On the Analysis of a Random Walk-Jump Chain with *Tree*-based Transitions, and its Applications to Faulty Dichotomous Search*

Anis Yazidi† and B. John Oommen‡

Abstract

Abstract: Random Walks (RWs) have been extensively studied for more than a century [1]. These walks have traditionally been on a *line*, and the generalizations for two and three dimensions, have been by extending the random steps to the corresponding neighboring positions in one or many of the dimensions. Among the most popular RWs on a line are the various models for birth and death processes, renewal processes and the gambler's ruin problem. All of these RWs operate "on a discretized line", and the walk is achieved by performing small steps to the current-state's neighbor states. Indeed, it is this neighbor-step motion that renders their analyses tractable. When some of the transitions are to non-neighbour states, a formal analysis is, typically, impossible because the difference equations of the steady-state probabilities are not solvable. One endeavor on such an analysis is found in [2]. The problem is far more complex when the transitions of the walk follow an underlying tree-like structure. The analysis of RWs on a *tree* have received little attention, even though it is an important topic since a tree is a counter-part space representation of a line whenever there is some ordering on the nodes on the line. Nevertheless, RWs on a tree entail moving to *non-neighbor* states in the space, which makes the analysis involved, and in many cases, impossible.

In this paper, we consider the analysis of one such fascinating RW. We demonstrate that an analysis of the chain is feasible because we can invoke the phenomenon of "time reversibility". Apart from the analysis being interesting in itself from an analytical perspective, the RW on the tree that this paper models, is a type of generalization of dichotomous search with faulty feedback about the direction of the search, rendering the real-life application of the model to be pertinent. To resolve this, we advocate the concept of "backtracking" transitions in order to efficiently explore the search space. Interestingly, it is precisely these "backtracking" transitions that naturally render the chain to be "time reversible". By doing this, we are able to bridge the gap between deterministic dichotomous search and its faulty version. The paper contains the analysis of the chain, reports some fascinating limiting properties, and also includes simulations that justify the analytic steady-state results.

*The second author is grateful for the partial support provided by NSERC, the Natural Sciences and Engineering Research Council of Canada.

†This author can be contacted at: Dept. of ICT, Oslo and Akershus University College, Oslo, Norway. E-mail: anis.yazidi@hioa.no.

‡Author's status: *Chancellor's Professor*, *Fellow: IEEE* and *Fellow: IAPR*. This author can be contacted at: School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6. The author is also an Adjunct Professor with University of Agder, Grimstad, Norway. E-mail: oommen@scs.carleton.ca.

Keywords: *Time Reversibility, Controlled Random Walk, Random Walk with Jumps, Dichotomous Search, Learning Systems*

1 Introduction:

The theory of Random Walks (RWs) and its applications have gained an "exponential" amount of research interest since the early part of the last century. From the recorded literature, one perceives that the pioneering treatment of a one-dimensional RW was due to Pearson in [3]. The RW is, usually, defined as a trajectory involving a series of successive random steps, which are, quite naturally, modeled using Markov Chains (MCs). MCs are probabilistic structures that possess the so-called "Markov property" – which, informally speaking, implies that the next "state" of the walk depends on the current state and not on the entire past states (or history). The latter property is also referred to as the "lack of memory" property, which imparts to the structure practical consequential implications since it permits the modeler to predict how the chain will behave in the immediate and distant future, and to thus quantify its behavior.

Applications of RWs: It would be no exaggeration to state that tens of thousands of papers have been written that either deal with the analysis of RWs or their applications. Embarking on a comprehensive survey would thus be meaningless. In all brevity, we mention that RWs have been utilized in a myriad of applications stemming from areas as diverse as biology, computer science, economics and physics. For instance, concrete examples of these applications in biology are the epidemic models described in [4], the Wright-Fisher model, and the Moran Model in [5] etc... RWs arise in the modeling and analysis of queuing systems [6], ruin problems [7], risk theory [8], and sequential analysis and learning theory as demonstrated in [9]. In addition to the above-mentioned *classical* application of RWs, recent applications include mobility models in mobile networks [10], collaborative recommendation systems [11], web search algorithms [12], and reliability theory for both software/hardware components [13] (pp. 83–111).

Classification of RWs: RWs can be broadly classified in terms of their Markovian representations. Generally speaking, RWs are either ergodic or possess absorbing barriers. In the simplest case, the induced MC is ergodic, implying that sooner or later, each state will be visited (w. p. 1), independent of the initial state. In such MCs, the limiting distribution of being in any state is independent of the corresponding initial distribution. This feature is desirable when the directives dictating the steps of the chain are a consequence of interacting with a non-stationary environment, allowing the walker to not get trapped into choosing any single state. Thus, before one starts the analysis of a MC, it is imperative that one understands the nature of the chain, i.e., if it is ergodic, which will determine whether or not it possesses a stationary distribution.

A RW can also possess absorbing barriers. In this case, the associated MC has a set of transient states which it will sooner or later never visit again. When the walker reaches an absorbing barrier, it is "trapped", and is destined to remain there forever. RWs with two absorbing barriers have also been applied to analyze problems akin to the two-choice bandit problems in [14] and the gambler's ruin problem in [7], while their generalizations to chains with multiple absorbing barriers have their analogous extensions.

Although RWs are traditionally considered to be uni-dimensional (i.e., on the line), multi-dimensional

RWs operate on the plane or in a higher dimensional space.

The most popularly-studied RWs are those with single step transitions. The properties of such RWs have been extensively investigated in the literature. A classical example a RW of this type is the ruin problem in [7]. In this case, a gambler starts with a fortune of size s, and decides to play until he is either ruined (i.e. his fortune decreases to 0), or until he has reached a fortune of M. At each step, the gambler has a probability, p, of incrementing his fortune by a unit, and a chance $q = 1 - p$ of losing a unit. The actual capital possessed by the gambler is represented by a RW on the line of integers from 0 to M, with the states 0 and M serving as the respective absorbing barriers. Of course, the game changes drastically to be ergodic if a player is freely given a unit of wealth if he is bankrupt, i.e., when his fortune is 0, and he forfeits a unit if he attains the maximum wealth of M. In these cases, the respective boundaries are said to be "reflecting".

Analysis of Ergodic RWs: Ergodic MCs possess the fascinating property that the probabilities of being in the various states converge to an asymptotic value, also known as the steady state or stationary distribution. For a chain with W states, characterized by the Markov matrix, H, this distribution, say Π, satisfies:

$$H^T \Pi = \Pi. \tag{1}$$

Most of the RWs that have been formally analyzed operate "on a discretized line", and since the walk is achieved by performing small steps to the current-state's neighbor states, such a neighbor-step motion renders their analyses tractable. This is because the asymptotic probability, π_i, of being in state i, can be written in terms of π_j, where $\{j\}$ are integers centered around, or in the neighborhood of i. This then reduces to solving difference equations of π_i in terms of the π_j's.

Analysis of Ergodic RWs with "Jumps" (RWJ): When some of the transitions are to non-neighbour states, the MC takes a "jumps" to such a non-neighbor state, rather than a step. A formal analysis of RWJs is, typically, impossible because there are no known techniques to solve the the corresponding difference equations of the steady-state probabilities. The literature on RWJs is extremely sparse. One example RWJ was reported in [2], and it was applied in the online tracking of spatio-temporal event patterns in [15,16].

Analysis of Ergodic RWs on Trees: Although RWs with with transitions on a line, such as the gambler's ruin problem, have been extensively studied for almost a century, as one can observe from [1], that problems involving the analysis of RWs on a tree are intrinsically hard and have received little research attention. This is because they involves the hardest concepts of two arenas: Firstly, they involve specific RWJs, where the transitions are to non-neighbor states. Secondly, the non-neighbor states have an additional constraint in that they are associated with an underlying tree structure, as opposed to a line. In this paper, we consider the analysis of one such fascinating RWJ. Although the analysis is seemingly impossible, we shall show that because of the nature of the tree-based transitions, the phenomenon of time reversibility can be invoked, and thus the analysis can be achieved.

Application of our tree-based RW: Although the analysis of our chain is pioneering and is a contribution in its own right, it turns out that our RW is a generalization of a dichotomous search when the feedback about the direction of the search is faulty or erroneous. In fact, typical classically-known dichotomous search schemes are deterministic in the sense that they work with non-faulty feedback. Thus, it is possible to eliminate a whole subtree at each iteration, effectively shrinking the search space by half, until the target node is discovered. However, under faulty feedback possessing a stochastic nature, the most intuitive way to solve the problem is to perform a large enough number of tests at each level of the tree before deciding whether to eliminate either the left or right subtree, and to then proceed iteratively until the deepest level is reached [17]. Unfortunately, however, if by mistake a subtree that contains the target node is eliminated the whole search process is "misled". More precisely, schemes such as those reported in [17] are prone to error[1].

How then do we circumvent the problem alluded to above? In our solution, we propose that the RW does "backward" transitions in the tree so that to avoid the problem of sampling with a large number of tests at each level. In addition, the RW is made "non-absorbing" so that to avoid getting trapped into a subtree that does not include the target node. All of these concepts will be explained presently.

We believe that the current RW analyzed in this paper is important in its own right since it is known that a dichotomous search of a sorted list is an order of magnitude faster than a sequential search. One should further note that any line structure can be easily mapped into a tree structure if the nodes on the tree have some inherent ordering between themselves.

History of tree-based RWs: To present the results of this paper in the right perspective, we mention that the problem of considering RWs on a tree for parameter optimization [19] was first introduced by the authors of the current paper in [20] and extended by Zhang et al. [18] to the case of symmetrical environments. The current work is a generalization of our previous work [20] as we derive a general result which does not involve the golden ratio condition as a lower bound on the transition probabilities along the optimal path in the tree, but rather the more intuitively appealing ratio, namely $\frac{1}{2}$. It can thus be applied in many random search problems. Furthermore, the transitions of the RW analyzed here are different from the one proposed in [20] as the only allowed self-transitions take place at the leaves while no self transition takes place at the root node. Additionally, this paper provides a guideline for the analysis of RWs on tree structures, and thus, the design of RW schemes can be rendered more useful in designing engineering applications involving some dichotomous faulty feedback responses. Finally, and most importantly, the current RW on a tree has a very genuine interpretation as a form for dichotomous search with faulty feedback.

1.1 Contributions of this Paper

The novel contributions of this paper are the following:

- We submit a pioneering analysis for Random Walks with Jumps (RWJs) where the "jumps" are done along the nodes of a tree superimposed over the line;
- The RWJ permits self loops at certain "terminal" states of the chain;

[1] These algorithms can be designed to yield a high probabilistic guarantee of reaching the target. To do this, they formulate a limiting assumption: In order to choose a number of tests at each level that is large enough, one needs to know the exact probability by which the environment suggests the right direction of the search. In this paper, we assume that the latter probability is unknown and that the only known clue is that this quantity has a lower bound of $\frac{1}{2}$. The case when the probability of correct transition is less than $\frac{1}{2}$ is referred to as the symmetric environment [18].

- The analysis that we submit utilizes the phenomenon of *time reversibility*, which provides us with a powerful tool for analyzing such RWJs. We hope that our current analysis can pave the way towards more interest and analysis of RWs of similar type.

- Our RW is a generalization of a dichotomous search with faulty feedback about the direction of the search. We show that under a simple condition on the correctness of the feedback from an Oracle (Environment), i.e, p, it is possible to accumulate the mass of the steady state probability of the RW arbitrarily close to the optimal target node in the tree.

- We advocate the concept of probabilistic "backtracking" in dichotomous search which permits the mechanism to "target" the direction of the search whenever the RW wrongly moves in a subtree which does not include the target node. Interestingly, these "backtracking" transitions naturally lead to bridging the gap between dichotomous search and "reversible Markov chains". In fact, in the case of our chain, the property of "reversibility " of our Makov Chain emerges as a consequence of backtracking, which is a very intriguing result in itself. Ironically, the classical dichotomous search schemes only use "downtracking" and never invoke a "backtracking" operation. Although the idea behind "backtracking" is simple, its implications are powerful, and leads to an anlysis that allows for time reversibility.

- The current work is an example where relatively straightforward but powerful scientific concepts can be applied to yield significant results. This simplicity renders it easily readable and offers the readers a promising tool in the design of AI-based engineering applications. We are currently investigating further applications of the RW specially in problems involving some advanced types of local search.

1.2 Organization of this Paper

After having introduced the problem in Section 1, we proceed in Section 2 to introduce the powerful concept of Time Reversible MCs which makes our analysis possible. In Section 3, we describe the RW as well as present some analytical results. In Section 4, we report experimental results that support our theoretical results which include steady-state results and simulation results for the chain's transient behavior.

2 Time Reversible Markov Chains

A fundamental contribution of this paper is the analysis of a RW that is superimposed on a tree structure using the concepts of time reversibility. This model can be see as a generalization of a deterministic dichotomous search. Since this is crucial to this paper, this phenomenon is briefly surveyed here.

Some Markov chains have the property that the process behaves in just the same way regardless of whether time is measured forwards or backwards. Kelly [21] made an analogy saying that "if we take a film of such a process and then run the film backwards, the resulting process will be statistically indistinguishable from the original process." This property is described formally in the following definition.

Definition: A stochastic process $X(t)$ is time reversible if a sequence of states $(X(t_1), X(t_2), \ldots, X(t_n))$ has the same distribution as the reversed sequence $(X(t_n), X(t_{n-1}), \ldots, X(t_1))$ for all $t_1, t_2, \ldots, t_n$. □

Consider a stationary Ergodic Markov chain (that is, a Markov chain that has been in operation for a long time) having transition probabilities M_{st} and stationary probabilities $P\{\pi_s\}$. Suppose that starting at some time we trace the sequence of states going backwards in time. That is, starting at time t, consider the sequence of states $X_t,\ X_{t-1},\ X_{t-2},\ \ldots\ X_0,$ It turns out that this sequence of states is itself a Markov chain with transition probabilities $Q_{st} = (\,P\{\pi_t\}/P\{\pi_s\}\,) * M_{ts}$. If $Q_{st} = M_{st}$ for all $s,\ t,$ then the Markov chain is said to be *time reversible*. Note that the condition for time reversibility, namely $Q_{st} = M_{st},$ can also be expressed as

$$P\{\pi_s\}\, M_{st} \;=\; P\{\pi_t\}\, M_{ts} \qquad \text{for all} \quad s \neq t\,. \tag{2}$$

The condition in the above equation can be stated as, for all states s and t, the rate at which the process goes from s to t (namely $P\{\pi_s\}\, M_{st}$) is equal to the rate at which the process goes from t to s (namely $P\{\pi_t\}\, M_{ts}$). It is worth noting that this is an obvious necessary condition for time reversibility since a transition from s to t going backward in time is equivalent to a transition from t to s going forward in time. Thus, if $\pi_m = s$ and $\pi_{m-1} = t,$ then a transition from s to t is observed if we are looking backward, and one from t to s if we are looking forward in time.

The following theorem adapted from Ross and used universally ([21–23]) gives the necessary and sufficient condition for a finite ergodic Markov chain to be time reversible. The proof of the theorem can be found in [23] (see Page 143).

Theorem 1. *A finite ergodic Markov chain for which $M_{st} = 0$ whenever $M_{ts} = 0$ is time reversible if and only if starting in state s, any path back to s has the same probability as the reversed path. That is, if*

$$M_{s,s_1}\, M_{s_1,s_2}\, \ldots\, M_{s_k,s} \;=\; M_{s,s_k}\, M_{s_k,s_{k-1}}\, \ldots\, M_{s_1,s}$$

for all states $s,\ s_1,\ \ldots,\ s_k$. □

Using the above theorem, we state the result that any tree structure associated with a finite stationary Markov process is time reversible. This follows from the avenue that a Markov chain resulting from the "transition" operations on any tree structure is time reversible. In fact, this result is not totally new. Kelly (see [21], Page 9) proved the following lemma.

Lemma (Adapted from Kelly [21].) *If the graph G associated with a stationary Markov process is a tree, then the process is time reversible.* □

Although Kelly reported this result, he did not demonstrate how to associate a tree with a stationary Markov chain. In this paper, we shall give a formal definition for one such tree structure by organizing the points on the line along a tree and prove the corresponding theorem regarding its time reversibility. The application of time reversibility in the domain of self-organizing lists has been reported elsewhere [24].

3 The Random Walk on Tree

The space of the search is arranged in the form of a binary tree with depth $D = log_2(N)$, where N is an integer. The Random Walker (RW) searches for the target leaf node by orchestrating a controlled random walk on a tree. We assume the existence of an "Oracle", also referred to as the Environment, Ξ, which informs the RW, possibly erroneously (i.e., w. p. p), which way it should move to reach the target node[2].

3.1 Definitions

Construction of hierarchy. The search space is constructed as follows: First of all, the hierarchy is organized as a balanced binary tree with maximal depth D. For convenience, we will use the same notation adopted in [20, 25] and index the nodes using both their depth in the tree and their relative order with respect to the nodes located at the same the depth.

Root node. The hierarchy root (at depth 0), which we call $S_{\{0,1\}}$.

Nodes at depth d. Node $j \in \{1, ..., 2^d\}$ at depth d, called $S_{\{d,j\}}$, where $0 < d < D$ which has two children nodes $S_{\{d+1,2j-1\}}$ and $S_{\{d+1,2j\}}$. We say that $S_{\{d+1,2j-1\}}$ is the *Left Child* of $S_{\{d,j\}}$ and $S_{\{d+1,2j\}}$ is its *Right Child*.

Nodes at depth D. At depth D, which represents the maximal depth of the tree, the nodes do not have children.

Convention regarding the Leaves' notation. In a same vein, since level "$D + 1$" is nonexistent, we use the convention that *Right Child* of a leaf node is the same as the leaf node in question itself. Similarly, the *Left Child* of a leaf node is the leaf node itself. Formally, we say that:

$$\textit{Left Child}(S_{\{D,j\}}) = \textit{Right Child}(S_{\{D,j\}}) = S_{\{D,j\}} \text{ for } j \in \{1, ..., 2^D\}.$$

Target Node. The **target node** is a unique leaf node. We will later show that by imposing a simple condition on the "effectiveness" of the Environment (a concept defined presently), that we can concentrate the RW to be arbitrarily close to the target node.

Non-Target Node. They are leaf nodes different from the target node.

Effectiveness of the Walker. The probability to make a transition along the shortest path to the target node is denoted p and is a constant. Effectively, this probability can be seen as the probability of the RW making a "correct" movement towards the target node. However, whenever the random walker is situated at the target node itself, the effectiveness translates into the probability of it staying in the target node, i.e, this is represented by the probability of self-transition.

3.2 Structure of the Search Space

We intend to organize the search space in the form of a balanced binary tree. The random walker searches for the target node by operating a random walk on the tree, moving from one tree node to another.

[2]The reader should note that we could have defined the RW without even invoking the concept of the "Oracle" or Environment, Ξ, since it is merely a fictitious concept introduced to ease the understanding and readability of the paper. In addition, it emphasizes the analogy with dichotomous search as we can see the Environment as a type of faulty Oracle as opposite to a "perfect" Oracle, as in a classical dichotomous search.

At any given time instance, the RW finds itself at a node $S_{\{d,j\}}$ in the tree, where $j \in \{1, \ldots, 2^d\}$ and $0 \leq d \leq D$. The Environment Ξ essentially instructs the RW, possibly erroneously (i.e., with probability p), which way it should move to reach the target node.

- *Reverse Transitions*: Transitions of this type correspond to a movement to a lower level in the hierarchy. This happens when the RW moves to the immediate *Parent* which allows the RW to escape from getting trapped in a wrong subtree, i.e. one that does not contain target node.
- *Top-down Transitions*: Transitions of this type correspond to a movement to a deeper level in the hierarchy. The RW performs a transition to a deeper level in the hierarchy by choosing a *Child* node which is, hopefully, contained in a subtree that contains the target node.
- The only reflexive transitions (self-transitions) are allowed at the leaf node. We emphasize that we do not allow a reflexive transition in the root node as done in our previous work [20].

This concludes the description of our RW. We shall now investigate its convergence properties.

3.3 Analysis of the Solution

In this section, we shall prove that the RW is asymptotically optimal. We shall show that, eventually, based on an informed series of guesses, the RW will be able to concentrate its moves within nodes in the tree that are associated with the target node, and this will be true if p is larger than 0.5.

Theorem 2. *We suppose that the effectiveness p of the environment is strictly larger than 0.5. Let $S_{\{D,j^*_D\}}$ be the target node. Formally, $Lim_{D\to\infty}\pi_{\{D,j^*_D\}} \to 1$.*

Proof: Our intention is to prove that as D increased indefinitely, $Lim_{D\to\infty}\pi_{\{D,j^*_D\}} \to 1$. We shall prove this by analyzing the properties of the underlying Markov chain.

Let H be the corresponding transition matrix. Clearly, H represents a single closed communicating class whose periodicity is unity. The chain is ergodic, and the limiting probability vector is given by the eigenvector of H^T corresponding to eigenvalue unity.

$\Pi = [\pi_{\{0,1\}}, \pi_{\{1,1\}}, \pi_{\{1,2\}}, \ldots, \pi_{\{D,1\}}, \pi_{\{D,2\}}, \ldots, \pi_{\{D,2^D\}}]$. Then Π satisfies

$$H^T\Pi = \Pi. \tag{3}$$

Since the tree is a complete binary tree, in total, the tree contains $2^{D+1} - 1$ nodes (since $2^{D+1} - 1 = 1 + 2 + ... + 2^D$).

We now specify the elements of the transition matrix, H. For each node in the tree, we will give the expression for the transition probabilities to the next states (nodes). To achieve this task, we distinguish three cases, namely whether the considered node is a root node, intermediate node or a leaf node.

Let j^*_D be the relative index of the target node among the leaf nodes located at level D.

Let $Q_{\{d,j\},\{D,j^*_D\}}$ denote the shortest path in the balanced tree from the node $S_{\{d,j\}}$ to the target node $S_{\{D,j^*_D\}}$. Whenever there is no confusion, we will denote the latter quantity as $Q_{\{d,j\}}$.

Transitions at the root node. Consider the root node $S_{\{0,1\}}$. Since we do not allow a self-transition (from the root node to itself) as in [20], this translates simply to the fact that $p_{\{0,1\},\{0,1\}} = 0$.

Concerning the transitions to the children nodes from the root node, two cases emerge according to whether the target node belongs to the left subtree or to the right subtree:

- If $S_{\{0,1\},\{1,1\}} \in Q_{\{0,1\}}$, then:

  $p_{\{0,1\},\{1,1\}} = p$

  $p_{\{0,1\},\{1,2\}} = 1 - p.$

- If $S_{\{0,1\},\{1,2\}} \in Q_{\{0,1\}}$

  $p_{\{0,1\},\{1,1\}} = 1 - p$

  $p_{\{0,1\},\{1,2\}} = p.$

Transitions at intermediate nodes. Consider an intermediate node $S_{\{d,j\}}$, i.e, node $j \in \{1, \ldots, 2^d\}$ at depth d where $0 < d < D$. In order to specify the transitions probabilities at an intermediate node, we have to consider the following three cases:

1. If $S_{\{d,j\}} \notin Q_{\{d,j\}}$, then:

   $p_{\{d,j\},\{d-1,\lceil j/2 \rceil\}} = p$

   $p_{\{d,j\},\{d+1,2j-1\}} + p_{\{d,j\},\{d+1,2j\}} = 1 - p$

   Informally, the above equations simply mean that backtracking has highest probability if the subtree rooted in $S_{\{d,j\}}$ does not contain the target node.

2. Else if $S_{\{d,j\}} \in Q_{\{d,j\}}$, then:

   (a) If $S_{\{d+1,2j-1\}} \in Q_{\{d,j\}}$, then:

   $p_{\{d,j\},\{d+1,2j-1\}} = p$

   $p_{\{d,j\},\{d-1,\lceil j/2 \rceil\}} + p_{\{d,j\},\{d+1,2j\}} = 1 - p.$

   (b) If $S_{\{d+1,2j\}} \in Q_{\{d,j\}}$, then:

   $p_{\{d,j\},\{d+1,2j\}} = p$

   $p_{\{d,j\},\{d-1,\lceil j/2 \rceil\}} + p_{\{d,j\},\{d+1,2j-1\}} = 1 - p$

Transitions at the leaf nodes. The transitions probabilities at a leaf node depend on whether the latter is a target node or a non-target node. We consider each of these cases individually.

- For a non-target leaf node:

$p_{\{D,j\},\{D-1,\lceil j/2 \rceil\}} = p$

$p_{\{D,j\},\{D,j\}} = 1 - p$

- For the target node:

  $p_{\{D,j\},\{D,j\}} = p$

  $p_{\{D,j\},\{D-1,\lceil j/2 \rceil\}} = 1 - p.$

To clarify situations, we present a graphical example for the case when the depth of the tree is 3. Figure 1 specifies the transition matrix of the associated Markov chain when node $S_{\{3,7\}}$ is the *target leaf node*.

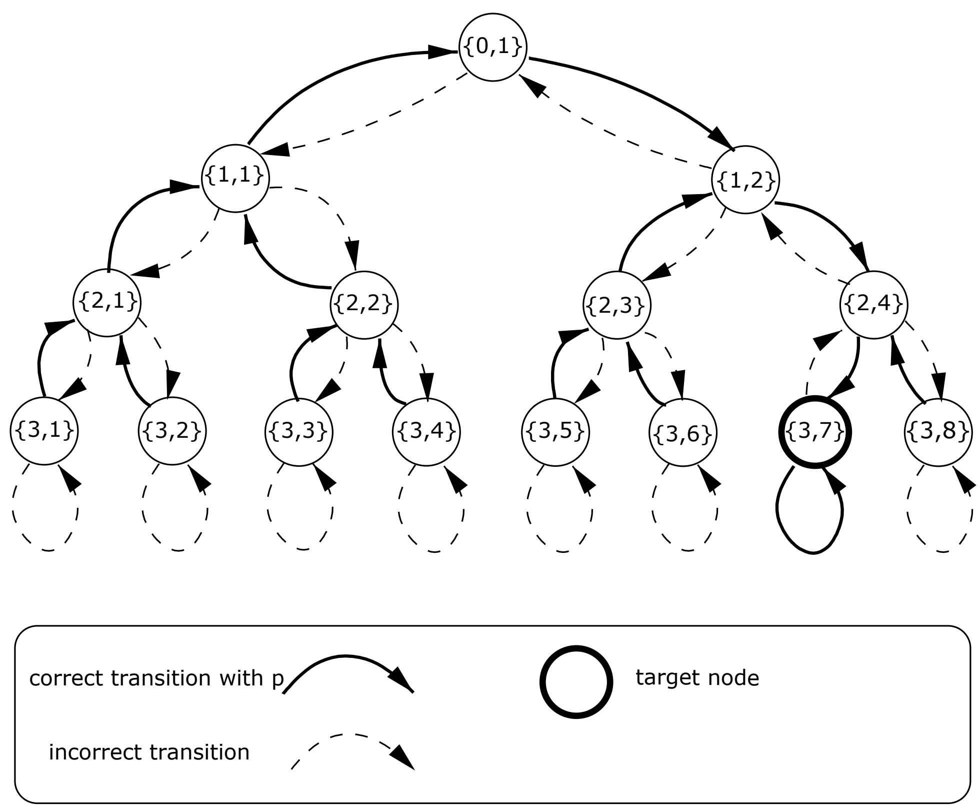

Figure 1: An example of a Markov chain for the RWJ in which the depth of the tree is 3.

The main idea of proof is to demonstrate that the limiting probability increases geometrically with the state indexes along the shortest path to the target node. As a consequence of this, the limiting probability can be shown to be concentrated within an arbitrarily small interval around the target node, simply by increasing the size of the tree.

Let j_D^* be the relative index of the target node among the leaf nodes located at level D.

Let $Q_{\{0,1\},\{D,j_D^*\}}$ denote the shortest path in the balanced tree from the root node $S_{\{0,1\}}$ to the target node $S_{\{D,j_D^*\}}$.

$Q_{\{0,1\},\{D,j_D^*\}}$ is composed of the sequence of the indexes of the nodes leading to the target node $S_{\{D,j_D^*\}}$ when starting from the root $S_{\{0,1\}}$. Thus, $Q_{\{0,1\},\{D,j_D^*\}} = [\{0,1\}, \{1,j_1^*\}, \ldots, \{l,j_l^*\}, \ldots, \{D,j_D^*\}]$, where j_d^* is the relative index of the node at depth d that belongs to the optimal path $Q_{\{0,1\},\{D,j_D^*\}}$. Equivalently, $S_{\{l,j_d^*\}}$

is the node at depth l ($0 \leq l \leq D$) that is on the path that starts from the root node and that leads to the target node. Clearly, $j_0^* = 1$.

Let $S_{\{D,j\}}$ be a non-target leaf node. Also, let $Q_{\{D,j\},\{D,j_D^*\}}$ be the shortest path in the tree that connects the non-target node $S_{\{D,j\}}$ to the target node $S_{\{D,j_D^*\}}$. Following the tree structure of the search space, the path $Q_{\{D,j\},\{D,j_D^*\}}$ is clearly composed by the concatenation of the two following sub-paths:

- A sub-path that originates from the non-target node $S_{\{D,j\}}$ and that does not intersect with the path $Q_{\{0,1\},\{D,j_D^*\}}$;
- And a second sub-path that intersects with $Q_{\{0,1\},\{D,j_D^*\}}$ (or more exactly a sub-path of $Q_{\{0,1\},\{D,j_D^*\}}$). Let $S_{\{k,j_k^*\}}$ be the node whose index comes first in the ordered list of node indexes forming this sub-path (i.e., the head of the sequence).

Therefore, $Q_{\{D,j\},\{D,j_D^*\}}$ can be seen as the concatenation of $Q_{\{D,j\},\{k,j_k^*\}}$ and $Q_{\{k,j_k^*\},\{D,j_D^*\}}$.

Informally speaking, moving along $Q_{\{D,j\},\{D,j_D^*\}}$ involves performing a series of *reverse transitions* from the non-target leaf node to the first node who is the root of the subtree containing the target node, and then performing *top-down transitions* in the direction of the target leaf node until the target node is attained.

Let us first study the transitions along the "reverse" path $Q_{\{D,j\},\{k,j_k^*\}}$. Examining the balance (equilibrium) equation of the MC at the the non-target leaf node $S_{\{D,j\}}$ gives:

$$\pi_{\{D-1,\lceil j/2\rceil\}} = \frac{p_{\{D,j\},\{D-1,\lceil j/2\rceil\}}}{p_{\{D-1,\lceil j/2\rceil\},\{D,j\}}}\pi_{\{D,j\}}. \tag{4}$$

We now observe that the Markov Chain is *time reversible* [21]. We thus resort to the time reversibility property in order to deduce the following equation:

$$\pi_{\{d-1,\lceil j/2\rceil\}} = \frac{p_{\{d,j\},\{d-1,\lceil j/2\rceil\}}}{p_{\{d-1,\lceil j/2\rceil\},\{d,j\}}}\pi_{\{d,j\}}, \tag{5}$$

where d denotes any given level in the tree such that $k \leq d \leq D$, implying that:

$$\pi_{\{d-1,\lceil j/2\rceil\}} = \frac{p}{p_{\{d-1,\lceil j/2\rceil\},\{d,j\}}}\pi_{\{d,j\}}. \tag{6}$$

It is easy to note that $\frac{p}{p_{\{D-1,\lceil j/2\rceil\},\{D,j\}}} < 1$ since $p_{\{D-1,\lceil j/2\rceil\},\{D,j\}} > p$.

Similarly, we consider the transitions along the path $Q_{\{k,i_k^*\},\{D,j_D^*\}}$. Examining the balance (equilibrium) equation of the Markov Chain at the the non-target node $S_{\{D,j_D^*\}}$ gives:

$$\pi_{\{D,j_D^*\}} = \frac{p_{\{D-1,\lceil j_D^*/2\rceil\},\{D,j_D^*\}}}{p_{\{D,j_D^*\},\{D-1,\lceil j_D^*/2\rceil\}}}\pi_{\{D-1,\lceil j_D^*/2\rceil\}}. \tag{7}$$

$$\pi_{\{D,j_D^*\}} = \frac{p}{p_{\{D,j_D^*\},\{D-1,\lceil j_D^*/2\rceil\}}}\pi_{\{D-1,\lceil j_D^*/2\rceil\}}. \tag{8}$$

By making use of the time reversibility property of the Markov Chain, we can easily deduce that along the "top-down" path $Q_{\{k,i_k^*\},\{D,i_D^*\}}$, for $k \leq d \leq D$:

$$\pi_{\{d,j_d^*\}} = \frac{p}{p_{\{d,j_d^*\},\{d-1,\lceil j_d^*/2\rceil\}}}\pi_{\{d-1,\lceil j_d^*/2\rceil\}}. \quad (9)$$

We define e_1 as $Min(\frac{p}{p_{\{d-1,\lceil j/2\rceil\},\{d,j\}}})$ for indexes $\{d,j\}$ such that nodes $S_{\{d,j\}}$ are along the "reverse" path $Q_{\{D,j\},\{k,j_k^*\}}$. Clearly, the latter quantity, e_1, is always greater than 1.

In an analogous manner, we denote e_2 to be the quantity $e_2 = Min(\frac{p}{p_{\{d,j_d^*\},\{d-1,\lceil j_d^*/2\rceil\}}})$ for indexes $\{d,j\}$ such that the nodes $S_{\{d,j\}}$ are along the "top-down" path $Q_{\{k,i_k^*\},\{D,i_D^*\}}$, for $k \le d \le D$. Then, $e_2 > 1$.

With these balance (equilibrium) equations in place, we are ready to deduce the relationship that relates the stationary probability of the target node to the stationary probability of any non-target node. Combining Eq. (6) and Eq. (9), and applying the reasoning behind the recurrence relationships we see that:

$$\pi_{\{D,j_D^*\}} > e^{D-k}\pi_{\{D,j\}},$$

where $e = e_1e_2$. Note that since we have $e_1, e_2 > 1$, then $e > 1$.

To finalize the proof, we consider the steady state of the Markov Chain for for any finite depth D. To do this, we invoke arguments similar to those used in [26]. Since $e > 1$, the limiting probabilities Π increases geometrically with the state indexes, along the shortest path to the target node, until its reaches its maximum at $\pi_{\{D,j_D^*\}}$. Since e^D increases exponentially and we are speaking about the mean of an increasing geometric progression, most of the mass will be concentrated among an arbitrarily small number of states close to the target node $S_{\{D,j_D^*\}}$.

Thus, as D goes to infinity the steady state will be centered around $\pi_{\{D,j_D^*\}}$ and will thus be be arbitrarily close to 1. Hence the theorem. □

Theorem 3. *We suppose $p > 0.5$. Let the steady probability of the Markov Chain be:*

$$\Pi = [\pi_{\{0,1\}}, \pi_{\{1,1\}}, \pi_{\{1,2\}}, \ldots, \pi_{\{D,1\}}, \pi_{\{D,2\}}, \ldots, \pi_{\{D,2^D\}}]. \quad (10)$$

Then, $Max(\Pi) = \pi_{\{D,j_D^*\}}$, where $Max(.)$ is the max operator applied to a vector and that returns the maximum element with tie breaks[3].

Proof: The proof of the theorem is straightforward and is already implicitly proven in the proof of Theorem 2. In fact we have already expressed this in the previous proof as since we know that the limiting probabilities Π *increases geometrically* with the state indexes along the shortest path to the target node, until its reaches its maximum at $\pi_{\{D,j_D^*\}}$. □

4 Simulation Results

In this section, we report some representative experimental results submitted to confirm the validity of the theoretical results we have obtained.

[3] In this case, the maximum is unique.

Since the convergence is asymptotic as the number of levels D tends to infinity, and due to the consequential explosion of the memory requirements, we were not able to perform simulations for large values of the tree depth D, i.e., those that can be considered to be large enough to be seen to tend to "infinity". In order to counter this limitation, we defined a neighborhood $N^*(i)$ as the i^{th} neighborhood of the target node which simply means all nodes that are located at the subtree rooted at node $\pi_{\{D-i,j^*_{D-i}\}}$. Observe that the target node belongs to subtree rooted at node $S_{\{D-i,j^*_{D-i}\}}$, and that the latter subtree is composed of i levels. Thus, we have, instead, reported the steady state probability of the neighborhood $N^*(i)$ which, informally, is the sum of the steady state probabilities of the nodes in $N^*(i)$, i.e., belonging to the subtree rooted at node $S_{\{D-i,j^*_{D-i}\}}$.

4.1 Empirical Verification of the Optimality

In this section, we report the steady probability in $N^*(3)$ using simulation, by running the random walk for 10^7 iterations. We varied the number of nodes in the tree D from 8 to 12, and varied the effectiveness of the environment p using the following three values: $p = 0.7$, $p = 0.85$ and $p = 0.95$.

In intermediate nodes, the RW is always faced with three transition alternatives (parent, right child or left child). The Oracle/Environment suggests the correct transition with the probability p. To further enrich the experiments, we performed experiments with different ways to "unevenly" distribute the remaining probability $1-p$ among the two incorrect transitions at the intermediate nodes. However, we discovered that distributing this probability mass "evenly" or "unevenly" (for example, with probabilities $\frac{1}{3}$ and $\frac{2}{3}$) had no effect on the convergence results[4]. We thus chose to "evenly" divide the remaining probability $1-p$ in between the two incorrect transitions, i.e., each incorrect transition occurs with the probability $\frac{1-p}{2}$ in the intermediate nodes. More specifically, if we suppose, for example, that at an intermediate node, the correct search direction is a transition to the parent, i.e., to do a backtracking operation, then the transition to the left child and right child will both takes place with probability $\frac{1-p}{2}$, while the transition to the parent will occur with probability p.

In all the experiments that we report the results for, we chose the target node to be the rightmost node in the hierarchy, i.e, node $S_{\{D,2^D\}}$. As expected, from Table 1, we observe that the values of the steady state probability in $N^*(3)$ approached 1. The latter value increased with p (which represented the quality of the Oracle/Environment, and it attained its maximum values for $p = 0.95$. In fact, for $p = 0.95$, the steady probability was greater than 0.999 for all the different values of D. Similarly, for $p = 0.85$, the steady probability exceeded 0.99 for all the different values of D.

[4]This is also in line with the theoretical results where we did not impose any condition on distributing $1-p$. Thus, for the sake of brevity, we will not report those experiments with "uneven" distributions of $1-p$ over the incorrect alternatives, and limit the reported results to the case of "even" distributions, i.e., $\frac{1-p}{2}$.

D	$p = 0.7$	$p = 0.85$	$p = 0.95$
8	0.9530518	0.9922004	0.9993106
9	0.9527884	0.9922102	0.9993274
10	0.9532058	0.9920532	0.9993156
11	0.9526474	0.9922458	0.9992978
12	0.9529796	0.9921286	0.9999534

Table 1: Convergence to the neighborhood $N^*(3)$ for various values of D and p.

In order to observe the rate of convergence by which the RW centers around the target node, we resorted to running an ensemble of $1,000$ experiments so as to observe the chain's transient behavior. In the following four figures, Figures 2, 3, 4 and 5, we report the evolution of the steady state probability of $N^*(3)$ for the values of $p = 0.6$, $p = 0.7$, $p = 0.8$, $p = 0.9$ respectively. In every experiment, we used 200 iterations, since we observed that the convergence in all the cases occurred in less than 200 iterations. The results from the figures display the speed with which fast our scheme converges. In fact, we observe from Figure 5 that the convergence took place before the 20 iterations transpired, and this was within a neighborhood of 0.98 for a value of $p = 0.9$. However, as illustrated in Figure (2) for a value of p as small as 0.6, the scheme converged slower. This is, of course, understandable because of the uncertain behavior of the Oracle/Environment. In this case, it took around 140 iterations for the scheme to converge.

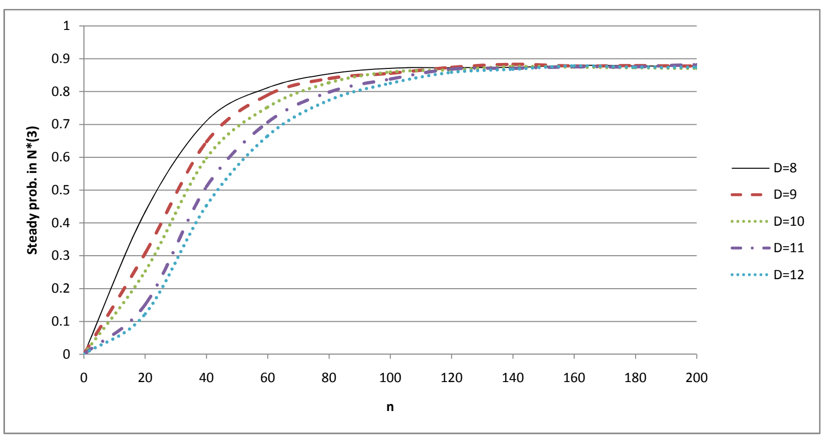

Figure 2: Plot of the evolution of the steady probability in the neighborhood $N^*(3)$ as a function of time, n, for $p = 0.6$ and for different values of D.

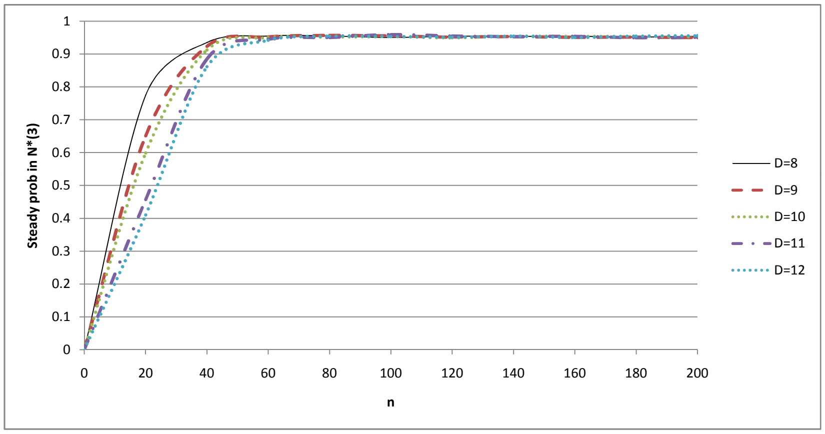

Figure 3: Plot of the evolution of the steady probability in the neighborhood $N^*(3)$ as a function of time, n, for $p = 0.7$ and for different values of D.

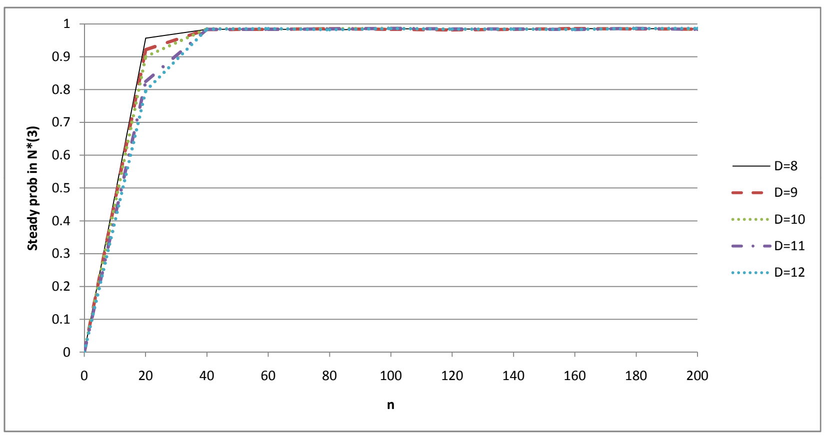

Figure 4: Plot of the evolution of the steady probability in the neighborhood $N^*(3)$ as a function of time, n, for $p = 0.8$ and for different values of D.

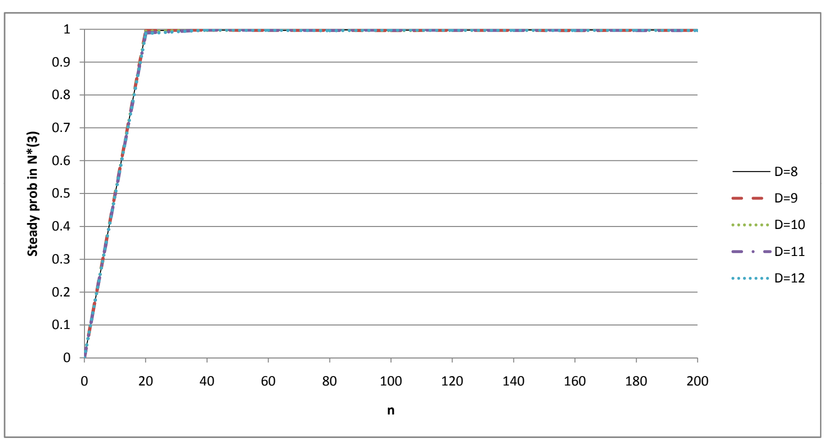

Figure 5: Plot of the evolution of the steady probability in the neighborhood $N^*(3)$ as a function of time, n, for $p = 0.9$ and for different values of D.

5 Conclusions

This paper concerns the field of Random Walks (RWs), typically used to model birth and death processes, renewal processes and the gambler's ruin problem. RWs have been studied for more than a century, and they have been primarily analyzed for walks on a *line*. RWs for two and three dimensions, have been modeled by enforcing the random steps to be made to the corresponding neighboring positions in one or multiple dimensions. RWs typically operate "on a discretized line" by forcing the walker to perform small steps to the current-state's neighbor states, rendering the analysis tractable. When any of the transitions are to non-neighbour states (referred to as "jumps " as opposed to steps), a formal analysis of the RW is, typically, untractable, except in a very few cases such as the one examined in [2]. This paper concerns the case when the jumps are performed as though a tree is superimposed on the line, and the jumps are to the children or the parent of the node where the walker is.

RWs on a tree entail moving to *non-neighbor* states in the space, which makes the analysis involved, and in many cases, impossible. However, in this case, we demonstrate that an analysis of the chain is feasible because we can invoke the phenomenon of "time reversibility". We, however, permit the operation of "backtracking", and interestingly, it is precisely these "backtracking" transitions that naturally render the chain to be "time reversible". Our analysis is possible because we have used the latter property of the chain. Further, this RW turns out to be a type of generalization of dichotomous search with faulty feedback, which has numerous real-life applications.

The paper has derived the formal analysis of the chain, and has also listed its fascinating limiting theoretical properties. The paper has also presented simulations that justify the chain's analytic steady-state and transient characteristics.

References

[1] W. Feller, *An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition*. Wiley, 3 ed., January 1968.

[2] A. Yazidi, O.-C. Granmo, and B. J. Oommen, "On the analysis of a random interleaving walk–jump process with applications to testing," *Sequential Analysis*, vol. 30, no. 4, pp. 457–478, 2011.

[3] K. Pearson, "The Problem of the Random Walk," *Nature*, vol. 72, p. 342, August 1905.

[4] H. C. Berg, *Random Walks in Biology*. Princeton University Press, Revised ed., September 1993.

[5] M. A. Nowak, *Evolutionary Dynamics: Exploring the Equations of Life*. Belknap Press of Harvard University Press, September 2006.

[6] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory (Wiley Series in Probability and Statistics)*. Wiley-Interscience, February 1998.

[7] L. Takacs, "On the classical ruin problems," *Journal of the American Statistical Association*, vol. 64, no. 327, pp. pp. 889–906, 1969.

[8] J. Paulsen, "Ruin theory with compounding assets – a survey," *nsurance: Mathematics and Economics*, vol. 22, no. 1, pp. 3 – 16, 1998. Special issue on the interplay between insurance, finance and control.

[9] G. H. Bower, "A turning point in mathematical learning theory," *Psychological Review*, vol. 101, no. 2, pp. 290 – 300, 1994.

[10] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002.

[11] F. Fouss, P. A., J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 3, pp. 355–369, 2007.

[12] A. Altman and M. Tennenholtz, "Ranking systems: the pagerank axioms," in *EC '05: Proceedings of the 6th ACM conference on Electronic commerce*, (New York, NY, USA), pp. 1–8, ACM, 2005.

[13] P. G. Bishop and F. D. Pullen, "A random walk through software reliability theory," *Mathematical structures for software engineering*, pp. 83–111, 1991.

[14] B. J. Oommen, "Absorbing and ergodic discretized two-action learning automata," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-16, pp. 282–293, March/April 1986.

[15] W. Jiang, C.-L. Zhao, S.-H. Li, and L. Chen, "A new learning automata based approach for online tracking of event patterns," *Neurocomputing*, vol. 137, pp. 205–211, 2014.

[16] A. Yazidi, O.-C. Granmo, and B. J. Oommen, "Learning-automaton-based online discovery and tracking of spatiotemporal event patterns," *Cybernetics, IEEE Transactions on*, vol. 43, no. 3, pp. 1118–1130, 2013.

[17] M. Ben Or and A. Hassidim, "The bayesian learner is optimal for noisy binary search (and pretty good for quantum as well)," in *IEEE 49th Annual IEEE Symposium on Foundations of Computer Science, 2008. FOCS'08.*, pp. 221–230, IEEE, 2008.

[18] J. Zhang, Y. Wang, C. Wang, and M. Zhou, "Symmetrical hierarchical stochastic searching on the line in informative and deceptive environments," *To appear in IEEE Transactions on Cybernetics*, 2016.

[19] B. J. Oommen, "Stochastic searching on the line and its applications to parameter learning in nonlinear optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 27, no. 4, pp. 733–739, 1997.

[20] A. Yazidi, O.-C. Granmo, B. J. Oommen, and M. Goodwin, "A novel strategy for solving the stochastic point location problem using a hierarchical searching scheme," *Cybernetics, IEEE Transactions on*, vol. 44, no. 11, pp. 2202–2220, 2014.

[21] F. Kelly, *Reversibility and stochastic networks*. Wiley series in probability and mathematical statistics. Tracts on probability and statistics, J. Wiley, 1987.

[22] S. Karlin and H. Taylot, *A First Course in Stochastic Processes*. Academic Press, 1975.

[23] S. Ross, *Introduction to Probability Models*. Academic Press, 1980.

[24] B. Oommen and J. Dong, "Generalized swap-with-parent schemes for self- organizing sequential linear lists," in *Proceedings of ISAAC'97, the 1997 International Symposium on Algorithms and Computation*, (Singapore), pp. 414–423, December 1997.

[25] O.-C. Granmo and B. J. Oommen, "Solving stochastic nonlinear resource allocation problems using a hierarchy of twofold resource allocation automata," *IEEE Transactions on Computers*, vol. 59, pp. 545–560, 2009.

[26] B. J. Oommen, "Stochastic searching on the line and its applications to parameter learning in nonlinear optimization," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-27B, pp. 733–739, 1997.