IIE Transactions (2014) **46**, 164–184

ISSN: 0740-817X print / 1545-8830 online
DOI: 10.1080/0740817X.2013.803639

A group testing algorithm with online informational learning

EUGENE KAGAN and IRAD BEN-GAL*

Department of Industrial Engineering, Tel-Aviv University, Ramat-Aviv, 69978, Israel
E-mail: Bengal@tau.ac.il

Received November 2011 and accepted April 2013

An online group testing method to search for a hidden object in a discrete search space is proposed. A relevant example is a search after a nonconforming unit in a batch, while many other applications can be related. A probability mass function is defined over the search space to represent the probability of an object (e.g., a nonconforming unit) to be located at some point or subspace. The suggested method follows a stochastic local search procedure and can be viewed as a generalization of the Learning Real-Time A* (LRTA*) search algorithm, while using informational distance measures over the searched space. It is proved that the proposed Informational LRTA* (ILRTA*) algorithm converges and always terminates. Moreover, it is shown that under relevant assumptions, the proposed algorithm generalizes known optimal information-theoretic search procedures, such as the offline Huffman search or the generalized optimum testing algorithm. However, the ILRTA* can be applied to new situations, such as a search with side information or an online search where the probability distribution changes. The obtained results can help to bridge the gap between different search procedures that are related to quality control, artificial intelligence, and information theory.

Keywords: Information search, search and screening, group testing, real-time heuristic search

1. Introduction

The group testing strategy was originally proposed during WWII by Dorfman (1943) as a sequential testing strategy to identify syphilis-infected blood samples. The strategy inspects a batch of units simultaneously (originally blood samples that were blended together), taking into account that most probably these units are conforming (uninfected). Therefore, there is a high probability to approve many units by a single group test. If, however, the test indicates that there exists a nonconforming unit (infected blood sample) in the set, then the set is partitioned into subsets, to each of which a smaller group testing is applied. This iterative procedure of partitioning and testing terminates when the nonconforming unit is found. The main motivation of the strategy is to minimize the testing efforts in terms of minimum average number of tests in a sort of a zooming-in search procedure.

The applicability of group testing to quality and reliability is evident and has been recognized along the years. Dorfman (1943, p. 436) himself associated group testing with the classic sequential-sampling procedure for quality inspection:

> Often in testing the results of manufacture, the work can be reduced greatly by examining only a sample of the population and rejecting the whole if the proportion of defectives in the sample is unduly large.

A general formulation of the group testing can be obtained by representing it as problem of search after a hidden target in a discrete domain. Such a search is a fundamental problem that can be associated with numerous methods in stochastic optimization, knowledge engineering, and data retrieval. In particular, consider a target (e.g., a nonconforming unit) that is located somewhere within a discrete domain and a searcher (testing agent) that is looking for this target. At each step, the action available to the searcher is to check a sub-domain in order to determine whether the target is located somewhere within this sub-domain or not. The procedure terminates if the searcher finds the target in a sub-domain that contains only one point. That is, if the searcher finds the target with certainty. The searcher goal is to choose a sequence of sub-domains such that the search terminates in a minimal expected number of steps.

Such an example is found in Herer and Raz (2000) that was motivated from the pasteurized foods industry and pertains to the process of filling plastic cups with cottage cheese drawn from a large container. For the case of the concentration of curd in the container deviating from the given specifications, all of the cups filled with cottage cheese after this point in time are regarded as nonconforming units. Moreover, the authors indicate that since it is not practical to wait for the laboratory results in mid-batch, all of the product in the container is used to fill up the cups, which

*Corresponding author

are stamped with the processing time and held until the test results are ready. If one or more of the cups sampled shows an unacceptable concentration of curd, then it and all cups that were produced afterwards are discarded. If one or more of the cups sampled shows that the level of curd is within the acceptable range, then it and all cups that were produced before it are accepted. Thus the key question in Herer and Raz (2000) is how many and which product units (cups) should be sampled to find the first nonconforming unit at minimal cost. The equivalence to group testing is clear since each tested cup represents all of the untested cups that were produced before it up to the last tested cup. Note that a simultaneous testing of several cups can be conducted at the same time. Moreover, the results can be represented by more than a binary "conforming" and "nonconfirming" result that can also be prone to observation noise. The proposed model addresses the case of group testing that can be considered also as a search after a nonconfirming unit. We further consider that such a search can be updated by new side information gathered through the search.

The group testing approach has many practical applications in modern real-life scenarios. One such example is the search procedures to locate a specific mobile device in a cellular network (e.g., see Krishnamachari *et al*. (2004)). Since the mobile device often moves between network cells, the network controller has imperfect information regarding the device location and is required, at the time of an incoming call, to find the device location quickly. The process of searching for a mobile station over the cellular network is called *paging*. The network controller pages for a particular cellular device in a relatively large area, obtains an indication of were it is located within this area, and then zooms into this sub-domain until the cellular telephone is reached. In the framework of the considered search problem, the mobile device is considered as a moving target. Another practical example is the identification of a bad segment in a communication network, where the group of connected segments can be tested by sending an input to the first segment and observing the output at the last segment. A related example can be found in Brice and Jiang (2009) that considers a multistage fault detection and isolation procedure with applications to commercial video broadcasting systems, as suggested by AT&T.

Mining data objects or records with specific features in large databases is another related area that uses a group testing approach. In this case, the tested group is represented by those objects that share similar features, such that a database query can identify all of the objects in a group, while a specific searched object might be an object with specific features (see, e.g., Vitter (1999) and Ben-Gal (2004)). Sensor-based location management in service applications provides a lot of ground to find objects with the same sensors signature. Finally, let us note that in the modern cloud-computing environment, a group testing approach could be of great benefit for predictive maintenance, as it can be used to identify specific faulty modules (sometime associated with remote services) that are stored in the cloud by using input/output testing of different groups of modules until the faulty module is identified.

This article introduces a group testing algorithm to search for a target and formulates it in a framework of an *online Stochastic Local Search* (SLS). Unlike the offline search procedure, an online search does not assume that all of the information regarding the possible locations of the target is available *a priori*. The advantage of an online search procedure lies in its robustness with respect to imperfect information, thus its ability to cope with unexpected changes during the search, including the accumulation of new information. This is in contradiction to *offline* procedures, such as the optimal Huffman search procedure (Huffman, 1952), where even a slight deviation in the probability of the located target can result in an extremely inefficient search. Moreover, an online search procedure can absorb new information that is revealed during the search.

In the framework of SLS, the searcher (a problem solver in the general case) starts with some initial location in the search space (feasible solution in the general case) and iteratively moves from its current location to a neighboring location, while adaptively obtaining information through observations. The neighborhood of each location includes a set of candidate locations on which the searcher has more accurate information; thus, the decision at each step is obtained mainly on the basis of local information with linear complexity, which reduces the search space immensely and avoids computational intractability due to exponential complexity. We follow the SLS approach for the group testing problem and particularly consider the *Learning Real-Time A** (LRTA*) algorithm that is well known in artificial intelligence (Korf, 1990). The original LRTA* algorithm addresses a different search problem; mainly, it looks for the shortest path between a searcher and a target on a weighted graph. It does not apply group testing at all. However, it is found that with new definitions and routines—mainly related to the metric of the search space and the distance measures used between the searcher and the target—a generalized version of this algorithm, which we call *Informational LRTA** (ILRTA*) algorithm, leads to promising results for the considered group testing search.

The ILRTA applies a group testing search to a set of points, defined as a partition of the sample space. It uses entropy-based distance measures, such as the Rokhlin distance (Rokhlin, 1967) and the Ornstein distance (Ornstein, 1974), to quantify the distances between different partitions of the sample space. The distance measures satisfy the metric requirements; therefore, they can be applied within a similar framework of the LRTA* algorithm.

In this article, we study the main properties of the suggested ILRTA* algorithm. We compare it against the optimal Huffman search, which is an *offline* bottom-up search procedure, and against the near-optimal GOTA, which is an *online* top-down search procedure. We show that under a bottom-up search condition with fixed information, the ILRTA* generates an optimal Huffman search tree, while under a top-down search condition it performs according to the Generalized Optimum Testing Algorithm (GOTA). To the best of our knowledge, such a unified scheme of the two procedures and has not been previously proposed. Moreover, unlike these two search procedures, the ILRTA* is more robust to changes of side information, such as the new distance estimations from the searcher to the target or new probability estimates of the target location. This property is shown to yield a great advantage when such side information is available; for example, when the assumptions regarding the location of the target change during the search. The performance of the ILRTA* algorithm is studied by numerical simulations showing its superiority under certain condition. Finally, the ILRTA* allows a straightforward generalization to a search by multiple searchers after multiple targets (Kagan and Ben-Gal, 2010), which is relevant to sensor fusion problems. It can be applied with slight changes to a search after a moving target (Kagan and Ben-Gal, 2006). Note that these two latter applications are beyond the scope of this article.

The article is organized as follows. Section 2 presents a review of the related literature. Section 3 describes a general group testing model for an online testing procedure and presents information-theoretic metrics that are later used for the search. Section 4 formulates the suggested ILRTA* algorithm and presents its main properties. It establishes the relations between the ILRTA* algorithm and known information-theoretic testing algorithms, such as the optimal *offline* Huffman search (Zimmerman, 1959) and the near-optimal GOTA (Hartmann *et al*., 1982). Section 5 presents simulated results and statistical comparisons between the ILRTA* algorithm and known search procedures. Section 6 provides a general discussion and summarizes the work. The Appendix contains the proofs of the statements that appear in the body of text.

2. Literature review

One of the earliest works on group testing was suggested by Zimmerman (1959) and was based on the Huffman coding procedure (1952), which is well known in information and coding theory. The analogy between testing (e.g., to find a nonconforming unit) and coding relies on the fact that each unit in the set can be decoded by the sequenced testing outcomes of subsets to which it belongs (Ben-Gal, 2004). The length of the code is thus analogous to the length of the testing procedure that is required to identify the nonconforming unit. Zimmerman's offline method is constructed by using the probability of each unit to be the nonconforming one. An optimal testing tree is constructed offline by the Huffman coding procedure and then the test procedure follows this testing tree, assuming that no new information is revealed. Since the Huffman coding algorithm is optimal in the sense of minimal average code length, the analogous search procedure is optimal in the sense of minimal average search length (Zimmerman, 1959).

Other group testing methods that are based on coding procedures have been suggested. Ben-Gal (2004) analyzed the popular *weight balance tree* group testing algorithm, which is based on the simple principle of successive partitions of the search set into equi-probable subsets (known also as Shannon–Fano coding). At each step of the algorithm, the subset that contains the searched item (i.e., nonconforming unit) is partitioned into equi-probable subsubsets and the process repeats itself until the searched for item is found. Herer and Raz (2000) considered a similar group testing search for a defective product in a lot produced by a process with a constant failure rate, as described above.

A different online search approach that looks for the shortest path between the search agent and the target has been suggested by Korf (1990). He developed a real-time version of the known A* algorithm that is based on an SLS method with heuristic distance measures. His real-time algorithm of search for a static target is known as the LRTA* algorithm. The LRTA* algorithm operates on a given graph in which the vertices represent states (locations in the search space in our case) while the edges between the states are weighted (distances between locations in our case). In the considered search problem, the LRTA* algorithm applies heuristic distance estimations between the feasible target states and the target's state. The iterative LRTA* algorithm finds the shortest path between the searcher's initial state and the target's state.

Based on the LRTA* algorithm, Ishida and Korf (1991, 1995) suggested an algorithm to search for a moving target. The results of these algorithms were later reviewed by Shimbo and Ishida (2003). Subsequently, Bulitko and Lee (2006) and Bulitko *et al*. (2007) proposed a unified framework for a general search procedure for the LRTA* algorithm. Koenig and Simmons (1995), Koenig and Likhachev (2006), and Sun *et al*. (2009) considered an implementation of the LRTA* algorithm over nondeterministic domains and suggested suitable adaptive algorithms that can be applied for the search for both static and moving targets. Other versions of the LRTA* algorithm were developed by the IRCL research group (IRCL Research Group, 2012). Note that all of the above-mentioned algorithms as opposed to this study do not follow a group testing approach; thus, the searcher checks a single point in the domain at each search stage. Moreover, these algorithms are focused on finding the shortest path between the searcher and the target and not on finding the target itself, as aimed at in this study.

Using an information-theoretic approach, Hartman *et al.* (1982) suggested a near-optimal search procedure known as the GOTA. In this online algorithm, policy decision and searching are conducted simultaneously. The GOTA follows a general objective of a maximum entropy search, while implementing some properties of the available partitions of the sample space. In this article, we generalize the GOTA by using SLS principles and the ideas that appear in the LRTA* algorithm (Korf, 1990).

Further generalization of the information-theoretic procedures of search was also suggested by Abrahams (1994) and has been applied to group testing tasks and to the construction of decision trees (Ben-Gal, 2004; Ben-Gal *et al*. 2008; Herer and Raz, 2000). Other group testing methods have been suggested under specific assumptions (Gupta and Malina, 1999). Graff and Roeloffs (1974) considered a group testing procedure in the presence of noise. Yadin and Zacks (1988, 1990) considered shadowing and visibility conditions for a search in an heterogeneous sample space in the presence of obstacles. Related studies of the considered search problem were constructed by relying on dynamic programming and stochastic optimization methods (Ross, 1983; Bertsekas, 1995).

3. A group testing search scheme

3.1. *General formulation of the problem*

Let us start with a general formulation of the search procedure that corresponds to Dorfman's group testing scheme (Dorfman, 1943). Recall that for the purpose of generality the *nonconforming unit* is called the *target*, while the *testing agent* is called the *searcher*.

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a finite *sample space* of n points where the target can be located. At each time moment $t = 0, 1, 2, \ldots$, the searcher chooses a subset $A^t \subset X$ and observes it. It is assumed that the observation is perfect; hence, the resulting observation $z^t = 1$ if the target is located in one of the points of A^t, and $z^t = 0$ otherwise. All subsets $A \subset X$ that are available to the searcher are included in the *search space* $\mathbf{X}$. If the searcher's choice of subsets A is not restricted, then the search space is the power set $\mathbf{X} = 2^X$ of the sample space X that consists of all possible subsets of X, including the empty set $\emptyset$ and the set X itself.

The general search procedure of search after a static target is formulated as follows (Ross, 1983; Bertsekas, 1995):

1. The target is located at point $x^0 \in X$, which is unknown to the searcher.
2. At time moment t, the searcher chooses a subset $A^t \in \mathbf{X}$, observes it, and obtains an observation result:

$$z^t = \mathbf{1}(x^0,\ A^t) = \begin{cases} 1 & \text{if } x^0 \in A^t, \\ 0 & \text{otherwise.} \end{cases}$$

3. If $z^t = 1$ and $A^t = \{x^0\}$ or if $z^t = 0$ and $X \backslash A^t = \{x^0\}$ then the search terminates. Otherwise, the time increases as $t = t + 1$ and the search continues from step 2.

It is assumed that the searcher does not know where the target is located; however, he/she knows the probability distribution of the target's location over points in the sample space. This is equivalent to having the distribution of the location of a nonconforming unit, which is a viable assumption when the characteristics of the production system are known (Herer and Raz, 2000; Ben-Gal, 2004). In addition, it is assumed that the searcher has perfect information of the observation result z^t; i.e., without observation noise. The goal of the study is to find a test policy for choosing the subsets $A^t \in \mathbf{X}$ such that the search procedure terminates in a minimal expected number of steps.

The approach to search over partitions of the search space now follows.

Recall that at each time moment t the searcher obtains a perfect observation result $z^t \in \{0, 1\}$. The selection of a set $A^t \in \mathbf{X}$ and its observation implies that if $z^t = \mathbf{1}(x^0, A^t) = 1$, then $z^t = \mathbf{1}(x^0, X \backslash A^t) = 0$ and vice versa. Hence, such a selection can be represented by a partition that we call the *searcher's partition*, $\alpha^t = \{A^t, X \backslash A^t\}$. Similarly, the target location can be represented by the, target's partition $\tau = \{\{x^0\}, X \backslash \{x^0\}\}$, where $x^0 \in X$ is the point in which the target is located. Accordingly, the above general search procedure can be reformulated on the set χ of partitions as follows (Jaroszewicz, 2003; Kagan and Ben-Gal, 2006):

1. The target chooses a partition $\tau \in \chi$, which is unknown to the searcher.
2. At time moment t, the searcher chooses a partition $\alpha^t \in \chi$.
3. If $\alpha^t = \tau$ or α^t is a refinement of τ then the search terminates. Otherwise, time increases to $t = t + 1$ and the search continues from step 2.

The searcher's goal, as indicated above, is to find a policy for choosing a sequence of partitions $\alpha^t \in \chi$ that guarantees termination of the search within a minimal expected number of steps. In the algorithms below the searcher is allowed to choose both binary and non-binary partitions that correspond to the number of possible subgroups. Each subgroup is represented by a symbol that is associated with the result of the group test. The partitions are refined iteratively on the basis of previously chosen partitions. For example, in a ternary partition the three possible groups of items might be associated with positive, negative, and undetermined results. It is assumed that the searcher's choice is based on the observation results, in particular, by using these to update the probability of the target's location. Notice that from a group testing point of view, a feasible selection of a final partition $\gamma = \{X, \emptyset\}$ corresponds to the test of a full sample space X.

In the next sections we implement the general search procedure by relying on the SLS/LRTA* algorithm (Korf, 1990) and particularly by defining partitions as states on the LRTA* graph. Since the general LRTA* algorithm requires a distance measure between states, we start by introducing two metrics that can be implemented directly on the set of partitions.

3.2. *Distance measures and search space metrics*

Recall that $X = \{x_1, x_2, \ldots, x_n\}$ is a finite sample space. Define a probability mass function $p_i = p(x_i) = \Pr\{x^0 = x_i\}$ representing the probability of the target being located at point $x_i \in X$ (or, equivalently, the probability of a unit to be nonconforming), $i = 1, 2, \ldots, n$, where $\sum_{i=1}^{n} p(x_i) = 1$. We call it the *location probability*. If there is no available information on the target location, then it is assumed that the probability is uniform $p_1 = p_2 = \ldots = p_n$ based on the principle of the maximum entropy (Cover and Thomas, 1991).

Let χ be a set of partitions of the sample space X and let $\alpha = \{A_1, A_2, \ldots\} \in \chi,\ A_i \cap A_j = \emptyset,\ i \neq j,\ \cup_{A\in\alpha} = X$, be a partition of X. For a subset $A \in \alpha$, denote $p(A) = \sum_{x\in A} p(x)$. The *entropy* of a partition α is defined as follows (Sinai, 1977):

$$H(\alpha) = -\sum\nolimits_{A\in\alpha} p(A)\log p(A), \tag{1}$$

where, following the conventions in information theory (Cover and Thomas, 1991), the logarithm is taken to base 2 and $0\log 0 \equiv 0$.

Consider a partition $\beta = \{B_1, B_2, \ldots\} \in \chi,\ B_i \cap B_j = \emptyset,\ i \neq j$. Then, the *conditional entropy* of partition α given partition β is defined as (Sinai, 1977)

$$H(\alpha \mid \beta) = -\sum\nolimits_{B\in\beta}\sum\nolimits_{A\in\alpha} p(A, B)\log p(A \mid B), \tag{2}$$

where $p(A, B) = p(A \cap B)$ and $p(A \mid B) = p(A \cap B)/p(B)$.

Denote the *refinement* between two partitions α and β by "$\prec$." If for every $A \in \alpha$ there exists $B \in \beta$ such that $A \subseteq B$, then we say that α *refines* β or that α is a *refinement* of β and write $\beta \prec \alpha$. The properties of the entropy and the conditional entropy of partitions are given by the following theorems. These properties are used by the proposed search algorithm.

Theorem 1 (Sinai, 1977). *For the partitions of X the following statements hold:*

(i) *if* $\alpha \prec \beta$ then $H(\alpha) \leq H(\beta)$;
(ii) $H(\alpha \mid \beta) = 0$ *if and only if* $\alpha \prec \beta$;
(iii) *if* $\alpha \prec \alpha'$ *then* $H(\alpha \mid \beta) \leq H(\alpha' \mid \beta)$;
(iv) *if* $\beta \prec \beta'$ *then* $H(\alpha \mid \beta') \leq H(\alpha \mid \beta)$.

Theorem 2 (Sinai, 1977; Hartmann *et al.*, 1982). *Let* α *and* β *be two partitions. Then* $\beta \succ \alpha$ *if and only if* $H(\beta \mid \alpha) = H(\beta) - H(\alpha)$.

Let us now present the Rokhlin distance (Rokhlin, 1967) that initially was suggested for the purposes of dynamical systems theory. The Rokhlin distance between partitions α and β of the sample space is defined as follows (Rokhlin, 1967):

$$d(\alpha, \beta) = H(\alpha \mid \beta) + H(\beta \mid \alpha), \tag{3}$$

where conditional entropy is defined by Equation (2).

In the next considerations, the metric properties of the Rokhlin distance (Sinai, 1977) are used:

(i) $d(\alpha, \beta) \geq 0$;
(ii) $d(\alpha, \alpha) = 0$;
(iii) $d(\alpha, \beta) \leq d(\alpha, \xi) + d(\xi, \beta)$ for any partition ξ of X.

Note that independent of Rokhlin (1967), López De Mántaras (1991) presented a definition of the same metric for partitions of the finite sample space X and applied it to attribute selection and search trees. The same form of an informational distance between two random variables was also defined by Lloris-Ruiz *et al*. (1993). On the basis of this definition, Jaroszewicz (2003) considered partitions of X with equi-probable points and applied it to problems of data mining. An independent research in data mining based on the similar ideas was conducted by Peltonen (2004).

Let us now clarify the relation between the Rokhlin metric and the popular Shannon's *mutual information* concept that is used in known search algorithms. Similar to the joint entropy and the mutual information between two random variables (Cover and Thomas, 1991), the joint entropy and the mutual information between two partitions α and β of X are defined as follows:

$$H(\alpha, \beta) = H(\alpha) + H(\beta \mid \alpha) = H(\beta) + H(\alpha \mid \beta), \tag{4}$$

$$I(\alpha; \beta) = H(\alpha) - H(\alpha \mid \beta) = H(\beta) - H(\beta \mid \alpha). \tag{5}$$

Thus, based on Equations (5) and (3), the relation between the mutual information $I(\alpha; \beta)$ and the Rokhlin metric $d(\alpha, \beta)$ for the partitions α and β is the following:

$$d(\alpha, \beta) = H(\alpha, \beta) - I(\alpha; \beta). \tag{6}$$

The Ornstein distance between partitions is now presented. Let α and β be two partitions of the sample space X with a probability mass function p. The *Ornstein distance* between α and β is defined as follows (Ornstein, 1974):

$$d_{\mathrm{Orn}}(\alpha,\ \beta) = p(X \setminus \cup_{i=1}^{k} (A_i \cap B_i)), \tag{7}$$

where $A_i \in \alpha$, $B_i \in \beta$, and $k = \max(|\alpha|, |\beta|)$. Here $|\cdot$ is the cardinality of the partition. If $|\alpha| > |\beta|$, then β is completed by empty sets, and if $|\beta| > |\alpha|$, then empty sets are added to α.

The Ornstein distance meets the metric requirements (Ornstein, 1974). The relation between the Rokhlin distance and the Ornstein distance is given in Lemma 1. This lemma supports the *admissibility property* of the distance and its estimation, which is required by the LRTA* algorithm and will be used in the ILRTA* algorithms suggested below.

Lemma 1. *If* α *and* β *are partitions of the same space X with probability function p, then* $d_{\mathrm{Orn}}(\alpha,\ \beta) \leq d(\alpha,\ \beta)$.

The proof of this lemma is given in the Appendix.

In light of Theorems 1 and 2, the distances depend on the structure of the considered partitions. Below we consider the dependence of the Rokhlin distances on the location probabilities defined on the sample space X.

In general, let $(\chi, \prec)$ be the lattice of partitions of the sample space X with a partial order determined by the refinement relation $\prec$. Denote the Rokhlin metric for a uniform distribution of the target's locations (i.e., an equiprobable sample space X) by $d_u(\alpha, \beta)$ and the Rokhlin metric for some other probability mass function p by $d_p(\alpha, \beta)$. Let $\theta = \{X, \emptyset\}$ be the trivial partition of the sample space X. Then, the following lemma can be formulated.

Lemma 2. *If every partition* $\alpha \in (\chi, \prec)$, $\alpha \neq \theta$, *satisfies* $H_p(\alpha) \neq H_u(\alpha)$, *then for every pair* (α, β), $\alpha, \beta \in (\chi, \prec)$ *it is true that* $d_p(\alpha, \beta) < d_u(\alpha, \beta)$.

In other words, the lemma states that the Rokhlin distance depends on the probability distribution over a sample space, and for the uniform distribution the values of the Rokhlin distances are greater than those obtained by any other distribution. As seen below, such a property provides a basis for the probabilities updating scheme during the search, while maintaining the admissibility requirement of the distances. The proof of this lemma is given in the Appendix.

In the next section we apply the presented metrics to the formulated search problem and implement them in the proposed ILRTA* algorithm.

4. The ILRTA* algorithm and its main properties

The proposed ILRTA* algorithm can be considered in parallel with the LRTA* algorithm (Korf, 1990). Similar to the LRTA*, the ILRTA* algorithm operates on a graph. However, unlike the LRTA*, in the ILRTA* the graph vertices represent partitions and not states. These partitions stand for the possible results of the searcher's group tests. For example, if the searcher tests a subset of points where the target can be located, the relevant partition of points following the test is represented by the corresponding vertex in the graph. The searcher's decision to move to a certain vertex in the graph represents a selection and an execution of a specific group test. The target location can be represented by the *target's partition* that corresponds to the target's location point and the complementary subset of all the other points. The search is terminated when the searcher selects a partition that is either identical or more refined with respect to the target's partition; we thus call it the *final partition* since the search ends at that stage. Reaching such a partition implies that the searcher obtains the necessary information regarding the target location, as a result of the group test. Similar to the LRTA* algorithm (Korf, 1990), the edges between the vertices in the ILRTA* algorithm are weighted by the corresponding distances measures. However, since these distance measures are between pairs of partitions, the ILRTA* algorithm applies specific information-theoretic distance measures and, particularly, it uses the Rokhlin distance metric, which is based on the relative entropies of these partitions (Rokhlin, 1967).

Similarly to other SLS procedures (Koenig and Simmons, 1995; Koenig and Likhachev, 2006), in the ILRTA* algorithm the partitions with *known* distances to the current searcher's partition define a neighborhood. The distances to all other points in the graph have to be estimated and refined when new information is obtained from the next group tests. The iterative ILRTA* algorithm seeks to find the shortest path between the searcher's partition and the target's partition and, thus, selects a path that corresponds to the group tests results. In summary, similar to other search algorithms, including the LRTA*, the ILRTA* implies a heuristic search based on distance estimations between the target's partition and feasible search partitions. The uniqueness of the ILRTA* is that it uses informational metrics and non-binary partitions to represent a group testing search process over a graph of partitions (see Jaroszewicz (2003) and Peltonen (2004)).

As indicated above, the ILRTA* uses the Rokhlin metrics as heuristic distance measures between the searcher's and the target's partitions. As shown in later sections, the implementation of such metrics results in an algorithm that *generalizes* known information-theoretic search methods and can be applied to practical problems as described in the Introduction, including, for example, the search after a nonconforming unit in a batch. Let us now further explain the main concepts of the ILRTA* search.

Recall that $X = \{x_1, x_2, \ldots, x_n\}$ is a sample space with location probabilities $p_i = p(x_i)$, $i = 1, 2, \ldots, n$, $\sum_{i=1}^{n} p_i = 1$, and let χ be a set of partitions of X that includes both the *trivial partition* $\{X, \emptyset\}$ and the *discrete partition* $\{\{x_1\}, \{x_2\}, \ldots, \{x_n\}\}$ of the sample space X. Note that if there is no *a priori* knowledge about the location probabilities, the ILRTA* can be initiated with uniform (equiprobable) probabilities and refine the distances as the search proceeds and new information is gathered. Using uniform probabilities will extend the search time until the target is found. These probabilities are used to find the distance between possible partitions in the search graph. In particular, assume that for every pair of partitions $\alpha, \beta \in \chi$ one can apply the Rokhlin distance $d(\alpha, \beta)$ and a distance estimation $\tilde{d}(\alpha, \beta)$ such that they satisfy:

$$\tilde{d}(\alpha, \beta) \leq d(\alpha, \beta), \quad \alpha, \beta \in \chi. \tag{8}$$

Property (8) is called the *admissibility property*. Note that following Lemma 1, one can use the Ornstein metric as a distance estimation; i.e., $\tilde{d}(\alpha, \beta) = d_{\mathrm{Orn}}(\alpha, \beta)$.

Let $\gamma \in \chi$ be a final partition that satisfies the goal of the search; thus, γ guarantees that the searcher finds the target. In its limit, the final partition is equivalent to the discrete partition $\gamma = \{\{x_1\}, \{x_2\}, \ldots, \{x_n\}\}$, although such a refined partition is not always needed (for example, if the target is located at point x_2 then a final partition $\gamma = \{\{x_1\}, \{x_2\}, \{x_3, \ldots, x_n\}\}$ is enough for the searcher to find the target without further refinements). Note that, without loss of generality, the *admissibility property* (8) enables the searcher to use a fixed "optimistic" distance estimation $\tilde{d}(\alpha, \gamma) = 0$ between a candidate partition $\alpha \in \chi$ (that represent a candidate group test for the next search step) and the final partition γ (as well as between any pair of partitions in the graph). This distance estimation is considered "optimistic," since it represents an *a priori* assumption of the searcher that the candidate partition $\alpha \in \chi$ will result in finding the target. Using such an estimation is conservative with respect to the search time: it prolongs the converging time of the ILRTA* algorithm, as shown later, since the distance estimations between any two partitions, including the final partition γ, can be set to zero at the beginning of the search and are gradually refined during the search until they converge to their real distance values. The refinement of the distances is based on the fact that in the neighborhood of a partition the distance estimates are accurate.

Neighborhood of states is a key concept in SLS algorithms and, thus, also in the ILRTA* algorithm: the distance estimation is always accurate between neighbor states. The neighbors in the suggested ILRTA* algorithm are defined by the proposed informational distances between the partitions. Practically, it means that when the searcher reaches a certain (current) partition, for example, $\gamma = \{\{x_1, x_2\}, \{x_3, \ldots, x_n\}\}$, there is a subset of neighborhood partitions that often corresponds to a refinement of this partition; for example, $\gamma = \{\{x_1\}, \{x_2\}, \{x_3, \ldots, x_n\}\}$, with an accurate distance measure from the current partition that can be reached by a certain group test. If the searcher selects to move to this partition (vertex in the graph) a new subset of neighborhood partitions is available; e.g., $\gamma = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4, \ldots, x_n\}\}$, and the process of learning continues until the target is found. Moreover, say that at a certain point in time the searcher is exposed to new (external) side information that changes the location probabilities (for example, in the nonconforming unit example the new information indicates that it is more reasonable to believe that the nonconforming unit is located toward the end of the batch) then this information is instantaneously absorbed in the new location probabilities (e.g., using a Bayesian update scheme) and transformed to new distance measures that are used by the searcher to select the next group test. Note that this notion of online local learning updates and convergence is very different from offline search procedures that assume that all of the information has to be taken into account at the beginning of the search to define *a priori* all the search steps.

Formally, for any partition $\alpha \in \chi$, the non-empty partitions set $N(\alpha) \subset \chi$ is the *neighborhood* of the partition α if set $N(\alpha)$ meets the following requirements.

1. $\alpha \notin N(\alpha)$.
2. For every partition $\xi \in N(\alpha)$ it follows that $\tilde{d}(\xi, \alpha) = d(\xi, \alpha)$.

The first requirement is used to avoid staying at the current partition, while the second requirement specifies an "optimistic" assumption, which indicates that the searcher knows the exact distances at least to the neighboring partitions. Note that if for some initial steps of the ILRTA*, assumption 2 is violated, then the searcher can assume that the distance to all partitions is less than the entropy of the sample space. That means that the searcher starts with a neighborhood that includes all possible partitions and then, after few iterations of estimation updates, the above assumption becomes valid for smaller neighborhoods.

Based on the informational distances and the above definitions, the suggested ILRTA* algorithm is outlined as follows. Let θ be an arbitrary initial partition, γ be the final partition, and $\tilde{d}_0(\alpha, \gamma)$, $\alpha \in \chi$, be the initial distance estimations. Assume that the distance estimations are specified by zeros, or a certain method of their calculation preserving admissibility requirement (8)—e.g., the Ornstein distance formula (7)—is given. Notice that the values $\tilde{d}_0(\alpha, \gamma)$ denote the estimations to the final partition, while the values $\tilde{d}(\xi, \alpha)$, as indicated in the above neighborhood requirement (2), denote the estimations between the neighboring partitions. Then, the ILRTA* algorithm follows the procedure below.

Algorithm ILRTA*.

Given $\theta, \gamma, \alpha \in \chi$, and $\tilde{d}_0(\alpha, \gamma)$.

1. Init distance estimations by $\tilde{d}_0(\alpha, \gamma)$: $\tilde{d}(\alpha, \gamma) \leftarrow \tilde{d}_0(\alpha, \gamma)$, $\alpha \in \chi$.
2. Init current partition α_{cur} by initial partition $\theta : \alpha_{cur} \leftarrow \theta$.
3. While $d(\alpha_{cur}, \ \gamma) \neq 0$ do
   a. Choose next partition α_{next}:
   $$\alpha_{next} \leftarrow \arg \min_{\alpha \in N(\alpha_{cur})} \{d(\alpha_{cur}, \ \alpha) + \tilde{d}(\alpha, \ \gamma)\}.$$
   b. Set distance estimation for current partition α_{cur}:
   $$\tilde{d}(\alpha_{cur}, \gamma) \leftarrow \max \left\{ \begin{array}{l} \tilde{d}(\alpha_{cur}, \gamma), \\ \min\limits_{\alpha \in N(\alpha_{cur})} \{d(\alpha_{cur}, \alpha) + \tilde{d}(\alpha, \gamma)\} \end{array} \right\}.$$
   c. Move to next partition:
   $$\alpha_{cur} \leftarrow \alpha_{next}.$$

End while.

The execution of the ILRTA* algorithm is illustrated in Fig. 1. In the figure, the triangles stand for the searcher's partitions, the star sign stands for the target's location, as reflected by the final partition, and the circles denote some arbitrary partitions in the space χ that can be chosen by the searcher for inspection. Note that if the *neighborhood* set is limited to few (e.g., 12) candidate points, the recursive implementation of the ILRTA* (Ben-Gal *et al*., 2007; Ben-Gal *et al*., 2008) can handle large sample space of thousands of points.

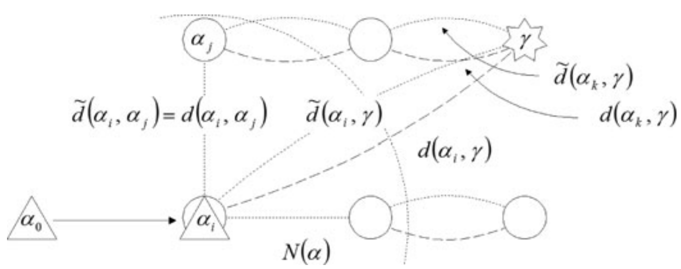


Fig. 1. Execution of the ILRTA* algorithm.

The figure illustrates a case where the searcher's location is represented by partition α_i. The neighborhood $N(\alpha)$ is represented by the dashed curve; thus, the distances to all partitions within the neighborhood, such as partition α_j, are known exactly and are equal to their estimates. The distances from α_i to the target's partition are estimated by $\tilde{d}(\alpha_i, \gamma)$, which is not necessarily equal to the accurate distance $d(\alpha_i, \gamma)$.

Notice that, as indicated above, it is assumed that if final partition γ is known, then the searcher calculates distance estimations using this partition. If, in contrast, such a partition is unknown, then the searcher, for example, can specify it as a discrete partition $\gamma = \{\{x_1\}, \{x_2\}, \ldots, \{x_n\}\}$ and calculate distance estimations to this partition. Any side information that changes the location probability will result in new estimated distances that will affect the searcher's selection.

Now let us show that for the ILRTA* algorithm the basic properties of the LRTA* algorithm (Korf, 1990) hold. Below new proofs of these properties are provided based on the proposed informational metrics. These properties provide an intuition to the implementation of the ILRTA* algorithm to classification problems (Ben-Gal *et al.*, 2007; Ben-Gal *et al.*, 2008) that are used for automated fault identification, as well as to studies of a moving target search (Kagan and Ben-Gal, 2006, 2007) that are relevant for quality inspection of dynamic faults. We proceed with the main properties of the ILRTA* algorithm.

Theorem 3. *If for every partition $\alpha \in \chi$ the admissibility assumption $\tilde{d}_0(\alpha, \gamma) \le d(\alpha, \gamma)$ holds, then the trial of the ILRTA* algorithm always terminates.*

Theorem 3 states that at the end of step 3 of the ILRTA* algorithm, when the trials are completed, the target is found. The proof of the theorem is based on the next two lemmas. Lemma 3 guarantees that the algorithm selects the final partition while it is in the neighborhood of the current partition. Lemma 4 states that the algorithm does not return to the previous partition at its next step. The complete proof of this theorem is given in the Appendix.

Lemma 3. *In the ILRTA* algorithm, if $\gamma \in N(\alpha)$ then the partition selection is $\alpha \leftarrow \gamma$ and the estimation update is $\tilde{d}(\alpha, \gamma) \leftarrow d(\alpha, \gamma)$.*

The proof of this lemma is given in the Appendix.

Lemma 4. *Let the current partition be $\alpha_{\text{cur}} = \beta$ and its previous partition be α. If there exists a partition $\omega \in N(\beta)$, $\omega \ne \alpha$, then in the next step, the ILRTA* algorithm will select ω.*

The proof of this lemma is given in the Appendix.

Let the algorithm be applied to certain input data $\theta, \gamma, \alpha \in \chi$, and $\tilde{d}_0(\alpha, \gamma)$, and it implements definite methods for calculating distances and distance estimations. Then, during its execution, the algorithm changes the distance estimations as specified in Step 3*b*. Such an execution of the algorithm over given data is called a *trial*. The next lemma guarantees that the ILRTA* algorithm preserves the admissibility property (8) of distances d and distance estimations $\tilde{d}$.

Lemma 5. *Throughout the trials of the ILRTA* algorithm, the admissibility property $\tilde{d}(\alpha, \gamma) \le d(\alpha, \gamma)$ holds for every partition $\alpha \in \chi$.*

The proof of this lemma is given in the Appendix.

Finally, assume that after the trial the obtained distance estimations are saved and are used as initial distance estimations for the next trial. Let us show that the suggested ILRTA* algorithm obtains an optimal solution in the sense of informational distances between partitions in the partitions space χ.

Theorem 4. *Solutions of the iterated trials of the ILRTA* algorithm converge to the sequence of partitions having a path length $d(\theta, \gamma)$, where $\tilde{d}(\alpha, \gamma) = d(\alpha, \gamma)$ for all $\alpha \in \chi$.*

Thus, the proposed routine is to repeatedly execute the ILRTA* algorithm, while keeping the updated distance results in the shortest path from the starting partition of the searcher to the target partition. The proof of this theorem is given in the Appendix.

The above formulated statements show that the suggested ILRTA* algorithm maintains similar properties to the LRTA* algorithm (Korf, 1990). Moreover, a determination of different probability measures on the sample space X allows us to apply the suggested ILRTA* algorithm for various tasks that are different from the considered search problem. One such example is the use of the ILRTA* principles for the construction of classification trees (Ben-Gal *et al.*, 2007; Ben-Gal *et al.*, 2008). Note that any level in the tree corresponds to a partition of the classified items; thus, the same distance measures can be applied to find the best partitioning variable. Further discussion on classification trees is available in the next section, particularly when comparing the ILRTA* to the GOTA, which is a generalized version of the Id3 algorithm (Quinlan, 1993).

4.1. *The ILRTA* and other information-theoretic search algorithms*

In the previous section, it was shown that the proposed ILRTA* algorithm converges with trials to the optimal solution. In this section, we show how the ILRTA*, under certain conditions, produces optimal or near-optimal search solutions in a single trial—solutions that are identical to those obtained by known information-theoretic search algorithms, such as the offline Huffman and the online GOTA search procedures. These entropy-based algorithms have been extensively used for group testing applications using nonconforming units trees (Herer and Raz, 2000; Ben-Gal, 2004).

4.1.1. *The ILRTA* algorithm and the optimal Huffman search*

We start by comparing the ILRTA* algorithm with the optimal Huffman search procedure (Zimmerman, 1959). The Huffman search method is based on the Huffman coding procedure (Huffman, 1952; Cover and Thomas, 1991) and is based on an offline construction stage. In this stage, a Huffman search tree is constructed. Later, the search is conducted according to the constructed tree. Below we show that under the requirements of the Huffman procedure the suggested ILRTA* algorithm results in the Huffman search tree.

Let α_l be a partition corresponding to the leaves of the Huffman tree; thus, l stands for the level of the deepest branch in the tree. The level of the root is zero, and γ is the final partition defined by the tree's leaves. Denote $A'_{\min} = \arg\min_{A\in\gamma} p(A)$ and $A''_{\min} = \arg\min_{A\in\gamma\setminus A'_{\min}} p(A)$, where, as above, $p(A) = \sum_{x\in A} p(x)$ and $p(X) = \sum_{x\in X} p(x)$. Then, according to the Huffman procedure (Huffman, 1952; Cover and Thomas, 1991), partition α_{l-j-1}, which corresponds to the $(l-j-1)$th level of the tree, $j = 0, 1, \ldots, l-1$, is the following:

$$\alpha_{l-j-1} = \left\{A'_{\min} \cup A''_{\min}, A_k\right\},$$

where $A_k \in \alpha_{l-j}$, $A_k \neq A'_{\min}$, $A_k \neq A''_{\min}$, $k = 1, 2, \ldots, |\alpha_{l-j}| - 1$.

We say that partitions α and β are *Huffman neighbors*, if one is a refinement of the other; i.e., if

$$\alpha = \{B_i \cup B_j, B_1, B_2, \ldots, B_{i-1}, B_{i+1}, \ldots, B_{j-1}, B_{j+1}, \ldots, B_n\},$$

or

$$\beta = \{A_i \cup A_j, A_1, A_2, \ldots, A_{i-1}, A_{i+1}, \ldots, A_{j-1}, A_{j+1}, \ldots, A_m\},$$

where $A_i \in \alpha$ and $B_i \in \beta$. Such a neighborhood is denoted by $N_{\text{Huf}}(\cdot)$. Indeed, if $\alpha \in N_{\text{Huf}}(\beta)$ then $\alpha \prec \beta$. As noted above, we assume that $\alpha \notin N_{\text{Huf}}(\alpha)$, $\alpha \in \chi$, and that if $\alpha \in N_{\text{Huf}}(\beta)$, then $\tilde{d}(\alpha, \beta) = d(\alpha, \beta)$.

By the use of the Huffman neighbors one can demonstrate that the selection of the next partition by the Huffman algorithm is determined by the conditional entropy. The formal decision-making rule that is used is given in the next lemma.

Lemma 6. *According to the Huffman procedure, given α_l and $\alpha_{\text{cur}} \prec \alpha_l$, the next partition α_{next} among all α values that are Huffman neighbors of α_{cur} is chosen as follows:*

$$\alpha_{\text{next}} = \arg\max_{\alpha\in N_{\text{Huf}}(\alpha_{\text{cur}})}\{H(\alpha_l \mid \alpha_{\text{cur}}) - H(\alpha_l \mid \alpha)\},$$

where $H(\alpha \mid \beta)$ stands for the conditional entropy of partition α given partition β.

The proof of this lemma is given in the Appendix.

Lemma 6 provides the basic formalization for implementing the Huffman procedure over the partitions space, as follows:

Huffman procedure.

Given α_0, α_l, χ,

1. Init current partition α_{cur} by α_l.
2. While $H(\alpha_{\text{cur}} \mid \alpha_0) \neq 0$ do
   a. Choose next partition α_{next}:

$$\alpha_{\text{next}} \leftarrow \arg\max_{\alpha\in N_{\text{Huf}}(\alpha_{\text{cur}})}\{H(\alpha_l \mid \alpha_{\text{cur}}) - H(\alpha_l \mid \alpha)\},$$

   or, since $\alpha \in N_{\text{Huf}}(\alpha_{\text{cur}})$, as

$$\alpha_{\text{next}} \leftarrow \arg\max_{\alpha\in N_{\text{Huf}}(\alpha_{\text{cur}})}\{H(\alpha)\}.$$

   b. Set partitions set as:

$$\chi \leftarrow \chi\setminus\{\alpha_{\text{next}}\}.$$

   c. Move to next partition:

$$\alpha_{\text{cur}} \leftarrow \alpha_{\text{next}}.$$

   End while.

We can now draw the relation between the ILRTA* algorithm and the Huffman procedure. Let $\alpha, \beta \in \chi$ be two partitions of the sample space X, θ be an initial partition, and γ be the final partition. Then, based on the above procedure, the following theorem can be stated.

Theorem 5. *Let $\tilde{d}(\alpha, \beta) = 0$ if $\alpha \notin N_{\text{Huf}}(\beta)$ and $\beta \notin N_{\text{Huf}}(\alpha)$. Then, if $\gamma \prec \theta$, the search tree constructed by the ILRTA* algorithm is equivalent to the Huffman tree constructed by the Huffman procedure.*

The proof of this theorem is given in the Appendix.

Theorem 5 states that the ILRTA* algorithm under the Huffman conditions generates the same search tree as the

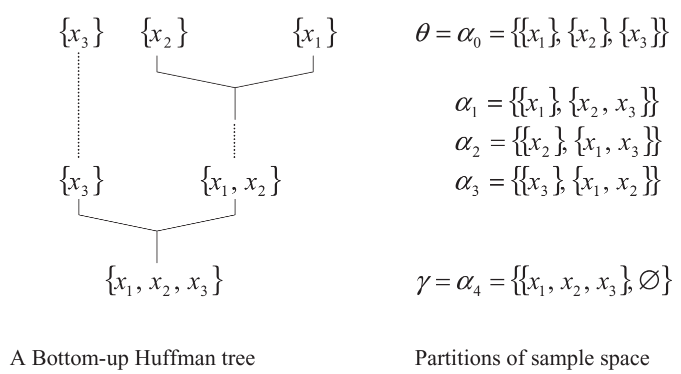


Fig. 2. Bottom-up Huffman tree and partitions of the sample space.

Huffman procedure. Thus, under these conditions the suggested ILRTA* algorithm is optimal in the sense of a minimal average number of search steps as the Huffman procedure.

The relation between the ILRTA* algorithm and the Huffman procedure can be illustrated by the following small example. Let $X = \{x_1, x_2, x_3\}$ be a sample space with location probabilities $p(x_1) = 0.1$, $p(x_2) = 0.3$, and $p(x_3) = 0.6$. The partition space χ of the sample space X consists of the following partitions:

$$\alpha_0 = \{\{x_1\}, \{x_2\}, \{x_3\}\},\ \alpha_1 = \{\{x_1\}, \{x_2, x_3\}\},$$
$$\alpha_2 = \{\{x_2\}, \{x_1, x_3\}\},\ \alpha_3 = \{\{x_3\}, \{x_1, x_2\}\},$$
$$\alpha_4 = \{\{x_1, x_2, x_3\}, \emptyset\}.$$

Recall that the Huffman algorithm creates a decision tree in a bottom-up procedure; i.e., the initial partition θ for this procedure is $\theta = \alpha_0 = \{\{x_1\}, \{x_2\}, \{x_3\}\}$ and the final partition γ is $\gamma = \alpha_4 = \{\{x_1, x_2, x_3\}, \emptyset\}$. As indicated above, a choice of final partition $\gamma = \{X, \emptyset\}$ is associated with the test of a full sample space X.

Given the above location probabilities and the initial and final partitions, the Huffman tree has a form shown on the left-hand side of Fig. 2. On the right-hand side of the figure we present the possible partitions of the sample space X that correspond to the relevant level of the tree.

In Fig. 2 it is seen that the initial partition $\theta = \alpha_0 = \{\{x_1\}, \{x_2\}, \{x_3\}\}$ has three possible Huffman neighbors $\alpha_1 = \{\{x_1\}, \{x_2, x_3\}\}$, $\alpha_2 = \{\{x_2\}, \{x_1, x_3\}\}$, and $\alpha_3 = \{\{x_3\}, \{x_1, x_2\}\}$. From these partitions, the Huffman procedure selects partition $\alpha_3 = \{\{x_3\}, \{x_1, x_2\}\}$. Thus, the first test to apply is either to look for the nonconforming unit at x_3 or to conduct a group test for nonconformity at $\{x_1, x_2\}$.

Now let us consider the actions of the ILRTA* algorithm on the same partitions space $\chi = \{\theta = \alpha_0, \alpha_1, \alpha_2, \alpha_3, \gamma = \alpha_4\}$ with the defined Rokhlin distance between the partitions. The graph that represents the partition space χ and the matrix of the Rokhlin distances among the partitions are shown in Fig. 3. Note that the ILRTA* algorithm in this case starts from the maximal refined partition among all possible partitions of the sample space and it ends in the less refined partition.

As shown in Fig. 3, a minimal non-zero Rokhlin distance d between the initial partition $\theta = \alpha_0$ and the other partitions α_1, α_2, and α_3 is reached with partition α_3, which is selected by the ILRTA* algorithm. The final partition $\gamma = \alpha_4$ is then selected since it is the only available partition in the Huffman neighborhood α_3 and it gives the minimum distance over all non-zero Rohklin distances. Thus, as expected, both the Huffman and the ILRTA* procedures result in the same sequence of partitions for group testing.

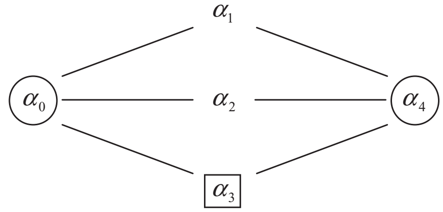
In the next section we consider the relation between the ILRTA* algorithm and the top-down GOTA, where all of the points are grouped together in the same subset.

4.1.2. *The ILRTA* algorithm and the GOTA*

We start by describing the GOTA (Hartmann *et al.*, 1982) in terms of its actions over the partitions set. Let us note that the GOTA is one of the original top-down search procedures and it has been shown to converge to the optimal solution with respect to the minimum number of search steps. It utilizes an entropy-based criterion that was later applied in a similar fashion to many popular data mining search algorithms, such as C.45 and Id3 (Quinlan, 1993).

Again, let X be a sample space given the location probabilities, and let χ be a set of all partitions of X. In addition, let $g : \chi \times \chi \to \boldsymbol{R}^+$ be some cost function, which indicates the cost $g(\alpha, \beta) > 0$ of the step from partition α to partition β.

The GOTA, as a top-down procedure, starts from the initial trivial partition $\theta = \{X, \emptyset\}$ and proceeds to the final partition $\gamma \succ \theta$. Thus, in terms of refinement relations, in the GOTA the relations between the partitions are opposite to the relations obtained by the Huffman procedure. Suppose that each test of the GOTA has a binary



d	α_0	α_1	α_2	α_3	α_4
α_0	0	0.83	0.41	0.32	1.30
α_1		0	1.24	1.15	0.47
α_2			0	0.32	0.88
α_3				0	0.97
α_4					0

Fig. 3. Implementation of the ILRTA* algorithm on the partitions space.

outcome. If α_{cur} is the current partition, then for the next chosen partition $\alpha_{\mathrm{next}} \in \chi$ it holds true that $\theta \prec \alpha_{\mathrm{cur}} \prec \alpha_{\mathrm{next}} \prec \gamma$ and $|\alpha_{\mathrm{next}}| = |\alpha_{\mathrm{cur}}| + 1$.

In the GOTA, for a given current partition α_{cur}, the next partition α_{next} is selected by the following criterion:

$$\arg\max_{\alpha \in \chi} \left(H(\gamma|\alpha_{\mathrm{cur}}) - H(\gamma|\alpha)/g(\alpha_{\mathrm{cur}},\ \alpha) \right),$$

over such partitions $\alpha \in \chi$ that $\alpha \succ \alpha_{\mathrm{cur}}$ and $|\alpha| = |\alpha_{\mathrm{cur}}| + 1$, and the GOTA terminates when the selected partition satisfies $\alpha_{\mathrm{cur}} \succ \gamma$.

Assume that all possible tests that partition the sample space are available. We say that partition β is a GOTA-*neighbor* of partition α if $\beta \succ \alpha$ and $|\beta| = |\alpha| + 1$. In other words, if $\beta = \{B_1, B_2, \ldots, B_n\}$ and it is a neighbor of α, then

$$\alpha = \{B_i \cup B_j, B_1, B_2, \ldots, B_{i-1}, B_{i+1}, \ldots, B_{j-1}, B_{j+1}, \ldots, B_n\},$$

where $B_i \in \beta$. Such a neighborhood is denoted by $N_{\mathrm{G}}(\cdot)$. Using these terms, the GOTA can be formalized over the set of partitions as follows.

Algorithm GOTA.

Given α_0, γ, χ,

1. Init current partition α_{cur} by α_0.
2. While $H(\gamma \mid \alpha_{\mathrm{cur}}) \neq 0$ do
   a. Choose next partition α_{next}:
   $$\alpha_{\mathrm{next}} \leftarrow \arg \max_{\alpha \in N_{\mathrm{G}}(\alpha_{\mathrm{cur}})} \left\{ \frac{H(\gamma \mid \alpha_{\mathrm{cur}}) - H(\gamma \mid \alpha)}{g(\alpha_{\mathrm{cur}},\ \alpha)} \right\}.$$
   b. Set partitions set as:
   $$\chi \leftarrow \chi \backslash \{\alpha_{\mathrm{next}}\}.$$
   c. Move to next partition:
   $$\alpha_{\mathrm{cur}} \leftarrow \alpha_{\mathrm{next}}.$$
   End while.

In the original GOTA (Hartman *et al.*, 1982), the cost function $g(\alpha, \beta) = \sum_{A \in (\beta \backslash \alpha)} p(A)$ is applied. If, instead, for every pair of partitions $(\alpha, \beta) \in \chi \times \chi, \alpha \neq \beta$, it is assumed that $g(\alpha, \beta) = 1$, then the selection by the GOTA is equivalent to the selection by the Huffman procedure. Note, however, that the Huffman procedure and the GOTA use different neighborhoods and different termination conditions that represent the different directions of the decision tree: bottom-up by the Huffman procedure and top-down by the GOTA.

Denote by $s = \langle \theta = \alpha_0, \alpha_1, \ldots, \alpha_n \succ \gamma \rangle$ a sequence of the partitions such that $\alpha_{i-1} \prec \alpha_i$ and $|\alpha_i| = |\alpha_{i-1}| + 1$, $i = 1, 2, \ldots, n$, and let S be a set of all available sequences s. For the GOTA, the following theorem is proven (Hartman *et al.*, 1982).

Theorem 6 (Hartman et al., 1982). *If for the cost function g it is true that $g(\alpha, \beta) \leq g(\alpha, \xi) + g(\xi, \beta)$, $\alpha, \beta, \xi \in \chi$, then the GOTA selects the sequence s with the minimal sum of costs $G = \sum_{i=1}^{n} g(\alpha_{i-1}, \alpha_i)$.*

In other words, the theorem states that the GOTA is near optimal in the sense of the given cost function.

Let us now clarify the relation between the ILRTA* algorithm and the GOTA, while assuming that the GOTA's requirements for the partition sequences and for the cost function g are valid. Assume that the selection of the next partition in step 3*a* of the ILRTA* algorithm is weighted by a non-zero cost $g(\alpha_{\mathrm{curr}}, \alpha)$ such that

$$\alpha_{\mathrm{next}} \leftarrow \arg \min_{\alpha \in N(\alpha_{\mathrm{cur}})} \left\{ \frac{1}{g(\alpha_{\mathrm{cur}},\ \alpha)} \left(d(\alpha_{\mathrm{cur}},\ \alpha) + \tilde{d}(\alpha,\ \gamma) \right) \right\}.$$

Then the following statement holds true.

Theorem 7. *If $\tilde{d}(\alpha, \beta) = 0$ for $\alpha \notin N_{\mathrm{G}}(\beta)$, $\beta \notin N_{\mathrm{G}}(\alpha)$ and if the ILTRA* and the GOTA have the same cost function g, for which $g(\alpha, \beta) = -g(\beta, \alpha)$, $\alpha, \beta \in \chi$, then the ILTRA* and the GOTA procedures result in the same search plan.*

The proof of this theorem is given in the Appendix.

From the near-optimality of the GOTA that is stated by Theorem 6, and from the equivalence between the ILRTA* algorithm and the GOTA, it follows that under the requirements of Theorem 7, the ILRTA* algorithm is near optimal in the same sense as the GOTA.

To illustrate the execution of the ILRTA* under the GOTA conditions, let us consider the following example.

Let $X = \{x_1, x_2, x_3, x_4\}$ be a sample space with the location probabilities $p(x_1) = 0.1$, $p(x_2) = 0.2$, $p(x_3) = 0.3$, and $p(x_4) = 0.4$. The partition space χ of the sample space X consists of the following partitions:

$\alpha_0 = \{\{x_1, x_2, x_3, x_4\}, \emptyset\}$,
$\alpha_1 = \{\{x_1\}, \{x_2, x_3, x_4\}\}$, $\alpha_2 = \{\{x_2\}, \{x_1, x_3, x_4\}\}$,
$\alpha_3 = \{\{x_3\}, \{x_1, x_2, x_4\}\}$, $\alpha_4 = \{\{x_4\}, \{x_1, x_2, x_3\}\}$,
$\alpha_5 = \{\{x_1, x_2\}, \{x_3, x_4\}\}$, $\alpha_6 = \{\{x_1, x_3\}, \{x_2, x_4\}\}$,
$\alpha_7 = \{\{x_1, x_4\}, \{x_2, x_3\}\}$, $\alpha_8 = \{\{x_1\}, \{x_2, x_3\}, \{x_4\}\}$,
$\alpha_9 = \{\{x_1\}, \{x_2\}, \{x_3, x_4\}\}$, $\alpha_{10} = \{\{x_1\}, \{x_3\}, \{x_2, x_4\}\}$,
$\alpha_{11} = \{\{x_2\}, \{x_1, x_3\}, \{x_4\}\}$, $\alpha_{12} = \{\{x_2\}, \{x_1, x_4\}, \{x_3\}\}$
$\alpha_{13} = \{\{x_3\}, \{x_1, x_2\}, \{x_4\}\}$, $\alpha_{14} = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}\}$.

Suppose that the cost function g is conventionally defined as $g(\alpha, \beta) = \sum_{A \in (\beta \backslash \alpha)} p(A)$. The matrix of the Rokhlin distances with costs $g(\alpha, \beta)$ is shown in Fig. 4.

The corresponding graph that represents partition space χ is shown in Fig. 5.

Following Figs. 4 and 5, a minimum non-zero Rokhlin distance d, as weighted by the GOTA cost between the initial partition $\theta = \alpha_0$ and the partitions α_i, $i = 1, \ldots, 7$, is reached for partition α_1, which is chosen by the ILRTA* algorithm. The minimum non-zero distance between the obtained partition α_1 and partitions α_i, $i = 8, \ldots, 13$, is

d	α_0	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	α_{11}	α_{12}	α_{13}	α_{14}
α_0	0	0.47	0.72	0.88	0.97	0.88	0.97	1.0	0	0	0	0	0	0	0
α_1		0	0	0	0	0	0	0	0.99	0.76	0.92	1.05	1.02	1.10	0
α_2			0	0	0	0	0	0	0.64	0.54	0.57	1.0	0.95	0.85	0
α_3				0	0	0	0	0	0.48	0.28	0.59	0.64	0.86	0.99	0
α_4					0	0	0	0	0.65	0.19	0.32	0.92	0.51	1.0	0
α_5						0	0	0	0.48	0.92	0.41	0.64	0.60	0.99	0
α_6							0	0	0.39	0.19	0.81	0.92	0.51	0.60	0
α_7								0	0.72	0.16	0.30	0.52	0.97	0.57	0
α_8									0	0	0	0	0	0	0.97
α_9										0	0	0	0	0	0.99
α_{10}											0	0	0	0	0.92
α_{11}												0	0	0	0.81
α_{12}													0	0	0.72
α_{13}														0	0.92
α_{14}															0

Fig. 4. Matrix of the Rokhlin distances weighted by costs for the ILRTA* algorithm acting under GOTA requirements.

then reached for partition α_9. Following this selection, the single remaining partition that is chosen is the final partition $\gamma = \alpha_{14}$.

The above procedures and the obtained results show that under defined conditions the ILRTA* algorithm acts as known (offline optimal and online near-optimal) information-theoretic search procedures. In fact, it follows that the ILRTA* algorithm can be considered as a generalization of these known search algorithms, and it is reduced to these algorithms by suitable definitions of the neighborhood. To the best of our knowledge, such an observation has not been previously published.

5. Simulation studies using the ILRTA* algorithm

This section presents numerical simulations using the ILRTA* algorithm. In most of the reported simulations the Rokhlin metric was used as a measure of the distance between partitions; however, in one case the Ornstein distance was used as the distance estimation to represent a situation with side information. Roughly speaking, the Ornstein distance measures the difference between two partitions such that it provides a probability that these partitions are different. Using the Ornstein distance as a distance estimator between the candidate partition and the final partition provides a probability measure that the search process will terminate in the next step.

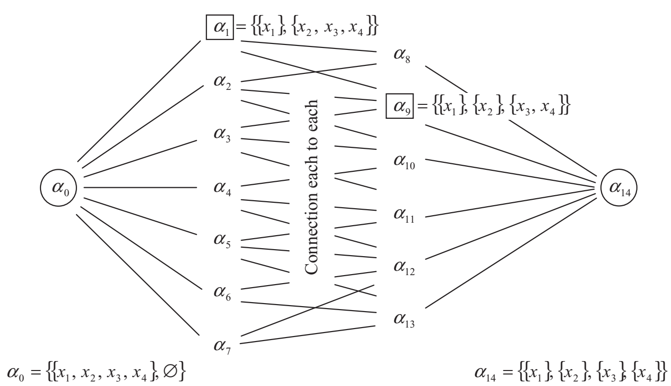


Fig. 5. Graph of partitions with a GOTA neighborhood for the ILRTA*.

Based on practical considerations, we assumed that at the beginning of the search a simple offline search procedure can be used, such as the Huffman search. The offline procedure results in a fast "zoom-in" process to a smaller search subspace, where the target can be located. Then, in the vicinity of the target the online ILRTA* procedure can be used to account for new side information. The use of an offline procedure is particularly appealing when *a priori* information on the target's location is updated gradually at an increasing rate: the information is relatively fixed at the beginning of the search, relatively far from the target; therefore an offline procedure is feasible, while most of the information updates are revealed as the searcher reaches the target's neighborhood, thus requiring the application of an online procedure. The justification for such an assumption can be found in many practical examples. For example, in software testing the exact location of a bug is refined once the faulty module has been identified. Similarly, in quality inspection of wafers, the information on the exact location of the contaminated spots is evident once the visual inspection equipment is focused on the contaminated area. The same principle applies to the location of a nonconforming unit in a batch (Herer and Raz, 2000), a search for a mobile device in a cellular network (Krishnamachari *et al.*, 2004),

Table 1. Simulation results of the ILRTA* algorithm in the vicinity of the target

Distance, estimation, neighborhood	*Number of steps*			
	Min	*Max*	*Mean*	*Std. deviation*
Rokhlin distance, zero estimation, Huffman neighborhood	1	8	5.575	1.866
Rokhlin distance, zero estimation, GOTA neighborhood	2	8	6.025	1.609
Rokhlin distance, zero estimation, general neighborhood	1	25	6.020	3.700
Rokhlin distance, Ornstein estimation, general neighborhood	1	14	2.790	1.864

and the identification of a bad segment in a communication network (Brice and Jiang, 2009). In all of these examples, most of the information updates are revealed as being at the target's neighborhood.

The use of an offline procedure at the beginning of the search is very appealing, since most of the offline group testing procedures (including the proposed ILRTA* with a Huffman neighborhood, as discussed in Section 4) can apply the entropy lower bound for the expected number of tests. This bound, in the worst case, is close to the entropy of the location probability plus one (Cover and Thomas, 1991; Ben-Gal, 2004). This fact implies that when the search space contains, for example, 10 000 points, it will require up to $\lceil \log_2 10\,000 \rceil = 14$ binary group tests or $\lceil \log_q 10\,000 \rceil$ q-ary group tests to reach the vicinity of the target. Taking these observations into account, in the simulated study we focused on the last stage of the search near the target, while ignoring the relatively fixed (and optimal) number of test rounds till this area is reached.

In the simulations we used a sample space X of 10 000 points. When comparing the different search algorithms we ignore the first fixed number of group tests that are required to reach the subset of nine points in the vicinity of the target. Each simulation contained 1000 replications of the search algorithm. The target's location was uniformly distributed over the sample space during the simulations. These replications were executed by a suitable C++ program. The pseudo-code of the main function of the simulation procedure is given at the end of the Appendix. In this implementation of the ILRTA* we build and store all partitions before the search, whereas in the recursive algorithm (Ben-Gal *et al*., 2007, Ben-Gal *et al*., 2008) the partitions are built consecutively during the search. Accordingly, at each step this implementation required polynomial time to build the partition and to select the searcher's next move.

The simulations executed different search procedures over the partition space with different types of neighborhood structures and two types of distance estimations. The different search conditions are represented by the different lines in Table 1: (i) a Huffman neighborhood with the Rokhlin distance and zero-distance estimations (i.e., similar to Huffman with no side information); (ii) a GOTA neighborhood with the Rokhlin distance and zero-distance estimations (i.e., similar to GOTA with no side information); (iii) a general neighborhood (defined by any possible refinement of the current partition) with a Rokhlin distance and zero-distance estimations (i.e., a general search with no side information); and (iv) a general neighborhood with the Rokhlin distance and the Ornstein distance estimations (i.e., general search with side information). In addition, for comparison purposes, a search by the ILRTA* algorithm versus the popular maximum entropy criterion, which results in a selection of the partition that obtains the maximal conditional entropy given the current partition, was also performed in line (v) in Table 1. Thus lines (i), (ii), and (v) in Table 1 can be considered as known methods that can be generalized by the ILRTA* algorithm. The results of the simulations are presented in Table 1.

We first analyzed the results for the cases that are not based on any side information (represented by the "zero estimation" cases). The histograms of the number of search steps for such cases are shown in Fig. 6.

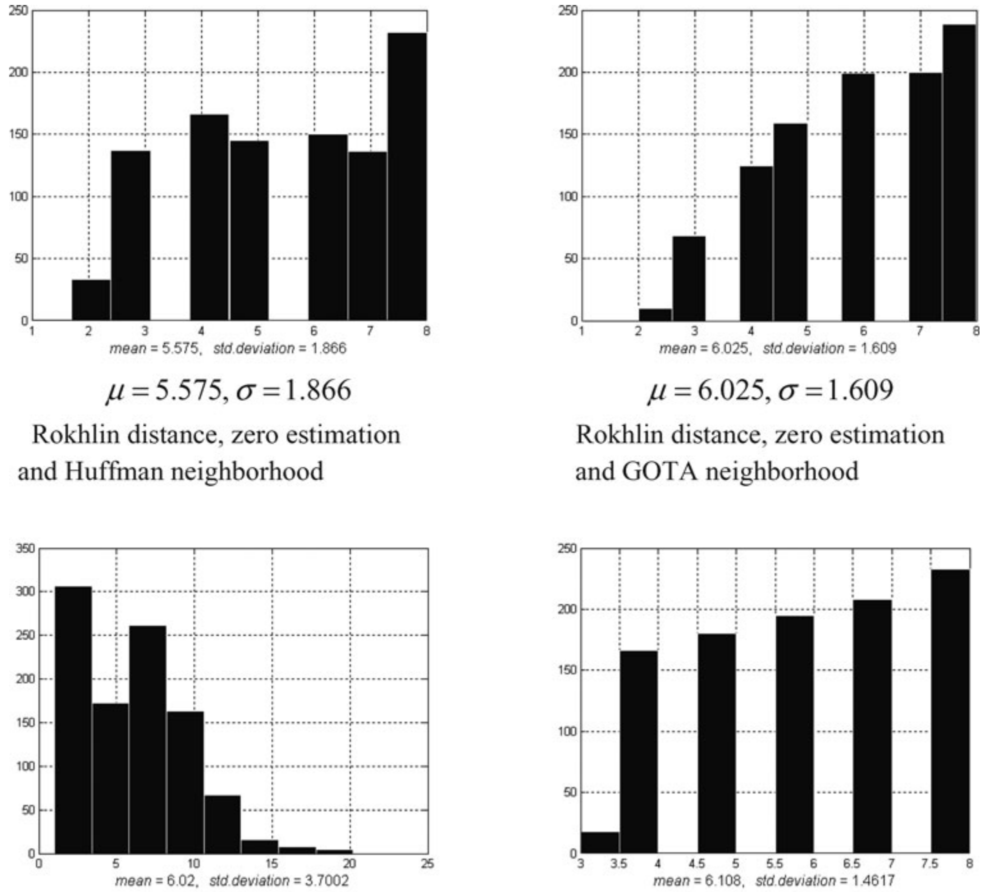
A comparison of the simulation results was performed by using Welch's t statistic with a significance level parameter of 0.05. Since the number of degrees of freedom for all tests was greater than 120, the statistic's critical value is approximately $t_{\text{stat}} = 1.960$. Results of the comparisons are presented in Table 2.

Note that the results obtained for the search using a Huffman neighborhood are statistically different from those obtained for searches using all the other neighborhood structures. This result confirms the known fact on the optimality of the Huffman procedure for a search after a static target when *a priori* information is fixed.

The next comparison was carried out between the online ILRTA* and the offline, computationally expensive Markov Decision Process (MDP) search process (White, 1993). The MDP procedure and its variants rely on an expectation regarding the target's locations at the next time steps, which determined the searcher's selections (Kagan and Ben-Gal, 2007).

We compared the ILRTA* algorithms using a Huffman neighborhood, a Rokhlin distance, and a zero estimation (with a mean of 5.575 and a standard deviation of 1.866) with an MDP model, which is based on the maximum probability criterion (with a mean of 4.928 and a standard deviation of 2.547). The histograms for these two search methods are shown in Fig. 7.

The observed value of the t-statistic in this comparison is $t_{\text{observed}} = 6.481$, and the statistical value is $t_{\text{stat}} = 1.960$.



$\mu = 5.575, \sigma = 1.866$
Rokhlin distance, zero estimation and Huffman neighborhood

$\mu = 6.025, \sigma = 1.609$
Rokhlin distance, zero estimation and GOTA neighborhood

$\mu = 6.020, \sigma = 3.700$
Rokhlin distance, zero estimation and general neighborhood

$\mu = 6.025, \sigma = 1.609$
Maximum probability criterion and general neighborhood

Fig. 6. Histograms for the number of search steps without distance estimations.

Thus, a Welch's t-test with a significance level of 0.05 shows that, in terms of a minimal expected number of search steps, the MDP search with an expectation is significantly better than the ILRTA* search. This result highlights that when the computational budget (and time) enables massive dynamic-programming type of computations, and when an expectation on the target location is available, the MDP approach outperforms the ILRTA*. This conclusion changes, as seen below, when side information is introduced during the search.

Finally, we compared the ILRTA* algorithm with non-zero distance estimations against the MDP model. In particular, we compared the search using the ILRTA* algorithm against the Rokhlin distance, an Ornstein estimation, and a general unrestricted neighborhood (with a mean of 2.790 and a standard deviation of 1.864) versus the MDP

Table 2. Statistical differences between the actions of the ILRTA* algorithm with different neighborhoods and estimations

Rokhlin distance, zero estimation	*Rokhlin distance, zero estimation*		*Maximum entropy criterion, general neighborhood*
	GOTA neighborhood	*General neighborhood*	
Huffman neighborhood	Significant, $t_{\text{observed}} = 5.776$	Significant, $t_{\text{observed}} = 3.396$	Significant, $t_{\text{observed}} = 7.111$
GOTA neighborhood	—	Insignificant, $t_{\text{observed}} = 0.039$	Insignificant, $t_{\text{observed}} = 1.207$
General neighborhood	—	—	Insignificant, $t_{\text{observed}} = 0.699$

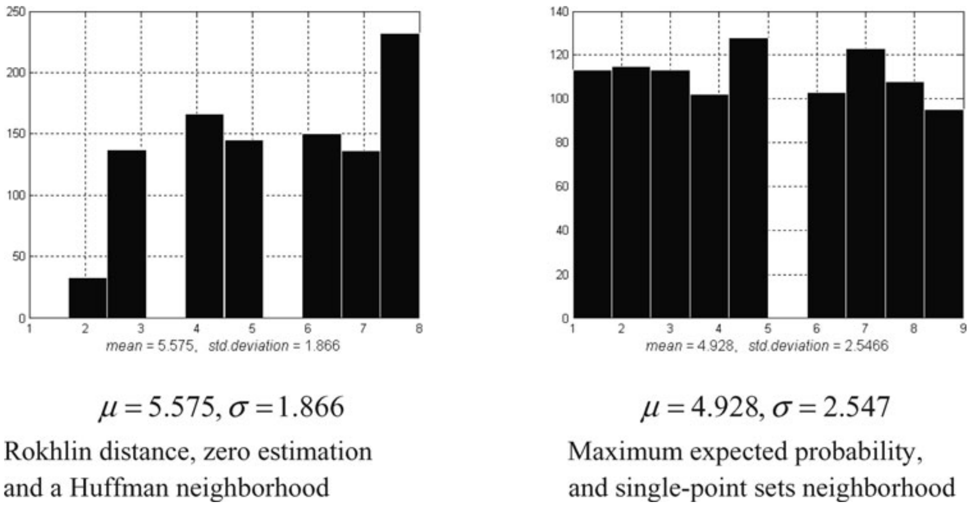


Fig. 7. Histograms for the number of search steps for the ILRTA* algorithm and the MDP model.

search with a maximum expected information criterion and multi-point subsets (with a mean of 4.459 and a standard deviation of 1.205). The histograms of the search steps for these two procedures are shown in Fig. 8.

The Welch's t-test indicates that the expected number of search steps obtained by the ILRTA* algorithm with side information is significantly smaller than the expected number of search steps obtained by the MDP model. The value of the t-statistic in this case is $t_{\text{observed}} = 23.777$, and for a significance level of 0.05 (with 1709 degrees of freedom) it is $t_{\text{stat}} = 1.960$.

The results of the simulations show that the group testing/search process depends significantly on the type of neighborhood, the stability of the available information, and the distance estimations used. The best results in the case of zero distance estimations are obtained for the Huffman neighborhood. An implementation of Ornstein distance estimations results in a significantly smaller number of search steps with respect to all the simulated cases, also for the cases with an unrestricted general neighborhood.

6. Discussion and summary

This article considers the problem of group testing/search for a static target over a discrete sample space; e.g,. for locating a nonconforming unit in a batch of units. The action available to the searcher is to check a subset of units and determine whether the target is included in this subset or not. The procedure terminates if the searcher finds the target in a subset that contains only one unit. The goal is to find a search procedure that terminates in a minimal average number of steps. For this purpose, the ILRTA* algorithm is

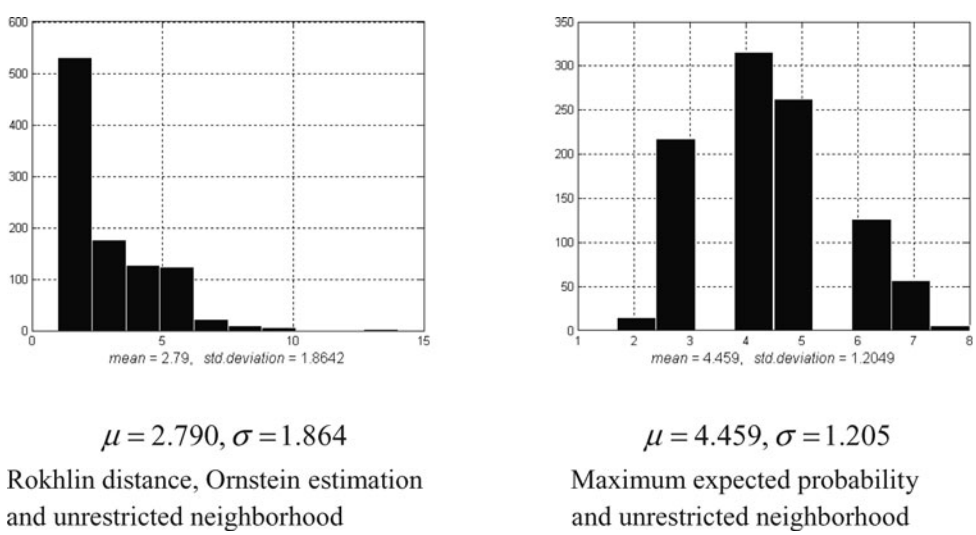


Fig. 8. Histograms for the number of search steps for the ILRTA* algorithm and MDP model for a search with expectations.

proposed as a solution approach of online group testing or sequential experimentation. The online search algorithm is based on an SLS framework. In contrast with known search methods, it allows taking into account side information that might be available at the neighborhood of the target. Examples for such side information exist in quality tests of wafers, printed circuit boards, software modules, chemical batches, and more.

The suggested ILRTA* algorithm, in contrast with other search methods, performs over the set of partitions of the sample space. It utilizes the Rokhlin metric and the Ornstein metric as distance measures among alternative partitions. However, in general, other distance measures that comply with the metric requirements can be used within the suggested framework. The required properties of the metrics are developed, and their dependence on the location probabilities and the space topology are studied.

It is shown that the suggested ILRTA* algorithm always converges and terminates. Moreover, the ILRTA* is compared against the known optimal Huffman search algorithm and the near-optimal GOTA. It is proven theoretically and demonstrated by numerical examples that under the conditions of the Huffman search the formulated ILRTA* algorithm results in the same optimal solutions, and under the conditions of GOTA, it results in near-optimal solutions that are equivalent to those obtained by the GOTA. In contrast with these known algorithms, the online ILRTA* algorithm can take into account side information regarding the target location that is advantageous in situations where new information is obtained during the search.

Features of the suggested algorithm are analyzed theoretically and studied by numerical simulations. Statistical analysis of simulated results shows that the suggested ILRTA* search algorithm outperforms known search procedures in various settings. Results of comparisons between the suggested ILRTA* against other known search procedures are summarized in Table 3. The table presents various search procedures that are applicable under various settings. Bold fonts mark those algorithms that provide the best results in terms of a minimal average number of search steps.

As it follows from Table 2, the proposed ILRTA* algorithm outperforms other known models in most cases, and in the presence of side information the ILRTA* algorithm achieves much superior results than the other considered algorithms.

Besides the trivial application of finding a nonconforming unit, the suggested algorithm can be applied to a number of practical problems, including, for example, classification of data sequences (Ben-Gal *et al*., 2007; Ben-Gal *et al*., 2008), where it leads to effective solutions. Finally, the ILRTA* algorithm can be generalized for a search after a moving target (Kagan and Been-Gal, 2006)—a problem that is equivalent to a search problem over a sample space with changing probability distributions or to a search by multiple searchers for multiple targets. These two generalizations are further considered in related papers (Kagan and Ben-Gal, 2006, 2010).

Table 3. Comparative study of the ILRTA* algorithm against other search algorithms

	Presence of side information	*Absence of side information*
Group tests with single-point sets		With off-line stages: **Ross's MDP**[2]
	On-line search: Probabilistic MDP Informational MDP **ILRTA* algorithm**[1]	On-line search: Probabilistic MDP Informational MDP **ILRTA* algorithm**[3]
Group tests with multi-points sets		With off-line stages: **Huffman**[5]
	On-line search: Informational MDP; **ILRTA* algorithm**[4]	On-line search: GOTA Informational MDP [6] **ILRTA* algorithm**[7]

[1]Requires non-zero distance estimations. The superiority of the ILRTA* algorithm is shown by numerical simulations.
[2]Requires offline dynamic programming solution. Optimality versus costs of search is proven analytically.
[3]Requires non-zero distance estimations. The superiority of the ILRTA* algorithm is shown by numerical simulations.
[4]The superiority of the ILRTA* algorithm is proven analytically and confirmed by numerical simulations.
[5]Requires offline creation of the search tree. The optimality is proven analytically.
[6]The informational MDP search and the GOTA search result in statistically equivalent numbers of search steps.
[7]The ILRTA* algorithm and by the Huffman procedure generate an equivalent search trees.

References

Abrahams, J. (1994) Parallelized Huffman and Hu–Tucker searching. *IEEE Transactions on Information Theory*, **40**(2), 508–510.

Ben-Gal, I. (2004) An upper bound on the weight-balanced testing procedure with multiple testers. *IIE Transactions*, **36**, 481–493.

Ben-Gal, I., Kagan, E. and Shkolnik, N. (2008) Constructing classification trees via data mining and design of experiments concepts, in *Proceedings of ENBIS 2008*. September 21–25, Athens, Greece.

Ben-Gal, I., Shkolnik, N. and Kagan, E. (2007) Greedy learning algorithms and their applications to decision trees, in *Proceedings of ENBIS 2007*. September 24–26, Dortmund, Germany.

Bertsekas, D.P. (1995) *Dynamic Programming and Optimal Control*, Athena Scientific Publishers, Boston, MA.

Brice, P. and Jiang, W. (2009) A context tree method for multistage fault detection and isolation with applications to commercial video broadcasting systems. *IIE Transactions on Quality and Reliability*, **41**(9), 776–789.

Bulitko, V. and Lee, G. (2006) Learning in real-time search: a unifying framework. *Journal of Artificial Intelligence Research*, **25**, 119–157.

Bulitko, V., Sturtevant, N., Lu, J. and Yau, T. (2007) Graph abstraction in real-time heuristic search. *Journal of Artificial Intelligence Research*, **30**, 51–100.

Cover, T.M. and Thomas, J.A. (1991) *Elements of Information Theory*, John Wiley & Sons, New York, NY.

Dorfman, R. (1943) The detection of defective members of large population. *Annals of Mathematical Statistics*, **14**, 436–440.
Graff, L.E. and Roeloffs, R. (1974) A group-testing procedure in the presence of test error. *Journal of the American Statistical Association*, **69**(345), 159–163.
Gupta, D. and Malina, R. (1999) Group testing in presence of classification errors. *Statistics in Medicine*, **18**, 1049–1068.
Hartmann, C.R.P., Varshney, P.K., Mehrotra, K.G. and Gerberich, C.L. (1982) Application of information theory to the construction of efficient decision trees. *IEEE Transactions on Information Theory*, **28**(4), 565–577.
Herer, Y.T. and Raz, T. (2000) Optimal parallel inspection for finding the first nonconforming unit in a batch—an information theoretic approach. *Management Science*, **46**(6), 845–857.
Huffman, D.A. (1952) A method of the construction of minimal-redundancy codes. *Proceedings of I.R.E.*, **40**, 1098–1101.
IRCL Research Group publications. http://sites.google.com/a/ualberta.ca/ircl/projects (accessed October 16, 2013).
Ishida, T. and Korf, R.E. (1991) Moving target search, in *Proceedings of IJCAI 1991*, pp. 204–210.
Ishida, T. and Korf, R.E. (1995) Moving target search: a real-time search for changing goals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(6), 609–619.
Jaroszewicz, S. (2003) Information-theoretical and combinatorial methods in data-mining. Ph.D. Thesis, University of Massachusetts, Boston, MA.
Kagan, E. and Ben-Gal, I. (2006) An informational search for a moving target, in *Proceedings of the IEEE 24th EEEI Convention*, pp. 153–155.
Kagan, E. and Ben-Gal, I. (2007) A MDP test for dynamic targets via informational measure, in *Proceedings of the IIERM 2007 Research Meeting*, pp. 39–53.
Kagan, E. and Ben-Gal, I. (2010) Search after a static target by multiple searchers by the use of informational distance measures, in *Proceedings of IE&M 2010*, p. 12.
Koenig, S. and Likhachev, M. (2006) Real-time adaptive A*, in *Proceedings of AAMAS 2006*, pp. 281–288.
Koenig, S. and Simmons, G. (1995) Real-time search in non-deterministic domains, in *Proceedings of IJCAI 1995*, pp. 1660–1667.
Korf, R.E. (1990) Real-time heuristic search. *Artificial Intelligence*, **42**(2–3), 189–211.
Krishnamachari, R., Gau, R.-H., Wicker, S.B. and Haas, Z.J. (2004). Optimal sequential paging in cellular networks. *Wireless Networks*, **10**, 121–131.
Lloris-Ruiz, A., Gomez-Lopera, J.F. and Roman-Roldan, R. (1993) Entropic minimization of multiple-valued logic functions, in *Proceedings of ISMVL 1993*, pp. 24–28.
López De Mántaras, R. (1991) A distance-based attribute selection measure for decision tree induction. *Machine Learning*, **6**, 81–92.
Ornstein, D.S. (1974) *Ergodic Theory, Randomness, and Dynamical Systems*, Yale University Press, New Haven, CT.
Peltonen, J. (2004) Data exploration with learning metrics. Ph.D. Thesis, Helsinki University of Technology, Finland.
Quinlan, J.R. (1993) *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San-Francisco, CA.
Rokhlin, V.A. (1967) Lectures on the entropy theory of measure-preserving transformations. *Russian Mathematical Surveys*, **22**, 1–52.
Ross, S.M. (1983) *Introduction to Stochastic Dynamic Programming*, Academic Press, New York, NY.
Shimbo, M. and Ishida, T. (2003) Controlling the learning process of real-time heuristic search. *Artificial Intelligence*, **146**, 1–41.
Sinai, Y.G. (1977) *Introduction to Ergodic Theory*, Princeton University Press, Princeton, NJ.
Sun, X., Yeoh, W. and Koenig, S. (2009) Efficient incremental search for moving target search, in *Proceedings of IJCAI 2009*, pp. 615–620.
Vitter, J.S. (1999) Online data structures in external memory. *Lecture Notes in Computer Science*, **1663**, 352–366.
White, D.J. (1993) *Markov Decision Processes*, John Wiley & Sons, Chichester, UK.
Yadin, M. and Zacks, S. (1988) Visibility probability on line segments in three-dimensional spaces subject to random Poisson fields of obscuring spheres. *Naval Research Logistics*, **35**, 555–569.
Yadin, M. and Zacks, S. (1990) Multiobserver multitarget visibility probabilities for Poisson shadowing processes in the plane. *Naval Research Logistics*, **37**, 603–615.
Zimmerman, S. (1959) An optimal search procedure. *American Mathematics Monthly*, **66**, 690–693.

Appendix

Below we present the proofs of the theorems and lemmas that were formulated in the main text.

Proof of Lemma 1. Using the definition of the conditional entropy (2) and the formulas of relative probabilities, we obtain that

$$d(\alpha, \beta) = -\sum_{i=1}^{k}\sum_{j=1}^{k} p(A_i \cap B_j) \times \left(\log\frac{p(A_i \cap B_j)}{p(A_i)} + \log\frac{p(A_i \cap B_j)}{p(B_j)}\right).$$

Since the probabilistic measure is additive, we also have

$$d_{Orn}(\alpha, \beta) = 1 - \sum_{i=1}^{k} p\,(A_i \cap B_i).$$

Thus, we need to prove the correctness of the following inequality:

$$R = \sum_{i=1}^{k}\left(p(A_i \cap B_i) + \sum_{j=1}^{k} p(A_i \cap B_j) \times \log\frac{p(A_i)p(B_j)}{[p(A_i \cap B_j)]^2}\right) \geq 1.$$

If $\alpha = \beta$, then:

$$\sum_{i=1}^{k}\sum_{j=1}^{k} p\left(A_i \cap B_j\right)\log\frac{p(A_i)p(B_j)}{[p(A_i \cap B_j)]^2} = 0,$$

$$\text{and}\quad R = \sum_{i=1}^{k} p\,(A_i \cap B_i) = 1.$$

Let $\alpha \neq \beta$, and suppose that the partitions $\alpha = \{A_1, \dots, A_{k-1}, A_k\}$ and $\beta = \{B_1, \dots, B_{k-1}, B_k\}$ are such that $A_i = B_i$, $i = 1, 2, \dots, k-2$, and $A_{k-1} \neq B_{k-1}$ and $A_k \neq B_k$.

Let $\alpha' = \{A'_1, \dots, A'_{k-1}, A'_k\}$ and $\beta' = \{B'_1, \dots, B'_{k-1}, B'_k\}$ be such partitions that $A'_i = A_i$, $B'_i = B_i$; $i = 1, 2, \dots, k-2$; $A'_{k-1} = B'_{k-1}$, $A'_k = B'_k$; $p(A'_{k-1}) + p(A'_k) = p(A_{k-1}) + p(A_k)$; and $p(B'_{k-1}) + p(B'_k) = p(B_{k-1}) + p(B_k)$. Denote the left side of the inequality for the partitions $\alpha' = \beta'$ by R'. Indeed, $R' = 1$. Since $\alpha' = \beta'$, we have $p(A'_{k-1}) + p(A'_k) =$

$p(B'_{k-1}) + p(B'_k)$. Hence, $p(A_{k-1}) + p(A_k) = p(B_{k-1}) + p(B_k)$, and so $A_{k-1} \subset B_{k-1}$ and $B_k \subset A_k$ or $B_{k-1} \subset A_{k-1}$ and $A_k \subset B_k$.

Let for certainty $A_{k-1} \subset B_{k-1}$ and $B_k \subset A_k$. Consider the last addendums of R' and R. According to the assumptions for α' and β', $R' = m' + r'$ and $R = m + r$. Thus, we obtain

$$r' = p(A'_{k-1}) + p(A'_k) = p(A'_{k-1}) + p(B'_k),$$
$$r = p(A_{k-1}) + p(A_{k-1}) \log \frac{p(B_{k-1})}{p(A_{k-1})} + p(B_k) + p(B_k) \log \frac{p(A_k)}{p(B_k)}.$$

Since $A_{k-1} \subset B_{k-1}$ and $B_k \subset A_k$, it holds true that $p(A_{k-1}) < p(B_{k-1})$ and $p(A_k) < p(B_k)$. Thus, $\log(p(B_{k-1})/p(A_{k-1})) > 0$ and $\log(p(A_k)/p(B_k)) > 0$.

Now taking into account that (according to definition of the partitions α' and β') $m' = m$, we obtain $r' < r$, and so $R' < R$.

The use of the same reasoning to the remaining cases with $A'_i = B'_i$, $i = (k-2), (k-1), k$ and so on for $i = 1, 2, \ldots, k$ is similar. Thus, if $\alpha = \beta$ then $R = 1$, and if $\alpha \neq \beta$ then $R > 1$. ■

Proof of Lemma 2. The proof of the lemma is based on the following three claims that are to be proven first. Let $\theta = \{X, \emptyset\}$ be a trivial partition and let $\gamma = \{\{x_1\}, \{x_2\}, \ldots, \{x_n\}\}$, $x_i \in X$, $i = 1, \ldots, n$, be a discrete partition.

Claim 1. $d(\theta, \gamma) = \max_{(\alpha,\beta)\in\chi\times\chi} d(\alpha, \beta)$.

Proof. Assume that there exist two partitions $\alpha, \beta \in \chi$ such that $d(\alpha, \beta) > d(\theta, \gamma)$; that is,

$$H(\alpha \mid \beta) + H(\beta \mid \alpha) > H(\theta \mid \gamma) + H(\gamma \mid \theta).$$

Note that $H(\theta) = 0$, and since $\theta \prec \gamma$, it implies that $H(\theta \mid \gamma) = 0$. Moreover, partitions γ and $\xi \in \chi$ satisfy $H(\gamma \mid \xi) = H(\gamma) - H(\xi)$ and, in particular, $H(\gamma \mid \theta) = H(\gamma)$. Thus, $H(\alpha \mid \beta) + H(\beta \mid \alpha) > H(\gamma) - H(\theta) = H(\gamma)$ and $H(\alpha \mid \beta) + H(\beta \mid \alpha) > H(\gamma \mid \beta) + H(\beta)$. Hence,

$$H(\alpha \mid \beta) - H(\gamma \mid \beta) > H(\beta) - H(\beta \mid \alpha).$$

However, $H(\beta) \geq H(\beta \mid \alpha)$, and since $\beta \prec \gamma$ we obtain $H(\gamma \mid \beta) \geq H(\alpha \mid \beta)$. Thus, $H(\alpha \mid \beta) - H(\gamma \mid \beta) \leq 0$ and $H(\beta) - H(\beta \mid \alpha) \geq 0$, which contradict to the initial assumption. ■

Now we consider the Rokhlin metric for the above-mentioned probability mass functions u and p. Let $d_u(\alpha, \beta)$ denote the Rokhlin metric for the uniform (equi-probable) distribution of the target's locations and $d_p(\alpha, \beta)$ denote the Rokhlin metric for some arbitrary probability mass function with finite variance.

Claim 2. *For every partition $\alpha \in \chi$ it applies that $d_p(\alpha, \alpha) = d_u(\alpha, \alpha) = 0$ and that $d_p(\theta, \gamma) \leq d_u(\theta, \gamma)$.*

Proof. For every p one obtains

$$d_p(\theta, \gamma) = H_p(\theta \mid \gamma) + H_p(\gamma \mid \theta) = H_p(\gamma \mid \theta) = H_p(\gamma)$$

and it is known (Cover and Thomas, 1991) that $H_p(\gamma) \leq H_u(\gamma)$.

Let $\alpha, \beta \in (\chi, \prec)$. If $d(\theta, \alpha) = d(\theta, \beta)$ then $d(\alpha, \beta) = 0$. In fact, since it is a metric, if $\alpha = \beta$ then $d(\alpha, \beta) = 0$, regardless of p.

Let $\alpha \neq \beta$ and assume without loss of generality that $\alpha \prec \beta$. By definition, $d(\theta, \alpha) = H(\alpha \mid \theta) + H(\theta \mid \alpha) = H(\alpha)$, $d(\theta, \beta) = H(\beta \mid \theta) + H(\theta \mid \beta) = H(\beta)$, and $d(\alpha, \beta) = H(\alpha \mid \beta) + H(\beta \mid \alpha)$.

Since $\alpha \prec \beta$, one obtains $H(\alpha \mid \beta) = 0$ and $H(\beta \mid \alpha) = H(\beta) - H(\alpha)$, which results in $d(\alpha, \beta) = 0$, regardless of p.

Similarly, $d(\alpha, \gamma) = H(\gamma) - H(\alpha)$, $d(\alpha, \gamma) = H(\gamma) - H(\beta)$, so if $d(\alpha, \gamma) = d(\beta, \gamma)$, then $d(\alpha, \beta) = 0$. ■

Claim 3. *If $(\chi, \prec)$ is ordered, then metric d is non-decreasing in $(\chi, \prec)$; moreover, d is concave in $(\chi, \prec)$.*

Proof. Let $\alpha, \beta, \xi \in (\chi, \prec)$ and $\alpha \prec \xi \prec \beta$. Since $H(\alpha) \leq H(\xi) \leq H(\beta)$ it implies that

$$d(\theta, \alpha) \leq d(\theta, \xi) \leq d(\theta, \beta).$$

Thus, metric d is non-decreasing in $(\chi, \prec)$.

The concavity of d for the linear structure $(\chi, \prec)$ follows from concavity of the entropy (Cover and Thomas, 1991). ■

Proof of the Lemma. Under the condition of the lemma for every $\alpha \in (\chi, \prec)$, we have

$$d_p(\theta, \alpha) < d_u(\theta, \alpha).$$

In fact, if $d_p(\theta, \alpha) > d_u(\theta, \alpha)$, then, since d is concave and $d_p(\alpha, \beta) \leq d_p(\theta, \gamma)$, there exists such a partition $\xi \in (\chi, \prec)$ that $d_p(\theta, \xi) = d_u(\theta, \xi)$, which is impossible since $H_p(\alpha) \neq H_u(\alpha)$.

Let $\alpha, \beta \in (\chi, \prec)$ and let $\alpha \prec \beta$, without loss of generality. Both $d_p(\alpha, \alpha)$ and $d_u(\alpha, \alpha) = 0$ and, under the lemma's requirement, $d_p(\theta, \gamma) < d_u(\theta, \gamma)$, distance d_u increases faster than distance d_p. Hence,

$$d_p(\theta, \beta) - d_p(\theta, \alpha) < d_u(\theta, \beta) - d_u(\theta, \alpha).$$

Now recall that if $\alpha \prec \beta$ then $d(\alpha, \beta) = H(\beta) - H(\alpha)$ and that $d(\theta, \beta) = H(\beta)$, $d(\theta, \alpha) = H(\alpha)$; thus, we obtain the required inequality. ■

Proof of Lemma 3. Consider the triangle (α, β, γ) with the corresponding partitions in its vertices.

Let $\alpha_{\mathrm{cur}} = \alpha$, and suppose that the algorithm, in contrast to the statement of the lemma, selects partition β. That is,

$$d(\alpha, \beta) + \tilde{d}(\beta, \gamma) \leq d(\alpha, \gamma) + \tilde{d}(\gamma, \gamma) = d(\alpha, \gamma).$$

Since $\gamma \in N(\beta)$, it follows that $\tilde{d}(\beta,\gamma) = d(\beta,\gamma)$. Thus, $d(\alpha,\beta) + d(\beta,\gamma) \le d(\alpha,\gamma)$. But, according to the triangle inequality, $d(\alpha,\beta) + d(\beta,\gamma) \ge d(\alpha,\gamma)$.

Hence, the only possible selection of β is if $\beta = \gamma$. ■

Proof of Lemma 4. Consider the case of choosing partition β. For the chosen partition β, and according to the triangle inequality and admissibility assumption (8), it follows that

$$\tilde{d}(\alpha,\gamma) \le d(\alpha,\beta) + \tilde{d}(\beta,\gamma).$$

Thus, in step 3*b* of the ILRTA* algorithm the estimation updating is given by

$$\tilde{d}(\alpha,\gamma) \leftarrow d(\alpha,\beta) + \tilde{d}(\beta,\gamma).$$

Let $\omega \in N(\beta)$ and suppose that the next partition chosen after β is $\alpha \in N(\beta)$. Thus,

$$2d(\alpha,\beta) + \tilde{d}(\beta,\gamma) + \tilde{d}(\alpha,\gamma) \le d(\beta,\omega) + \tilde{d}(\omega,\gamma).$$

Taking into account that if $\alpha \in N(\beta)$ and $\omega \in N(\beta)$ then $d(\alpha,\beta) = \tilde{d}(\alpha,\beta)$ and $d(\beta,\omega) = \tilde{d}(\beta,\omega)$, leading to

$$2\tilde{d}(\alpha,\beta) + \tilde{d}(\beta,\gamma) + \tilde{d}(\alpha,\gamma) \le \tilde{d}(\beta,\omega) + \tilde{d}(\omega,\gamma).$$

Finally, recalling that the distance estimation $\tilde{d}$ meets the requirements of a metric we obtain

$$2\tilde{d}(\alpha,\beta) + \tilde{d}(\beta,\gamma) + \tilde{d}(\alpha,\gamma) \le \tilde{d}(\beta,\gamma).$$

Accordingly, $2\tilde{d}(\alpha,\beta) + \tilde{d}(\alpha,\gamma) = 0$ implies that both $\alpha = \beta$ and $\alpha = \gamma$, which is impossible. Therefore, the chosen partition has to be ω and the algorithm does not return to α in the next step. ■

Proof of Theorem 3. The existence of at least one path from α_{cur} to the final partition γ implies that for some α, $\gamma \in N(\alpha)$. If $\alpha_{\text{cur}} = \alpha$, then according to Lemma 3, the final partition γ is selected and the algorithm terminates.

Let $\alpha_{\text{cur}} = \beta \ne \alpha$ and $\gamma \notin N(\beta)$. Then, according to Lemma 4, the search algorithm does not return to the previous partition and chooses a partition from the neighborhood $N(\beta)$. If $\alpha \in N(\beta)$ is a single partition, then this partition will be chosen. Otherwise, assume that there exists a partition $\omega \in N(\beta)$, $\omega \ne \alpha$. Thus, according to Lemma 4, the algorithm chooses a partition ω. If $\gamma \in N(\omega)$, then, according to Lemma 3, the algorithm chooses the final partition γ and terminates. If $\gamma \notin N(\omega)$, then the same reasoning is applied to partition $\alpha_{\text{cur}} = \omega$.

Let $\alpha \notin N(\beta)$. If $\alpha_{\text{cur}} \in N(\beta)$ is a single partition in the neighborhood of β, then the next partition after β is α_{cur}. According to Lemma 4, the algorithm continues without returning to β and selects another partition from $N(\alpha_{\text{cur}})$.

Following backward induction, by applying a similar consideration to the partitions one step before the current selection, then two steps before the selection, and so on up to the initial partition θ, one obtains the statement of the theorem. ■

Proof of Lemma 5. If

$$\tilde{d}(\alpha_{\text{cur}},\gamma) \ge \min_{\alpha\in N(\alpha_{\text{cur}})}\{d(\alpha_{\text{cur}},\alpha) + \tilde{d}(\alpha,\gamma)\},$$

then the estimation updating is given by

$$\tilde{d}(\alpha_{\text{cur}},\ \gamma) \leftarrow \tilde{d}(\alpha_{\text{cur}},\ \gamma),$$

and the proposition follows directly from the inequality $\tilde{d}(\alpha,\gamma) \le d(\alpha,\gamma)$.

Let $\tilde{d}(\alpha_{\text{cur}},\ \gamma) < \min_{\alpha\in N(\alpha_{\text{cur}})}\{d(\alpha_{\text{cur}},\ \alpha) + \tilde{d}(\alpha,\ \gamma)\}$ and denote $\alpha_{\min} = \arg\min_{\alpha\in N(\alpha_{\text{cur}})}\{d(\alpha_{\text{cur}},\ \alpha) + \tilde{d}(\alpha,\ \gamma)\}$. We need to show that

$$d(\alpha_{\text{cur}},\ \alpha_{\min}) + \tilde{d}(\alpha_{\min},\ \gamma) \le d(\alpha_{\text{cur}},\ \gamma).$$

Suppose, on the other hand, that

$$d(\alpha_{\text{cur}},\ \alpha_{\min}) + \tilde{d}(\alpha_{\min},\ \gamma) > d(\alpha_{\text{cur}},\ \gamma).$$

Then, using the triangle inequality we obtain that

$$d(\alpha_{\text{cur}},\gamma) < d(\alpha_{\text{cur}},\alpha_{\min}) + \tilde{d}(\alpha_{\min},\gamma),$$
$$d(\alpha_{\text{cur}},\gamma) \le d(\alpha_{\text{cur}},\alpha_{\min}) + d(\alpha_{\min},\gamma).$$

Since $\alpha_{\min}$ is such a partition that supplies a minimal distance estimation, the segment $(\alpha_{\text{cur}},\ \alpha_{\min})$ is in the path that gives the exact distance value $d(\alpha_{\text{cur}},\ \gamma)$. Therefore,

$$d(\alpha_{\text{cur}},\ \gamma) = d(\alpha_{\text{cur}},\ \alpha_{\min}) + d(\alpha_{\min},\ \gamma),$$

and $\tilde{d}(\alpha_{\min},\gamma) > d(\alpha_{\min},\gamma)$, in contradiction to the assumption that $\tilde{d}(\alpha,\gamma) \le d(\alpha,\gamma)$ for every $\alpha \in \chi$. ■

Proof of Theorem 4. The statement of the theorem is an immediate consequence from Lemma 4, Lemma 5, and Theorem 3. In fact, let $\tilde{d}_0^i(\alpha,\gamma), \alpha \in \chi$, be an initial distance estimation in the *i*th trial. Then, if for the partitions $\alpha \in \chi$, which are included in the path on the *i*th trial, it follows that

$$\tilde{d}_0^i(\alpha,\gamma) < \tilde{d}^i(\alpha,\gamma) \le d(\alpha,\gamma),$$

then for the $(i+1)$th trial, these partitions have initial estimations:

$$\tilde{d}_0^{i+1}(\alpha,\gamma) = \tilde{d}^{i+1}(\alpha,\gamma) \le d(\alpha,\gamma).$$

Thus, for each path chosen on the next trial n we obtain

$$\tilde{d}_0^{i+n}(\alpha,\gamma) \to d(\alpha,\gamma) \quad \text{with} \quad n \to \infty,$$

for all partitions $\alpha \in \chi$ that are included in the path chosen in the *n*th trial.

Since the number of paths is finite and the distance estimations are strictly increasing with the trials, they will converge to their upper bound, which is provided by real distances according to admissibility requirement (8). Thus, after a finite number of trials, all distance estimations will be equal to the real distances, and the obtained path will follow such distances, as it is required. ■

Proof of Lemma 6. Let $\alpha_{\text{cur}} = \{A_1,\ A_2,\ \ldots,\ A_m\}$. Denote $p_i = p(A_i)$, $i = 1, 2, \ldots, m$, and suppose that $p_1 \le p_2 \le \ldots \le p_m$. It is clear that such ordering always exists and does not change the form of the Huffman tree.

Then, according to the Huffman procedure, the next partition $\alpha_{\text{next}} = \{A' = A_1 \cup A_2,\ A_3,\ \ldots,\ A_m\}$ with $p' = p_1 + p_2$, and we should prove that

$$\alpha_{\text{next}} = \{A_1 \cup A_2,\ A_3,\ \ldots,\ A_m\} = \arg\max_{\alpha \in N_{\text{Huf}}(\alpha_{\text{cur}})} \times \{H(\alpha_l \mid \alpha_{\text{cur}}) - H(\alpha_l \mid \alpha)\}.$$

Since $\alpha_{\text{cur}} \prec \alpha_l$ and $\alpha_{\text{next}} \prec \alpha_l$, for the conditional entropies $H(\alpha_l \mid \alpha_{\text{cur}})$ and $H(\alpha_l \mid \alpha_{\text{next}})$ according to Theorem 2 it holds true that $H(\alpha_l \mid \alpha_{\text{cur}}) = H(\alpha_l) - H(\alpha_{\text{cur}})$, and $H(\alpha_l \mid \alpha_{\text{next}}) = H(\alpha_l) - H(\alpha_{\text{next}})$.

Thus, we need to show that

$$H(p_1 + p_2, p_3, \ldots, p_m) \geq H(p_1 + p_i, p_2, \ldots, p_{i-1}, p_{i+1}, \ldots, p_m).$$

If $m = 2$, then the inequality is an obvious equality:

$$H(p_1 + p_2, 0) = H(p_1 + p_2, 0).$$

Let $m > 2$. Using the full form of the entropy, we obtain

$$-(p_1 + p_2)\log(p_1 + p_2) - \sum_{j=3}^{m} p_j \log p_j \geq -(p_1 + p_i)\log(p_1 + p_i) - \sum_{\substack{j=2 \\ j \neq i}}^{m} p_j \log p_j,$$

and, finally,

$$(p_1 + p_2)\log(p_1 + p_2) + p_i \log p_i \leq (p_1 + p_i)\log(p_1 + p_i) + p_2 \log p_2.$$

Recall that $0 < p_1 \leq p_2 \leq p_i$ and $p_1 + p_2 + p_i \leq 1$. Let us fix the probabilities p_1 and p_i, and consider the function:

$$f(p_2) = (p_1 + p_2)\log(p_1 + p_2) + p_i \log p_i - (p_1 + p_i)\log(p_1 + p_i) - p_2 \log p_2.$$

We need to show that $f(p_2) \leq 0$ for $p_1 \leq p_2 \leq p_i$. The first derivation $(\mathrm{d}/\mathrm{d}\,p_2)f$ of the function f is

$$(\mathrm{d}/\mathrm{d}\,p_2)f = \log(p_1 + p_2) - \log p_2 = \log(p_1 + p_2/p_2),$$

and the second derivation $(\mathrm{d}^2/\mathrm{d}(p_2)^2)f$ of the function f is

$$(\mathrm{d}^2/\mathrm{d}(p_2)^2)f = (1/\ln 2)(1/(p_1 + p_2) - 1/p_1).$$

Since $0 < p_1 \leq p_2 \leq p_i$, the first derivation $(\mathrm{d}/\mathrm{d}\,p_2)f$ is non-negative and the second derivation $(\mathrm{d}^2/\mathrm{d}(p_2)^2)f$ is negative. Hence, the function f increases with p_2 up to the value

$$f(p_i) = (p_1 + p_i)\log(p_1 + p_i) + p_i \log p_i - (p_1 + p_i)\log(p_1 + p_i) - p_i \log p_i = 0,$$

which is a maximum of the function f for $0 < p_1 \leq p_2 \leq p_i$.

Thus, given the current partition $\alpha_{\text{cur}} = \{A_1,\ A_2,\ \ldots,\ A_m\}$ such that $p(A_1) \leq p(A_2) \leq \ldots \leq p(A_m)$, the maximal value of the difference $H(\alpha_l \mid \alpha_{\text{next}}) - H(\alpha_l \mid \alpha_{\text{cur}})$ is reached for the next partition $\alpha_{\text{next}} = \{A' = A_1 \cup A_2,\ \ldots,\ A_m\}$. ■

Proof of Theorem 5. Consider the Huffman procedure. Note that as a bottom-up procedure, the Huffman procedure starts from the partition $\theta = \alpha_l$, and according to Lemma 6, the selection criterion is based on the difference

$$H(\theta \mid \alpha_{\text{cur}}) - H(\theta \mid \alpha), \quad \alpha \prec \alpha_{\text{cur}} \prec \theta, \quad \alpha \in \chi.$$

Hence, for the chosen partitions $\alpha_{l-i-1} \prec \alpha_l$, $i = 0, 1, \ldots, l-1$, by the Huffman procedure, it is true that $\theta = \alpha_l \succ \alpha_{l-1} \succ \ldots \succ \alpha_1 \succ \alpha_0 = \gamma$.

Based on Theorem 2, for every partition $\alpha \in \chi$ we get

$$H(\theta \mid \alpha_{\text{cur}}) - H(\theta \mid \alpha) = H(\theta) - H(\alpha_{\text{cur}}) - H(\theta) + H(\alpha) = H(\alpha) - H(\alpha_{\text{cur}}).$$

Thus, given the initial partition θ, and the selection criterion:

$$\alpha_{\text{next}} \leftarrow \arg\max_{\alpha \in N_{\text{Huf}}(\alpha_{\text{cur}})}\{H(\theta \mid \alpha_{\text{cur}}) - H(\theta \mid \alpha)\},$$

in step $2a$ of the Huffman procedure, we obtain

$$\alpha_{\text{next}} \leftarrow \arg\max_{\alpha \in N_{\text{Huf}}(\alpha_{\text{cur}})}\{H(\alpha) - H(\alpha_{\text{cur}})\},$$

which is equivalent to the selection criterion

$$\alpha_{\text{next}} \leftarrow \arg\min_{\alpha \in N_{\text{Huf}}(\alpha_{\text{cur}})}\{H(\alpha_{\text{cur}}) - H(\alpha)\}.$$

Now, consider the action choice in the ILTRA* algorithm:

$$\alpha_{\text{next}} \leftarrow \arg\min_{\alpha \in N(\alpha_{\text{cur}})}\left\{d(\alpha_{\text{cur}},\ \alpha) + \tilde{d}(\alpha,\ \gamma)\right\}.$$

Under the assumptions of the theorem, and taking into account the Huffman neighborhood, we get

$$\alpha_{\text{next}} \leftarrow \arg\min_{\alpha \in N_{\text{Huf}}(\alpha_{\text{cur}})}\{H(\alpha \mid \alpha_{\text{cur}}) + H(\alpha_{\text{cur}} \mid \alpha)\}$$

with $H(\alpha \mid \alpha_{\text{cur}}) = 0$ and $H(\alpha_{\text{cur}} \mid \alpha) = H(\alpha_{\text{cur}}) - H(\alpha)$. Hence,

$$\alpha_{next} \leftarrow \arg\min_{\alpha \in N_{Huf}(\alpha_{\text{cur}})}\{H(\alpha_{\text{cur}}) - H(\alpha)\},$$

which is equivalent to the selection criterion in the Huffman procedure.

Consider the estimation update in the ILRTA* algorithm. According to Lemma 4, the goal of the estimation update is to avoid a return of the algorithm to the previous partition in the same trial. This is equivalent to a deletion of this partition from the set of all available partitions—a rule that is executed by step $2b$ in the Huffman procedure.

Finally, we need to show that the termination rule $d(\alpha_{\text{cur}},\ \gamma) = 0$, which is used in the ILRTA* algorithm, is equivalent to the termination rule $H(\alpha_{\text{cur}} \mid \alpha_0) = 0$ of the Huffman procedure. In fact, for the Huffman procedure we have

$$d(\alpha_0,\ \alpha_{\text{cur}}) = H(\alpha_0 \mid \alpha_{\text{cur}}) + H(\alpha_{\text{cur}} \mid \alpha_0).$$

Since $\alpha_0 \prec \alpha_{\text{cur}}$, according to Theorem 1 it follows that $H(\alpha_0 \mid \alpha_{\text{cur}}) = 0$. Hence, $d(\alpha_0,\ \alpha_{\text{cur}}) = H(\alpha_{\text{cur}} \mid \alpha_0)$, resulting in the same termination rule. ■

Proof of Theorem 7. At first, let us demonstrate the equivalence of the action choice. Under the assumptions of the theorem, since $\alpha \succ \alpha_{\text{cur}}$ we obtain

$$d(\alpha_{\text{cur}}, \alpha) = H(\alpha_{\text{cur}} \mid \alpha) + H(\alpha \mid \alpha_{\text{cur}}) = H(\alpha) - H(\alpha_{\text{cur}}),$$

and the action choice of the ILTRA* algorithm is

$$\alpha_{\text{next}} \leftarrow \arg\min_{\alpha \in N(\alpha_{\text{cur}})} \{H(\alpha) - H(\alpha_{\text{cur}})\}.$$

Under the assumptions on the cost function g it follows that

$$\frac{H(\gamma \mid \alpha_{\text{cur}}) - H(\gamma \mid \alpha)}{g(\alpha_{\text{cur}}, \alpha)} = \frac{H(\alpha) - H(\alpha_{\text{cur}})}{g(\alpha_{\text{cur}}, \alpha)} = \frac{H(\alpha_{\text{cur}}) - H(\alpha)}{g(\alpha, \alpha_{\text{cur}})}.$$

Accordingly, the selection by the GOTA can be expressed as follows:

$$\alpha_{\text{next}} \leftarrow \arg\max_{\alpha \in N_G(\alpha_{\text{cur}})} \left\{ \frac{1}{g(\alpha, \alpha_{\text{cur}})} \left(H(\alpha_{\text{cur}}) - H(\alpha)\right) \right\},$$

which is equivalent to the selection by the ILRTA* algorithm with a proper neighborhood definition:

$$\alpha_{\text{next}} \leftarrow \arg\min_{\alpha \in N_G(\alpha_{\text{cur}})} \left\{ \frac{1}{g(\alpha_{\text{cur}}, \alpha)} \left(H(\alpha) - H(\alpha_{\text{cur}})\right) \right\}.$$

Now let us consider the termination rule. For the ILRTA* algorithm:

$$d(\alpha_{\text{cur}}, \gamma) = H(\alpha_{\text{cur}} \mid \gamma) + H(\gamma \mid \alpha_{\text{cur}}).$$

If $\alpha_{\text{cur}} = \gamma$, then both $H(\alpha_{\text{cur}} \mid \gamma) = 0$ and $H(\gamma \mid \alpha_{\text{cur}}) = 0$, and the equivalence of the GOTA's termination rule to the termination rule of the ILRTA* algorithm is satisfied.

Let $\alpha_{\text{cur}} \neq \gamma$. While $\alpha_{\text{cur}} \prec \gamma$, it follows that $H(\gamma \mid \alpha_{\text{cur}}) > 0$, and when $\alpha_{\text{cur}} \succ \gamma$, then $H(\gamma \mid \alpha_{\text{cur}}) = 0$. Thus, the GOTA's termination rule $H(\gamma \mid \alpha_{\text{cur}}) = 0$ is correct.

The reasons for the estimation update are similar to the ones used in the proof of Theorem 5, which addresses the Huffman procedure. ■

PSEUDO-CODE: MAIN FUNCTION OF THE ILRTA* SIMULATIONS

```
(1) Create: sample space X and partitions
    space χ.
(2) For sessions number = 0 to 1000 do
    a. Init sample space X by location
       probabilities.
    b. If required for comparison, create
       a Maximum-entropy search tree or a
       Huffman search tree.
    c. Init target's location and create
       corresponding TargetPartition.
    d. Do
         CurrentSearcherPartition ←
         Searcher.step()
       While
         CurrentSearcherPartition ≠
         TargetPartition
(3) Delete sample space X and partitions
    space χ.
```

Biographies

Eugene Kagan earned his M.Sc. (1991) and Cand.Sc. (Ph.D.) (2004) from Taganrog State University for Radio-Engineering (Russia) and Ph.D. (2010) from Tel-Aviv University, Israel. He has more than 20 years' research experience in applied mathematics and engineering and spent more than 10 years in industry and in software engineering. Currently he is with the Department of Computer Science and Applied Mathematics at the Weizmann Institute of Science, Israel. He is a co-author of *Probabilistic Search for Tracking Targets* (Wiley & Sons, 2013), authored more than 40 scientific papers and conference reports, and contributed to several collective monographs.

Irad Ben-Gal is the Head of the Department of Industrial Engineering & Management at Tel Aviv University. His research interests include statistical methods for control and analysis of complex processes, machine learning applications to industrial and service systems, and big data analytics. He holds a B.Sc. (1992) degree from Tel-Aviv University and M.Sc. (1996) and Ph.D. (1998) degrees from Boston University. He has written and edited five books, has published more than 80 scientific papers and patents, received numerous best papers awards, and supervised dozens of graduate students. He serves on the Editorial Boards of several international journals. He is the Chair-Elect of the Quality Statistics and Reliability Society of the Institute for Operations Research and Management Sciences, the European Network for Business and Industrial Statistics, and the International Statistical Institute. He has received several research grants and awards from General Motors, IEEE, Israel Ministry of Science, Israel Scientific Foundation, Israeli Prime Minister Ministry, and the European Community. He has led many R&D projects and worked with companies such as IBM, Proctor and Gamble, Kimberly-Clark, Applied Materials, SAP, and more. He was a co-founder of Context-Based 4casting (C-B4), a software company that develops novel predictive analytics tools for retail and service organizations.