USA: John Wiley and Sons.

Baluja, S. (1994), 'Population-based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning', Technical report, Carnegie Mellon University, Pittsburgh, PA, USA.

Baluja, S., and Davies, S. (1997), 'Using Optimal Dependency-trees for Combinational Optimization', in *Proceedings of the Fourteenth International Conference on Machine Learning (ICML-97)*, San Francisco, CA, USA: Morgan Kaufmann Publishers, pp. 30–38.

Chen, Y.-P., Chuang, C.-Y., and Huang, Y.-W. (2011), 'Inductive Linkage Identification on Building Blocks of Different Sizes and Types', *International Journal of Systems Science*, First published on: 11 April 2011 (iFirst).

Choi, S.S., Jung, K., and Moon, B.R. (2009), 'Lower and Upper Bounds for Linkage Discovery', *IEEE Transactions on Evolutionary Computation*, 13, 201–216.

Chuang, C.Y., and Chen, Y.P. (2007), 'Linkage Identification by Perturbation and Decision Tree Induction', in *Proceedings of 2007 IEEE Congress on Evolutionary Computation (CEC 2007)*, pp. 357–363.

Cooper, G.F., and Herskovits, E.H. (1992), 'A Bayesian Method for the Induction of Probabilistic Networks from Data', *Machine Learning*, 9, 309–347.

de Bonet, J., Isbell, C., and Viola, P. (1997), 'MIMIC: Finding Optima by Estimating Probability Densities', in *Advances in Neural Information Processing Systems* (Vol. 9), Cambridge, MA, USA: The MIT Press, p. 424.

Gámez, J.A., Mateo, J.L., and Puerta, J.M. (2007), 'EDNA: Estimation of Dependency Networks Algorithm', in *Proceedings of the 2nd International Work-conference on the Interplay Between Natural and Artificial Computation, Part I: Bio-inspired Modeling of Cognitive Tasks*, pp. 427–436.

Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Goldberg, D.E., Korb, B., and Deb, K. (1989), 'Messy Genetic Algorithms: Motivation, Analysis, and First Results', *Complex Systems*, 3, 493–530.

Gu, B., Hu, F., and Liu, H. (2001), 'Modelling Classification Performance for Large Data Sets', in *Advances in Web-aGe Information Management*, eds. X. Wang, G. Yu, and H. Lu, Berlin/Heidelberg: Springer, pp. 317–328.

Harik, G.R. (1999), 'Linkage Learning via Probabilistic Modeling in the ECGA', IlliGAL Report No. 99010, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, USA.

Harik, G.R., Lobo, F.G., and Goldberg, D.E. (1999), 'The Compact Genetic Algorithm', *IEEE Transactions on Evolutionary Computation*, 3, 287–297.

Harik, G.R. (1997), 'Learning Gene Linkage to Efficiently Solve Problems of Bounded Difficulty using Genetic Algorithms', PhD Dissertation, University of Michigan, Ann Arbor, MI, USA.

Heckerman, D., Geiger, D., and Chickering, D.M. (1995), 'Learning Bayesian Networks: The Combination of Knowledge and Statistical Data', *Machine Learning*, 20, 197–243.

Jiang, Q., Wang, Y., and Yang, X.Q. (2009), 'A Framework for Estimation of Distribution Algorithms Based on Maximum Entropy', in *Proceedings of the 5th International Conference on Natural Computation*, Tianjin, China, Piscataway, NJ, USA: IEEE Press, pp. 7–11.

Kargupta, H. (1996), 'SEARCH, Polynomial Complexity, and the Fast Messy Genetic Algorithm', Ph.D. Dissertation, University of Illinois at Urbana-Champaign, Urbana, IL, USA.

Kim, D.W., de Silva, C.W., and Park, G.T. (2010), 'Evolutionary Design of Sugeno-type Fuzzy Systems for Modelling Humanoid Robots', *International Journal of Systems Science*, 41, 875–888.

Leite, R., and Brazdil, P. (2007), 'An Iterative Process for Building Learning Curves and Predicting Relative Performance of Classifiers', in *Proceedings of the Aritficial Intelligence 13th Portuguese Conference on Progress in Artificial Intelligence*, Guimarães, Portugal, Berlin, Heidelberg: Springer-Verlag, pp. 87–98.

MacKay, D.J.C. (2003), *Information Theory, Inference and Learning Algorithms*, Cambridge: Cambridge University Press.

Morgan, J., Daugherty, R., Hilchie, A., and Carey, B. (2003), 'Sample Size and Modeling Accuracy of Decision Tree Based Data Mining Tools', *Academy of Information and Management Science Journal*, 6, 71–99.

Mühlenbein, H., and Höns, R. (2005), 'The Estimation of Distributions and the Minimum Relative Entropy Principle', *Evolutionary Computation*, 13, 1–27.

Mühlenbein, H., and Mahnig, T. (1999), 'FDA – A Scalable Evolutionary Algorithm for the Optimization of Additively Decomposed Functions', *Evolutionary Computation*, 7, 353–376.

Munetomo, M., and Goldberg, D. (1998), 'Identifying Linkage by Nonlinearity Check', IlliGAL Report No. 98012, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, USA.

Munetomo, M., and Goldberg, D.E. (1999), 'Identifying Linkage Groups by Nonlinearity/Non-monotonicity Detection', in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, pp. 433–440.

Pelikan, M., Goldberg, D.E., and Cantú-Paz, E. (1999), 'BOA: The Bayesian Optimization Algorithm', in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)*, pp. 525–532.

Pelikan, M., and Mühlenbein, H. (1999), 'The Bivariate Marginal Distribution Algorithm', in *Advances in Soft Computing – Engineering Design and Manufacturing*, Berlin, Germany: Springer-Verlag, pp. 521–535.

Pelikan, M., Sastry, K., and Goldberg, D.E. (2002), 'Scalability of the Bayesian Optimization Algorithm', *International Journal of Approximate Reasoning*, 31, 221–258.

Quinlan, J.R. (1993), 'Induction of Decision Trees', in *Readings in Knowledge Acquisition and Learning: Automating the Construction and Improvement of Expert Systems*, San Francisco, CA, USA: Morgan Kaufmann Publishers, pp. 349–361.

Rissanen, J. (1978), 'Modeling by Shortest Data Description', *Automatica*, 14, 465–471.

Rissanen, J. (1989), *Stochastic Complexity in Statistical Inquiry Theory*, River Edge, NJ, USA: World Scientific Publishing Co.

Rissanen, J. (1996), 'Fisher Information and Stochastic Complexity', *IEEE Transactions on Information Theory*, 1, 40–47.

Santana, R., Larrañaga, P., and Lozano, J.A. (2010), 'Learning Factorizations in Estimation of Distribution Algorithms Using Affinity Propagation', *Evolutionary Computation*, 18, 515–546.

Saridakis, K., and Dentsoras, A. (2009), 'Integration of Genetic Optimisation and Neuro-fuzzy Approximation in Parametric Engineering Design', *International Journal of Systems Science*, 40, 131–145.

Sastry, K., and Goldberg, D.E. (2004), 'Designing Competent Mutation Operators Via Probabilistic Model Building of Neighborhoods', in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*, pp. 114–125.

Solteiro Pires, E.J., Tenreiro Machado, J.A., and de Moura Oliveira, P.B. (2006), 'Dynamical Modelling of a Genetic Algorithm', *Signal Processing*, 86, 2760–2770.

Streeter, M.J. (2004), 'Upper Bounds on the Time and Space Complexity of Optimizing Additively Separable Functions', in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2004)*, pp. 186–197.

Ting, C.K., Zeng, W.M., and Lin, T.C. (2010), 'Linkage Discovery Through Data Mining', *IEEE Computational Intelligence Magazine*, 5, 10–13.

Tsuji, M., Munetomo, M., and Akama, K. (2006), 'Linkage Identification by Fitness Difference Clustering', *Evolutionary Computation*, 14, 383–409.

Wang, K. (2009), 'Application of Genetic Algorithms to Robot Kinematics Calibration', *International Journal of Systems Science*, 40, 147–153.

Yu, T.L., Sastry, K., Goldberg, D.E., and Pelikan, M. (2007), 'Population Sizing for Entropy-based Model Building in Discrete Estimation of Distribution Algorithms', in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2007)*, pp. 601–608.

Zwillinger, D., and Kokoska, S. (2000), *CRC Standard Probability and Statistics Tables and Formulae*, FL, USA: CRC Press LLC.