

Article

Not peer-reviewed version

---

# Analysis of the Reliability and Efficiency of Information Extraction Using AI-Based Chatbot: The More for Less Paradox

---

[Eugene Levner](#)<sup>\*</sup> and Boris Kriheli

Posted Date: 19 March 2025

doi: 10.20944/preprints202503.1427.v1

Keywords: information search; stochastic search policy; artificial intelligence (AI) chatbot; chatbot reliability; search efficiency; more-for-less paradox



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

*Article*

# Analysis of the Reliability and Efficiency of Information Extraction Using AI-Based Chatbot: The More for Less Paradox

Eugene Levner <sup>1,\*</sup> and Boris Kriheli <sup>2</sup>

<sup>1</sup> School of Computer Science, Faculty of Sciences, Holon Institute of Technology, Holon, Israel

<sup>2</sup> Department of Computer Science, Ashkelon Academic College, Ashkelon, Israel; borisk@hit.ac.il

\* Correspondence: levner@hit.ac.il

**Abstract:** This paper addresses the following problem of information extraction using an AI-powered chatbot. An AI chatbot processes a natural language conversation with a human user using semantic search and human-chatbot interaction to extract the most relevant and contextually appropriate information from available databases. During a natural language conversation between a human and an AI, the human user can modify and refine queries until he/she is satisfied with the chatbot's output. The reliability of an AI chatbot is defined as its ability to understand user queries and provide correct answers; in the current study, it is measured by the frequency (probability) of correct answers. The search efficiency of a chatbot indicates how accurate and relevant the information returned by the chatbot is; in this work, it is measured by the satisfaction level that the human user receives for a correct answer. We uncover a counterintuitive relationship between AI chatbot reliability and search efficiency: we demonstrate that, under fairly general conditions, a less reliable AI chatbot can have a higher expected search efficiency than a more reliable chatbot. This phenomenon aligns with a family of "more-for-less" paradoxes observed in various complex systems. Finally, we discuss the underlying mechanism of this paradox.

**Keywords:** information search; stochastic search policy; artificial intelligence (AI) chatbot; chatbot reliability; search efficiency; more-for-less paradox

## 1. Introduction

This paper addresses the problem of information extraction by an AI-powered chatbot that is not entirely reliable and may make mistakes in its responses. Unlike conventional search-and-rescue robots, which typically perform targeted searches using well-known pathfinding algorithms to autonomously navigate physical space, the AI chatbot processes a natural language conversation with a human user and extracts meaningful information using semantic search and human-AI interaction to generate the most relevant and contextually appropriate responses. During the user-AI natural language conversation, the human user can modify and refine queries until he/she is satisfied with the chatbot's output.

We analyse two key characteristics of human-AI interaction, reliability and search efficiency [1–3]. The reliability of an AI-based chatbot is reflected in its ability to understand user queries and provide correct answers; it is measured by the frequency (probability) of correct answers [1,2]. The search efficiency of a chatbot [3] shows how accurate and relevant the information returned by the chatbot is. In this study, search efficiency is measured by the level of satisfaction that a human user receives for a correct answer from the chatbot on a 10-point Likert scale. This is a well-known IBM SPSS rating system for measuring customer satisfaction, in which the highest score of 1.0 signifies "very satisfied" on a 10-point scale ranging from 0.1 to 1.0, while the lowest score of 0.1 means "very dissatisfied" [4].

In this work, we uncover a counterintuitive relationship between the reliability of AI chatbots and their search efficiency. We demonstrate that, under fairly general conditions, reducing the reliability of deployed chatbots can improve the search efficiency. In particular, a less reliable chatbot that stops searching after extracting relevant information may have higher expected efficiency than a more reliable chatbot. This phenomenon aligns with the family of “more-for-less” paradoxes observed in various complex systems. Finally, we discuss the underlying mechanism of this paradox.

The main contributions of this paper are as follows: (i) we present and formalize a multi-stage (multi-round) conversation process between a human user and AI chatbot, such as ChatGPT, which serves to extract relevant information and consists of multiple rounds of query modification/refinement during the conversation; (ii) the optimization information-extraction problem for the imperfect AI chatbot is formulated as a sequential discrete search problem with probabilistic parameters; (iii) a new efficient index-based algorithm for determining the optimal search policy is developed, which extends the known fast scheduling algorithms of Ross [5], Sweat [6] and Chew [7] for sequential discrete search problems; and finally (iv) the counterintuitive relationship between the AI chatbot reliability and search efficiency characteristics in the multi-round informational search process is analyzed and explained.

The remainder of the paper is organized as follows. In the next section, we introduce the main definitions that will be used in the sequel. Section 3 describes the information extraction problem to be solved. Section 4 provides a brief overview of the previous work. In Section 5, we propose a mathematical model and an efficient index-based algorithm to optimize the expected user satisfaction over an infinite horizon. In Section 6 we formulate and prove the paradoxical relationship between the reliability and efficiency of an optimal information extraction process using an AI chatbot. Numerical examples to support and illustrate the studied paradox are presented in Section 7. A summary of contributions and an outline of future research are provided in Section 8.

## 2. Preliminaries

In this section we provide the basic definitions that will be used in the sequel.

### 2.1. The Discrete Search Problems with Imperfect Robot's Sensors

The common *discrete search problem* is as follows [5–10]. A target is hidden in one of the known locations  $1, 2, \dots, N$  with a probability distribution  $p = (p_1, p_2, \dots, p_N)$ ,  $\sum_i p_i = 1$ , and remains in this location during the search. A search device (for example, the search-and-rescue robot) must perform a sequence of inspections at locations  $1, 2, \dots, N$  to find the target, and continues searching until the target is found.

The robot's sensor is *imperfect*, that is, for each location  $i$ , it has a non-zero overlook probability  $a_i$  associated with it, so that if the target is at location  $i$  and the robot is exploring that location, there is a chance (probability  $a_i > 0$ ) that the sensor will not detect the target for any  $i = 1, \dots, N$ .

Given the above probabilities, along with the duration of the inspections and the reward for detecting the target at a given location, the goal is to define a probabilistic search policy that finds the optimal sequence of robot movements that maximizes the expected total reward over an infinite time interval. Specific details regarding information search/extraction using AI-powered chatbots are presented in the next subsection.

### 2.2. The Information Extraction Problem Using AI Chatbots

The studied problem refers to searching and extracting relevant information from a large dataset in response to a user's query. The goal of an AI chatbot is to provide the most useful and accurate information and avoid irrelevant results.

While the task of information extraction using AI chatbots can be considered a special class of the general discrete search problem, these problems differ in their goals, physical environments, and solution methods. Search-and-rescue robots perform searches, typically after disasters, in physical

areas seeking to find and rescue survivors, while AI-powered chatbots perform information search and extraction aiming to find, extract, and generate relevant textual and visual answers from digital knowledge bases, documents, and the web. Search-and-rescue robots use discrete pathfinding algorithms such as A\*, PRM, and Dijkstra's algorithm to autonomously navigate physical spaces, while the conversation between a human user and an AI chatbot consists of multiple stages (rounds) during which the user modifies/refines queries while the chatbot improves its responses over time based on user feedback and learning from the user's interactions with the AI, until the user is satisfied or until a designated dialogue time interval has elapsed.

An AI-powered chatbot searches, discovers and extracts the most relevant information requested through a process called semantic search. This is an advanced information retrieval technology that focuses on understanding the meaning and content of a user's query, rather than simply searching and matching keywords. The chatbot typically scans hundreds of databases and their clusters depending on the complexity of the query, where thousands of documents can be scanned in milliseconds, making this process efficient. The process consists of the following main steps:

- query understanding, where the chatbot interprets the human user's request using natural language processing to extract the meaning and then convert the text into a numerical format,
- using semantic search, the chatbot compares the query with data stored in databases or multiple database clusters to find the most relevant matches,
- the chatbot performs ranking/filtering of the query to prioritize the most relevant pieces of information and refine its response, and finally
- once the relevant information has been found, the robot generates text using natural language and extracts the natural language response.

In the next section we present a mathematical model of the problem.

### 3. Description of the Efficiency-Maximizing AI-Based Search Problem

The target information is hidden in one of the known database clusters (called locations)  $1, 2, \dots, N$  with a known probability distribution  $\mathbf{p} = (p_1, p_2, \dots, p_N)$ ,  $\sum_i p_i = 1$ , and remains in this location during the search. The search device (in our case, the AI powered chatbot) must perform a sequence of scans in locations  $1, 2, \dots, N$  to find the target information, and continues the multi-round search until a human user declares on some round that the target has been found, or until a prespecified search time interval has elapsed; on both cases the search process stops.

The chatbot is *imperfect* (not completely reliable), i.e., at each stage, for each location  $i$ , it has a non-zero overlook probability  $a_i$ , so that if the requested target information is in location  $i$  and the chatbot scans this location, it does not recognize it and does not report the presence of the target information there,  $i = 1, \dots, N$ .

Given the above probabilities  $\mathbf{p} = (p_1, p_2, \dots, p_N)$  and  $\mathbf{a} = (a_1, a_2, \dots, a_N)$ , as well as the duration of all conversation rounds and the user satisfaction levels with the extracted target information in each conversation round, the objective of the information extraction problem under consideration is to determine a stochastic search policy to find the optimal sequence of chatbot location scans that maximizes the expected user satisfaction over an infinite time horizon. In Table 1, we highlight the main differences between the general discrete search and information extraction problems. The essential difference between them is that the present work is motivated by and focused on the practical issue of analyzing the properties of existing AI chatbots.

**Table 1.** Comparison of two problems.

<i>Problem</i> <i>Feature</i>	<i>Discrete search problem</i>	<i>AI powered chatbot information extraction problem</i>
Objective function	Expected (discounted) reward over infinite time interval	Expected efficiency over infinite horizon
Stopping rule	Search stops at robot's step when the sensor reports that it has detected a hidden target	Information search stops when the human user is satisfied or when the designated dialogue time has expired
Decision to be found	Optimal infinite sequence of robot steps (inspections of the locations).	Optimal infinite sequence of chatbot scans (inspections of digital databases).
Optimal policy $\pi_t^*$	The robot's steps are ordered according to the "attractiveness indices" of locations	Chatbot's scans are ordered according to the "attractiveness indices" of locations, defined in Section 5.

Consider the mathematical model in more detail. The discrete search area  $A$  contains  $N$  known database clusters (locations), in one of which a target information is hidden, with a known probability distribution  $p = (p_1, p_2, \dots, p_N)$ ,  $\sum_i p_i = 1$ . The AI chatbot inspects the area to find and identify a hidden target whose location is unknown in advance; we assume that this smart robot must sequentially perform location scans in each round, one after another. The AI chatbot is imperfect, this means that associated with each location  $i$  is a non-zero *overlook probability*  $a_i$ , that is,  $a_i$  is the conditional probability that the robot will not detect the target information in a single scan of location  $i$ , given that this location in fact contains the requested information,  $i \in A = \{1, \dots, N\}$ . Therefore, since  $a_i \neq 0$ , for the search to be successful, each location must be searched infinitely often, and, from the mathematical point of view, in general, the sequence of user-chatbot conversational rounds can be infinite.

The duration of the user's query modifications at each conversation round are assumed to be known in advance. Also, as we indicated above, in this work the search efficiency is measured by the satisfaction level that a human user feels whenever he/she is satisfied by the chatbot response during some stage (round) of the conversation; it is assumed to decrease monotonically in time during the user's conversation with the chatbot.

The goal of information extraction is to determine the optimal search strategy to find the target information with the maximum expected search efficiency (user's satisfaction) over an infinite time horizon.

In addition to probabilities  $p_i$  and  $a_i$  defined above, we use the following notation:

- policy  $\pi$  is an infinite sequence of conversation stages (rounds),  $\pi = (\pi_1, \pi_2, \dots)$ ;  $\pi_k$  is the number of a location (a database cluster) scanned at the  $k$ th round of the sequence  $\pi$ ,  $\pi_k \in A = \{1, \dots, N\}$ ,  $k = 1, 2, \dots$ ;
- $n$  is the (unknown) round of the sequence  $\pi$  at which the chatbot successfully extracts the targeted information;
- $u_i = u(\pi, n)$  is the number of unsuccessful scans (rounds) of location  $i = \pi_n$ , before the  $n$ th (successful) round of the sequence  $\pi$ ;  $u(\pi, n) \leq n-1$ ;



- $P_{\pi_n}$  is the conditional probability that after  $u(\pi, n)$  unsuccessful scans of location  $\pi_n$ , the chatbot will successfully detects that location (database cluster)  $\pi_n$  contains the target information, given that the latter information is at location  $\pi_n$ ;  $P_{\pi_n} = a_{\pi_n}^{u(\pi, n)}(1 - a_{\pi_n})$ ;
- $\Pi_{\pi_n} = P_{\pi_n}p_{\pi_n}$  is the unconditional probability that the chatbot will find the requested information at location  $\pi_n$ ;
- $t_k$  is the known time required for a human user to read, evaluate and refine the chatbot's response in round  $k$ ,  $k=1, 2, \dots$ ;
- $T_{\pi, n} = \sum_{i=1}^n t_{\pi_i}$ , the total time spent by a human user in conversation with the chatbot from round 1 to round  $n$  in sequence  $\pi$ ;
- $r(\pi) = r_{\pi_n}$  is the known Likert-scale satisfaction level obtained by the human user when the target information is found at location  $\pi_n$  of the sequence  $\pi$  (before discounting). If the search for the target information requires  $T_{\pi, n}$  time units, the level of satisfaction is discounted by  $e^{-dT_{\pi, n}}$  for a known discount factor  $d$ ,  $d \in (0, 1)$ , i.e., is defined as follows:  $R_{discount}(\pi) = r_{\pi_n} e^{-dT_{\pi, n}} = r_{\pi_n} e^{-d \sum_{i=1}^n t_{\pi_i}}$ .

In the problem under consideration, the satisfaction level is a random variable with value  $x = r_{\pi_n} e^{-d \sum_{i=1}^n t_{\pi_i}}$  and probability  $p(x) = P_{\pi_n} p_{\pi_n} = a_{\pi_n}^{u(\pi, n)}(1 - a_{\pi_n}) p_{\pi_n}$  depending on  $\pi$ . Then the expected discounted satisfaction (search efficiency) corresponds to the expected accumulated information obtained by the human user in the policy  $\pi$ , denoted  $F(\pi)$ , and will be defined over an infinite time horizon as follows:

$$F(\pi) = E(R_{discount}(\pi)) = E(r_{\pi_n} e^{-d \sum_{i=1}^n t_{\pi_i}}) = \sum_{n=1}^{\infty} r_{\pi_n} \cdot e^{-d \sum_{i=1}^n t_{\pi_i}} \cdot P_{\pi_n} p_{\pi_n} = \sum_{n=1}^{\infty} r_{\pi_n} \cdot e^{-d \sum_{i=1}^n t_{\pi_i}} \cdot a_{\pi_n}^{u(\pi, n)}(1 - a_{\pi_n}) p_{\pi_n} \cdot (1)$$

The goal is to define a stochastic policy for a search chatbot that determines where it should search at each round, that is, determines an optimal infinite sequence of chatbot actions that maximizes the expected discounted satisfaction of a human user  $F(\pi)$  over an infinite period of time.

## 4. Previous Work

This section is devoted to two different topics of this study: a review of efficient discrete search algorithms and a brief overview of paradoxical phenomena arising in operations research.

### 4.1. Index-Based Algorithms for the Discrete Search Problems

The discrete sequential search problem is one of the oldest problems of operations research, first formulated and solved by Bernard Koopman and his team during the World War II, when the military task was to provide effective methods of detecting submarines [11]. An excellent review of search theory and recent bibliography of the discrete search literature can be found in [8,12,13].

Several authors have studied sequential discrete search problems with overlooking probabilities in various settings and developed efficient index-based solution methods (see e.g., [5–10]). When a search device, such as a search-and-rescue robot or an unmanned aerial vehicle (drone), is not reliable, the inspection/search of a location is imperfect, and in this case any location may need to be inspected many, even infinite, times. Thus, the discrete search problem is to find an optimal strategy rather than the optimal search sequence. The latter optimal strategy depends on the random moment when the search device finds the target.

A typical stopping rule for a multi-step stochastic search process is: "Stop at a step when the target is found or when the assigned search time has elapsed". Other criteria for terminating the search process, called stopping rules, have been proposed by Ross [5] and Chew [7]. Trummel and Weisinger [14] and Wegener [15] showed that the general discrete search problem is NP-hard.

In a more recent study, Kriheli et al. [16] proposed a fast algorithm for searching for hidden targets by unmanned aerial vehicles under imperfect inspections with non-zero false-positive and false-negative detection probabilities.

Kress et al. [17] analysed several myopic and biology-inspired search strategies that minimize search time and have shown that a local index-based policy is optimal when target detection by an autonomous automated search device is accompanied by reliable verification work performed by a team of humans.

In this paper, we continue and extend the results mentioned in the previous work by focusing on the study of a multi-round search scenario in which imperfect inspections of clustered databases are performed by an AI-powered chatbot accompanied by a human user who verifies the chatbot's responses and refines his/her queries.

#### 4.2. Paradoxes in Operations Research

In teaching Operations Research and Artificial Intelligence courses, the authors have found that presenting counter-intuitive or paradox-like anomalies can be a great eye-opener for listeners. Studying such anomalies leads to a better understanding of algorithms and a deeper learning of OR and AI techniques. A typical example is the famous *Braess paradox* that arises in dynamic transportation networks and states that “adding a new wide road to a congested transport network can lead to an increase in overall travel time” [18,19]. Similar counterintuitive effects are known to also occur, for instance, in power transmission systems [20], biological and ecological systems [21], and exploiting generative Artificial Intelligence [22].

A mechanism of this paradox can be explained by changing drivers' priorities and behaviours after a network's reconstruction. For example, in response to a capacity addition more drivers begin to use the expanded highway; those previously not traveling at peak times shift to the peak hours, and public transport users shift to driving. From a mathematical perspective, such drivers' behaviour disrupts the equilibrium and leads to suboptimal traffic flow for everyone. This is because the Nash equilibrium of such a system is not necessarily optimal. The network change induces a new game structure between multiple drivers which leads to a new situation. Unlike a Nash equilibrium, where drivers have no incentive to change their routes, when the system is not in a Nash equilibrium, individual drivers can improve their respective travel times by changing the routes they take. It follows that drivers will continue to switch until they reach a new Nash equilibrium despite the reduction in overall performance.

Another phenomenon, called the *Jevons paradox*, describes a situation where technological progress increases the efficiency of resource use, reducing the amount required per user and lowering its cost. However, the falling cost of use increases demand. Consequently, while individual resource consumption decreases, total societal resource use by all users increases, regardless of whether this resource is fuel, or a computer, or an electronic device. So, this “paradox” is explained by the substantial societal increase in demand, which outweighs individual gains in resource efficiency, leading to a net increase (rather than decrease) of societal resource consumption [23,24]. These paradoxes warn against making decisions based solely on local or individual optimization without assessing the broader impact on the system.

One more paradox arising in scheduling theory is described by Spieksma and Woeginger [25] and states that “increasing the speed of some machines in a no-wait flow-shop problem can significantly worsen the optimal total time”. There are other, less well known anomalies studied in the literature, for example, so-called “more-for-less paradox” in the classical transportation problem [26] and Graham's multiprocessor scheduling anomaly [27].

The next sections present another paradoxical “more-for-less” phenomenon, according to which increasing the generative AI's reliability decreases its efficiency. Note that in all mentioned papers and in this one, the word “paradox” is used to denote a *counter-intuitive and surprising phenomenon*, not a classical paradox in the context of formal logic.

## 5. Finding the Optimal Policy $\pi^*$ for the Problem

In this section, we prove the optimality of the greedy index policy for the information extraction problem. We will use a basic method called the “interchange argument”, which involves exchanging two consecutive search locations. For completeness, we will describe this index policy for solving the problem under study. In what follows, we will use this policy to identify the paradoxical property of the problem. The idea of this method is as follows.

To find the optimal search strategy that maximizes the expected discounted satisfaction presented in (1), it suffices to compare the objective function  $F(\pi)$  for two conjugate infinite sequences  $\pi^1$  and  $\pi^2$ , in which exactly two arbitrary neighbouring locations  $\pi_n^1$  and  $\pi_{n+1}^1$  are swapped:

$$\begin{aligned}\pi^1 &= (\pi_1^1, \pi_2^1, \dots, \pi_{n-1}^1, \pi_n^1, \pi_{n+1}^1, \pi_n^1, \pi_{n+2}^1, \dots); \\ \pi^2 &= (\pi_1^1, \pi_2^1, \dots, \pi_{n-1}^1, \pi_n^2 = \pi_{n+1}^1, \pi_{n+1}^2 = \pi_n^1, \pi_{n+2}^2 = \pi_{n+2}^1, \dots).\end{aligned}$$

Having performed elementary arithmetic operations with expression (1), it is easy to obtain that

$$\begin{aligned}F(\pi^1) - F(\pi^2) &= \\ &[r_{\pi_n^1} \cdot e^{-dt_{\pi^1, n}} \cdot P_{\pi_n^1} p_{\pi_n^1} + r_{\pi_{n+1}^1} \cdot e^{-dt_{\pi^1, (n+1)}} \cdot P_{\pi_{n+1}^1} p_{\pi_{n+1}^1}] - \\ &[r_{\pi_{n+1}^1} \cdot e^{-d(T(\pi^1, (n+1)) - t_{\pi_n^1})} \cdot P_{\pi_{n+1}^1} p_{\pi_{n+1}^1} + r_{\pi_n^1} \cdot e^{-dt_{(\pi^1, n+1)}} \cdot P_{\pi_n^1} p_{\pi_n^1}] \quad (2)\end{aligned}$$

From (2) it follows that  $F(\pi^1) \geq F(\pi^2)$  iff

$$\frac{r_{\pi_n^1} \cdot e^{-dt_{\pi_n^1}} \cdot P_{\pi_n^1} p_{\pi_n^1}}{1 - e^{-dt_{\pi_n^1}}} \geq \frac{r_{\pi_{n+1}^1} \cdot e^{-dt_{\pi_{n+1}^1}} \cdot P_{\pi_{n+1}^1} p_{\pi_{n+1}^1}}{1 - e^{-dt_{\pi_{n+1}^1}}},$$

or

$$\frac{r_{\pi_n^1} \cdot e^{-dt_{\pi_n^1}} \cdot a_{\pi_n^1}^{u(\pi^1, n)} (1 - a_{\pi_n^1}) p_{\pi_n^1}}{1 - e^{-dt_{\pi_n^1}}} \geq \frac{r_{\pi_{n+1}^1} \cdot e^{-dt_{\pi_{n+1}^1}} \cdot a_{\pi_{n+1}^1}^{u(\pi^1, n+1)} (1 - a_{\pi_{n+1}^1}) p_{\pi_{n+1}^1}}{1 - e^{-dt_{\pi_{n+1}^1}}}. \quad (3)$$

Since neighbouring locations are chosen arbitrarily, in order for a sequence  $\pi^*$  to have a maximum expected efficiency  $F(\pi^*)$ , it is sufficient that all its components are chosen sequentially one after another in a non-decreasing order of the coefficients defined in (3). We come to the conclusion that the following two-index parameter  $Q_{i,j}$  characterizes the “attractiveness” of location  $i$ , given that this location has already been searched unsuccessfully  $j$  times,  $j \geq 1$ :

$$Q_{i,j} = \frac{r_i e^{-dt_i} \cdot a_i^j (1 - a_i) p_i}{1 - e^{-dt_i}}, \quad i=1, \dots, N; j=1, 2, \dots \quad (4)$$

Using relations (2)-(3) and notation (4), we obtain the following result:

**Proposition 1.** *If there have already been  $j = j(i)$  unsuccessful scans of each location  $i$ , where  $i \in \{1, \dots, N\}$  and  $j \geq 1$ , then choosing for the next scan the location  $i^*$  that has the maximum value of the parameter  $Q_{i,j}$  defined in (4), among all  $i$  for a fixed  $j = j(i)$ , ensures the maximum discounted satisfaction  $F(\pi)$  of the user.*

**Remark.** *The symbol  $u(\pi, n)$  in formula (3) denotes the same term as the symbol  $j$  in formula (4), i.e., the number of times a certain location  $i = \pi_n$  has already been unsuccessfully scanned by an unreliable chatbot. However, introducing such a double notation in this context is done for convenience, in order to write the formulas in a more readable form. The former notation is convenient when comparing two policies denoted  $\pi^1$  and  $\pi^2$ , while the latter will be more convenient below when describing the upcoming multi-round algorithm.*

### 5.1. Iterative Algorithm for Constructing the Optimal Sequence

Proposition 1 formulated above allows us to construct an efficient iterative algorithm for constructing an optimal search policy  $\pi^*$ .

The algorithm is defined for an indefinite number of search rounds over an infinite time horizon. It contains an *a priori* unknown number of rounds and continues until the human user is satisfied with the information obtained or until a predetermined search time has elapsed; in both cases, the user stops the search process. In essence, this procedure requires that at each round, for each subsequent search, exactly one known cluster of databases is selected and scanned, namely the cluster



having the maximum parameter  $Q_{ij}$  defined in (4) among the values of  $Q_{ij}$  that have not yet been selected.

The results of the calculations at any round  $k$ ,  $k = 1, 2, \dots$  will be represented as a  $k \times N$  matrix, denoted  $M_k$ , in which  $N$  columns correspond to predetermined locations for scanning and the number of rows corresponds to the number of search rounds performed before the algorithm stops. The elements  $m_{i,j}$  of the matrix  $M_k$  represent the parameters  $Q_{ij}$  from expression (4), computed sequentially round by round. Below we describe round 1 separately, since it differs from all subsequent rounds in that it contains an additional step designated 1.1, in which the first row of the matrix  $M_1$  is formed. After that we give a description of the current  $k$ th round, the same for all  $k \geq 2$ .

As mentioned, in this procedure, the subscript  $j$  denotes how many unsuccessful scans of location  $i$  ( $i=1, \dots, N$ ) were made by the chatbot before the start of round  $k$  ( $k=1, 2, \dots$ ). Therefore, this index can be provided with two indicators, namely  $i$  and  $k$ , and the subscript  $j$  will be replaced by the notation  $j_k(i)$ .

## 5.2. Rounds of the Algorithm

Let us first describe round 1, which consist of the following four steps.

Step 1.1: //Calculate the elements of the first row of the matrix  $M_1$  //

In accordance with expression (4), we set  $j = 1$  for all locations, calculate the parameters  $Q_{i,1}$ , for all  $i=1, \dots, N$ , and write them as the corresponding elements  $m_{i,1}$  of the first row of the matrix  $M_1$ .

Step 1.2. The chatbot performs the first (preliminary) scan of each location one by one and extracts the relevant information from each location. The chatbot being unreliable, the human user verifies that the target information from the scanned-once locations is not satisfiable yet and hence assign 1 to the number of unsuccessful checks for each location.

Step 1.3. //Determine the location  $i_1^*$  and scan it once more //

Choose location  $i_1^*$  such that  $Q_{i_1^*,1} = \max_i Q_{i,1}$ . Scan location  $i_1^* = \pi_1^*$  (for the second time). If the human user is satisfied with the target information extracted now from location  $\pi_1^* = i_1^*$ , compute the partial satisfaction level  $F(\pi^*)$  obtained so far, where  $\pi^* = \{\pi_1^*\}$  and terminate the search; highlight  $Q_{i_1^*,1}$  in cell  $(1, i_1^*)$  in bold and red (see numerical example in section 7). Otherwise, increase the number of unsuccessful scans for location  $i_1^*$  by 1, i.e., set  $j_2(i_1^*) := j_1(i_1^*) + 1 = 2$ . Go to the next round 2.

Let us now assume that the rounds  $1, \dots, k$  ( $k \geq 1$ ) are completed and the locations  $\pi_1^*, \dots, \pi_k^*$  are determined at these rounds, i.e., the optimal subsequence  $\pi_k^* = (\pi_1^*, \pi_2^*, \dots, \pi_k^*)$  is found. At the end of the  $k$ th round, all parameters  $Q_{i,j}$ ,  $i=1, \dots, N$ ,  $j=1, 2, \dots, k+1$  are calculated and represented as a known  $(k+1) \times N$ -matrix  $M_{k+1}$ . Accordingly, the corresponding indices  $j_1(i^*)$ ,  $j_2(i_2^*)$ ,  $\dots$ ,  $j_k(i_k^*)$ , and the objective function  $F(\pi^*)$  are calculated. Then the next,  $(k+1)$ th, round will be as follows.

Round  $k+1$ , where  $k = 1, 2, \dots$ , consists of the following three steps.

Step  $k+1.1$ . // Determine the location  $i_{k+1}^*$  to be scanned  $(k+1)$ th-in-turn by the iterative algorithm. //

Denote the second index  $j$  in the elements of the  $(k+1)$ th row of the matrix  $M_{k+1}$  by  $j_{k+1}(i)$  for all  $i$ ; among all parameters  $Q_{i,j_{k+1}(i)}$  select the location  $i_{k+1}^*$  (to be scanned as the next one) which has the maximum  $Q_{i,j_{k+1}(i)}$  as follows:  $Q_{i_{k+1}^*,j_{k+1}} = \max_{i=1, \dots, N} Q_{i,j_{k+1}(i)}$ . Place  $Q_{i_{k+1}^*,j_{k+1}}$  in cell  $(k+1, i_{k+1}^*)$  of the  $(k+1)$ th row of the matrix  $M_{k+1}$ , highlight this item in bold and red.

Step  $k+1.2$ . // The human user checks whether the location  $i_{k+1}^*$  contains the required target information; if yes, the search stops; otherwise increase by 1 the number of failed checks for location  $i_{k+1}^*$  //

Consider a current optimal subsequence  $\pi^* = \pi_{k+1}^*$  which for now is:  $\pi_{be\ k+1}^* = (\pi_1^*, \pi_2^*, \dots, \pi_{k+1}^*)$ ; compute the efficiency  $F(\pi_{k+1}^*)$ . Increase the number of unsuccessful scans of location  $\pi_{k+1}^*$  by 1, that is, change the second index for location  $\pi_{k+1}^*$  as follows:  $j_{k+2}(\pi_{k+1}^*) := j_{k+1}(\pi_{k+1}^*) + 1$ , leaving the second indices for all other locations unchanged:  $j_{i,k+2} := j_{i,k+1}$ ,  $i \neq i_{k+1}^*$ . Go to step  $k+1.3$ .

Step  $(k+1).3$ . //Determine all elements  $Q_{ij}$  in the  $(k+2)$ th row of matrix  $M_{k+2}$  //

Define the  $(k+2)$ th row of the matrix  $M_{(k+2)}$ , in which the elements  $Q_{ij}$  are calculated as follows: (i) place the parameter  $Q_{i_{(k+2)}, j_{(k+2)}}^*$  in cell  $(k+2, j_{k+2}(\pi_{k+1}^*))$ , and (ii) all other elements of the  $(k+2)$ th row of  $M_{(k+2)}$  must be the same as the corresponding elements in the previous row  $k+1$  in  $M_{(k+1)}$ , that is, set  $Q_{i, j_{k+2}(i)} := Q_{i, j_{k+1}(i)}$ , for  $i \neq i_{k+1}^*$ . Go to the next round. Rows  $1, \dots, k+1$  of the matrix  $M_{k+2}$  are left the same as the rows in the matrix  $M_{k+1}$ ; thus, the matrix  $M_{k+2}$  is built. The rounds are repeated until a human user is satisfied by the target information extracted by the chatbot.

This procedure is illustrated by a numerical example in Section 7. It illustrates the operation of the algorithm and shows that the paradoxical property manifests itself already at the early rounds of the iterative search process under consideration.

## 6. Comparison of Two Chatbots: The “Paradox”

The goal of this section is to uncover a counterintuitive property of the general AI chatbot information extraction problem when using optimal search policy, an AI based chatbot must find and extract target information in a multi-round iterative process. We suppose that since the search chatbot is unreliable, the human user is never satisfied with the chatbot's first answer and asks the latter to improve it with a refined query; in other words, a successful round is always preceded by at least one unsuccessful round, i.e., we assume that the parameter  $u(i)$  exceeds 1 for any  $i$ .

Suppose we have two search chatbots with the same time/efficiency characteristics and the same discount, but with different reliability. Let us denote chatbots 1 and 2, and assume that both are sufficiently reliable in the sense that their overlook probabilities are quite small:  $0 < a_i^s(s) < 0.5$ ,  $s = 1, 2$ , in all locations  $i = 1, \dots, N$ . For definiteness, let's assume that the first chatbot is more reliable, that is,  $0 < a_i^{(1)} < a_i^{(2)}$ ,  $i = 1, \dots, N$ . Both chatbots operate in the same environment, but due to different reliability, their optimal search policies may be different.

Let  $\pi^{*(1)}$  be the optimal policy for chatbot 1, and  $\pi^{*(2)}$  be the optimal policy for chatbot 2. Accordingly, let  $F_1(\pi^{*(1)})$  and  $F_2(\pi^{*(2)})$  denote the satisfaction functions obtained by the first and second chatbot, respectively.

The main theoretical result of this work is as follows:

**Proposition 2.** *If the chatbots are sufficiently reliable, that is,  $0 < a_i^{(1)} < a_i^{(2)}$ ,  $i = 1, \dots, N$ , then, when using the optimal search policy, a less reliable chatbot is more efficient, in the sense that its search efficiency is greater than the efficiency of a more reliable chatbot, i.e.,  $F_1(\pi^{*(1)}) < F_2(\pi^{*(2)})$ .*

To prove Proposition 2, we first need to prove the following auxiliary claim:

**Proposition 3.** *Let the function  $f(x)$  be  $x^{k-1}(1-x)$ , where  $k$  is a positive integer greater than 1. Then  $f(x)$  is increasing in  $x$  when  $0 < x < 0.5$ .*

**Proof.** We have:  $f(x) = x^{k-1}(1-x) = x^{k-1}-x^k$ ,  $k > 1$ .

Then  $f'(x) = (x^{k-1} - x^k)' = x^{k-2}(k-1-kx)$ . Since  $x > 0$ ,  $f'(x) > 0$  iff  $k-1-kx > 0$ , or, equivalently,  $f'(x) > 0$  iff  $x < (k-1)/k$ .

Note that  $(k-1)/k \geq 0.5$  for any integer  $k > 1$ . Then, for  $k > 1$ , we have:  $0 < x < 0.5 \leq (k-1)/k$ .

Therefore,  $f'(x) > 0$ , and, hence,  $f(x)$  is increasing.  $\square$

Now we return to the proof of Proposition 2. Let us denote the optimal expected discounted satisfactions for the chatbots 1 and 2 by  $F_1(\pi^{*(1)}) = E(R_{disc,1}(\pi^{*(1)}))$  and  $F_2(\pi^{*(2)}) = E(R_{disc,2}(\pi^{*(2)}))$ , respectively. Since the reliabilities of the two chatbots are different, the optimal search strategies of these smart robots, denoted  $\pi^{*(1)} = (\pi_1^{*(1)}, \pi_2^{*(1)}, \dots, \pi_m^{*(1)}, \dots)$  and  $\pi^{*(2)} = (\pi_1^{*(2)}, \pi_2^{*(2)}, \dots, \pi_m^{*(2)}, \dots)$ , respectively, in the general case can be different, despite that all other characteristics of them are pairwise identical. The optimal values of the corresponding expected discounted satisfaction,  $F_1(\pi^{*(1)})$  and  $F_2(\pi^{*(2)})$ , can also be different.

According to expression (1), we have:

$$F_1(\pi^{*(1)}) - F_2(\pi^{*(1)}) = \sum_{m=1}^{\infty} r_{\pi_m^{*(1)}} e^{-d \sum_{k=1}^m t_{\pi_k^{*(1)}}} \cdot (P_{\pi_m^{*(1)}} p_{\pi_m^{*(1)}} - P_{\pi_m^{*(2)}} p_{\pi_m^{*(1)}}) = \\ \sum_{m=1}^{\infty} r_{\pi_m^{*(1)}} e^{-d \sum_{k=1}^m t_{\pi_k^{*(1)}}} [a_{\pi_m^{*(1)}}^{u(\pi^{*(1)}, m)} (1 - a_{\pi_m^{*(1)}}) p_{\pi_m^{*(1)}} - a_{\pi_m^{*(2)}}^{u(\pi^{*(2)}, m)} (1 - a_{\pi_m^{*(2)}}) p_{\pi_m^{*(1)}}]. \quad (5)$$

If  $0 < a_{\pi_m^{*(1)}} < a_{\pi_m^{*(2)}} < 0.5$  for all  $m$ , then, according to Proposition 3, we have:

$$a_{\pi_m^{*(1)}}^{u(\pi^{*(1)}, m)} (1 - a_{\pi_m^{*(1)}}) p_{\pi_m^{*(1)}} - a_{\pi_m^{*(2)}}^{u(\pi^{*(2)}, m)} (1 - a_{\pi_m^{*(2)}}) p_{\pi_m^{*(1)}} \leq 0. \quad (6)$$

Then, from (5) and (6) it follows that, if  $0 < a_{\pi_m^{*(1)}} < a_{\pi_m^{*(2)}} < 0.5$  for all  $m$ , then

$$F_1(\pi^{*(1)}) \leq F_2(\pi^{*(1)}). \quad (7)$$

Next, it is easy to see that

$$F_2(\pi^{*(1)}) \leq F_2(\pi^{*(2)}), \quad (8)$$

since  $\pi^{*(2)}$  denotes the optimal policy for the chatbot 2, that is,  $\pi^{*(2)}$  guarantees the maximum expected satisfaction  $F_2(\pi)$ . From (7) and (8), it follows that a less reliable chatbot is more efficient in the sense that it provides greater expected satisfaction to the human user than a less reliable chatbot.  $\square$

**Discussion.** At this point, two related questions arise: whether the paradoxical situation under consideration occurs: (1) when the chatbot is absolutely reliable, i.e., its probability  $a_i = 0$ , and (2) when the chatbot is not-perfectly-reliable, that is, such that its overlook probability is *arbitrarily small*.

Let's answer the first question first. A simple counterexample clearly shows that the answer is: *there is no paradox*. For completeness, we provide the proof in the next section.

Let us now answer the second question. At first glance, and here the answer should not be paradoxical, since the objective function defined in expression (1) is continuous in the variable  $a_i$  and, therefore, one can expect to receive the same answer as in the case when the probability  $a_i = 0$ . However, this answer is incorrect – in fact, it is only true for the special case where the chatbot is reliable and retrieves the target information during the first round of conversation. In this case, the process is finite and terminates after a single consecutive scan of each locations. But if the chatbot is unreliable and thus does not detect the target in the first round, then, starting from the second round of the multi-round iterative search, the objective function (1) is discontinuous with respect to the discrete variable  $u_i$  (i.e., the number of unsuccessful scans of location  $i$ ), and the proof based of the continuity of the objective function is no longer valid. The correct answer to the second question is this: if the AI chatbot does not retrieve a target information in the first round of search, then during the iterative search procedure of Section 5.2, the less reliable chatbot is more efficient - for any positive overlook probability  $a_i$ , even as small as you wish.

We conducted extensive numerical experiments for various overlook probabilities, including very small ones, which confirmed that the paradoxical phenomenon in question exists in practice and is easily reproducible.

## 7. Numerical Examples

### 7.1. The Illustrative Example

Consider the numerical example with six locations  $N = 6$ , two chatbots and input data presented in Table 2. The notation used is explained in Section 3.

**Table 2.** Input data for example,  $d=0.1$ .

location $i$	1	2	3	4	5	6
data						
$p_i$	0.1	0.05	0.15	0.2	0.3	0.2
$t_i$ (Min.)	3	2	4	1	3	5
$r_i$	5	10	4	7	2	12

$a_i^{(1)}$	0.2	0.05	0.2	0.04	0.1	0.3
$a_i^{(2)}$	0.3	0.1	0.4	0.1	0.2	0.35

Calculations of the algorithm are presented in Tables 2 and 3 for chatbots 1 and 2, respectively, for 12 consecutive rounds. The table columns correspond to the six locations considered; the rows represent rounds of the search example considered. The elements in the cells are the coefficients  $Q_{ij}$ , calculated for 12 rounds, row by row.

**Table 3.** Calculations of the optimal policy for chatbot 1.

location round	1	2	3	4	5	6	$\pi^1$	$F_1(\pi_1)$
1	0.2287	0.1073	0.1952	<b>0.5112</b>	0.1543	0.3237	4	0.0486
2	0.2287	0.1073	0.1952	0.0204	0.1543	<b>0.3237</b>	6	0.1639
3	<b>0.2287</b>	0.1073	0.1952	0.0204	0.1543	0.0971	1	0.1964
4	0.0457	0.1073	<b>0.1952</b>	0.0204	0.1543	0.0971	3	0.2226
5	0.0457	0.1073	0.0390	0.0204	<b>0.1543</b>	0.0971	5	0.2237
6	0.0457	<b>0.1073</b>	0.0390	0.0204	0.0154	0.0971	2	0.2239
7	0.0457	0.0054	0.0390	0.0204	0.0154	<b>0.0971</b>	6	0.2258
8	<b>0.0457</b>	0.0003	0.0390	0.0204	0.0154	0.0291	1	0.2270
9	0.0091	0.0003	<b>0.0390</b>	0.0204	0.0154	0.0291	3	0.227004
10	0.0091	0.0003	0.0104	<b>0.0204</b>	0.0154	0.0291	4	0.227008
11	0.0091	0.0003	0.0104	0.0008	0.0154	<b>0.0291</b>	6	0.227312
12	0.0091	0.0003	0.0104	0.0008	<b>0.0154</b>	0.0087	5	0.227321

The number in cell  $(i, j)$  of these tables represents the parameter  $Q_{ij}$ , calculated by formula (4), where  $i=1, \dots, 12$ ;  $j=1, \dots, 6$ , for two chatbots. The tables represent the matrices  $M_k$  for two chatbots, introduced in Section 5.2. Let us see the calculations in Table 3 for chatbot 1; similar calculations are performed in Table 4 for chatbot 2.

For chatbot 1, in the first round, the first row of matrix  $M_1$  (i.e., the first row of Table 3) is filled in with the coefficients  $Q_{i,1}$ ,  $i=1, \dots, N$ . Then, all the locations are (unsuccessfully) scanned once, after which the  $Q^{*(1),1} = \max_i Q_{i,1} = Q_{4,1} = 0.5112$  and the corresponding location  $i^*(1) = 4$  are found. Next,

during this round, chatbot 1 scans location 4 once more; we assume that in this example it does not discover the relevant target information there. Then the number of unsuccessful scans of this location increases to 2, the corresponding parameter  $Q_{4,2}$  is calculated according to (4), (equals 0.0204) and placed in cell (2,4) of matrix  $M_2$ . All other elements of the second row of this matrix are left unchanged - the same as in the previous row of  $M_1$ . The first round of the algorithm is completed, and then the second round starts.

**Table 4.** Calculations of the optimal policy for chatbot 2.

location round	1	2	3	4	5	6	$\pi^2$	$F_2(\pi_2)$
1	0.3001	0.2032	0.2928	<b>1.1980</b>	0.2744	0.3507	4	0.1140
2	0.3001	0.2032	0.2928	0.1198	0.2744	<b>0.3507</b>	6	0.2389
3	<b>0.3001</b>	0.2032	0.2928	0.1198	0.2744	0.1227	1	0.2816
4	0.0900	0.2032	<b>0.2928</b>	0.1198	0.2744	0.1227	3	0.3208
5	0.0900	0.2032	0.1171	0.1198	<b>0.2744</b>	0.1227	5	0.3402
6	0.0900	<b>0.2032</b>	0.1171	0.1198	0.0549	0.1227	2	0.3476
7	0.0900	0.0203	0.1171	0.1198	0.0549	<b>0.1227</b>	6	0.3562
8	0.0900	0.0203	0.1171	<b>0.1198</b>	0.0549	0.0430	4	0.3574
9	0.0900	0.0203	<b>0.1171</b>	0.0120	0.0549	0.0430	3	0.3591
10	0.0900	0.0203	0.0187	0.0120	<b>0.0549</b>	0.0430	5	0.359311
11	<b>0.0900</b>	0.0203	0.0187	0.0120	0.0110	0.0430	1	0.359696
12	0.0027	0.0203	0.0187	0.0120	0.0110	<b>0.0430</b>	6	0.360275

In round 2, the next location to be scanned next is selected by computing  $Q_{i^*(2),2} = \max_i Q_{i,2} = Q_{6,2} = 0.3237$ , and, hence, the location  $i^*(2) = 6$  is chosen. Place  $Q_{i^*(2),2}$  in cell (2, 6) of the 2nd row of matrix  $M_2$ , and highlight this item in bold and red.

Next, during this round, chatbot 1 scans location 6 once more; we assume that the human user is not satisfied by the target information extracted there. Then the number of unsuccessful scans of this location increases to 2, the corresponding parameter  $Q_{6,3}$  is calculated according to (4), (and equals 0.0204) and placed in cell (3,6) of the next matrix  $M_3$ . All other elements of the third row of this matrix are left unchanged - the same as in the previous row of  $M_2$ . The second round is completed.

During the subsequent rounds, the algorithm filled in sequentially the next rows 3, 4, ..., 12 of Table 3, following the instructions of the algorithm as at round 2.

We assume that the human user is satisfied by the target information retrieved in the 12th round. The results obtained for each chatbot over 12 rounds are as follows: (1) The optimal policies for the chatbots 1 and 2 are equal to  $\pi^{*(1)} = (4,6,1,3,5,2,6,1,3,4,6,5)$  and  $\pi^{*(2)} = (4,6,1,3,5,2,6,4,3,5,1,6)$ , respectively. (2) The partial efficiency  $F_1(\pi^{*(1)})$  and, accordingly,  $F_2(\pi^{*(2)})$ , obtained at each round for two chatbots, is shown in the corresponding cells of the last column of Tables 3 and Table 4, respectively. Comparing these results, we can see that in each round of the search process, the optimal expected search efficiency for a less reliable chatbot is noticeably larger than the maximum efficiency of a more reliable chatbot.



### 7.2. An Absolutely Reliable Chatbot Does Not Have the Paradox in Question

Consider the following example, whose idea was suggested by I.M. Sonin. For the absolutely reliable chatbot numbered 1 set  $a_i^1 = 0$ ; for the chatbot 2 set  $a_i^2 = 0.1$ ; the known satisfaction levels  $r_i = 1.0$  for each chatbot and any location, and the prior probabilities that the target information is hidden at location  $i$  are  $p_i = 0.5$ ,  $i=1,2$ ;  $d = 0.1$ . Then the absolutely reliable chatbot 1 will inevitably find the target information in a right location either during the first search with a probability of  $1/2$ , or during the second search (after unsuccessful first search) with a probability of  $1/2$ , and the search process ends. Assuming  $d=0.1$ , the total objective function  $F^1(\pi^*)$  for the first chatbot will be:  $p_i \cdot 1.0 + p_i \cdot 1.0 \cdot e^{-0.1}$ , that is, about 0.96.

As for the second, unreliable chatbot, since  $a_i^2 > 0$ , the iterative search in the given two locations can have an infinite number of rounds, and the expected total satisfaction over an infinite horizon will obey expression (1). Elementary arithmetic operations on formula (1) using data for the chatbot 2 and the sum of the infinite geometrical progression show that the expected satisfaction  $F^2(\pi^*)$  of the second chatbot does not exceed 0.49 points, and in fact it will be even much less for very small  $a_i^2 > 0$ . We see that the efficiency of an absolutely reliable chatbot is greater, which means that *there is no paradox* for this case.

## 8. Concluding Remarks and Open Problems

The focus of this work is to study a realistic situation where the overlook probability of a chatbot is non-zero and thus the search process does not end during the first round of human-chatbot conversation. The main contribution of this paper is twofold. First, we derive and study a greedy index-based search policy and prove that it is optimal for the optimal search problem under study. Second, we identify a paradoxical “more-for-less” phenomenon: a less reliable chatbot has a higher efficiency for any positive overlook probability, even if it is very small.

Despite the “paradoxical” wording of the title, the result of this paper is quite logically explainable and completely compatible with common sense. The point is that if a more reliable chatbot has a lower overlook probability, then it is plausible to think that it will find the target information faster, i.e., in fewer search rounds and ultimately in less average search time.

This common sense observation is entirely consistent with the well-known fact from probability theory about the random variable  $x$ , which is called the geometric random variable with parameter  $p$ , and which represents the random number of independent unsuccessful trials (rounds) before the target is found; this probability is given by  $p(k) = p(1-p)^{k-1}$ ,  $k = 1, 2, \dots$ . In this case, it is known that the expected number of trials equals  $1/p = 1/(1-a_i)$  [28]. Thus, the higher the overlook probability, the greater the expected number of rounds and the average search time before the target information is found, and therefore the greater the total expected accumulated information and the search efficiency determined by expression (1). So, in essence, there is no contradiction in the obtained result.

It would be interesting to find what paradoxical situations arise for other search/extraction scenarios encountered in AI chatbot search, in particular for multi-target, multi-criteria, and other scenarios. A more general problem is to find and resolve real-world situations for good search policies (ideally, optimal or approximate with efficiency guarantees) based on advanced algorithmic ideas such as optimizing-with-learning [29] and search with reinforcement [30].

**Acknowledgements:** The authors are grateful to I.M. Sonin for helpful discussions.

**Authors contributions:** Both authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Boris Kriheli and Eugene Levner. Both authors read and approved the final manuscript.

**Statement about funding:** No funds, grants, or other support was received.

**Statement about none of competing interests:** The both authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Nicolescu, Luminița, and Monica Teodora Tudorache, 2022. Human-computer interaction in customer service: the experience with AI chatbots—a systematic literature review. *Electronics* 11, no. 10.
2. Izadi, Saadat, and Mohamad Forouzanfar, 2024. Error correction and adaptation in conversational AI: A review of techniques and applications in chatbots." *AI* 5, no. 2: 803-841.
3. Akdemir, Doğan Mert, and Zeki Atıl Bulut, 2024. Business and customer-based chatbot activities: The role of customer satisfaction in online purchase intention and intention to reuse chatbots. *Journal of Theoretical and Applied Electronic Commerce Research* 19, no. 4: 2961-2979.
4. Alabbas, Abeer, and Khalid Alomar, 2025. A weighted composite metric for evaluating user experience in educational chatbots: Balancing usability, engagement, and effectiveness." *Future Internet* 17, no. 2.
5. Ross, S. M., 1969. A problem in optimal search and stop. *Operations Research*, 17(6), pp.984-992.
6. Sweat, C. W., 1970. Sequential search with discounted income, the discount a function of the cell searched, *The Annals of Mathematical Statistics*, 1(5), pp.1446-1455.
7. Chew M.C. Jr., 1973 Optimal stopping in a discrete search problem. *Operations Research* 21(3):741–747. 4
8. Ross, S.M., 1983. Optimal search. Ch. 5 and 6 in Ross, S.M., 1983. *Introduction to Stochastic Dynamic Programming*. Academic Press, New York, pp.63-71.
9. Levner, E., 1994. Infinite-horizon scheduling algorithms for optimal search for hidden objects. *International Transactions in Operational Research*, 1(2), pp.241-250.
10. Cheng, T.C.E., Kriheli, B., Levner, E., Ng, C.T., 2021. Scheduling an autonomous robot searching for hidden targets. *Annals of Operations Research*, Springer, 298(1), pp. 95-109.
11. Koopman, B., 1956. The theory of search, II. Target detection, *Operations Research*, 4, 324-346.
12. Stone, L.D., 1969. *Theory of Optimal Search*, New York: Academic Press.
13. Stone, L.D., Royset, J.O., Washburn, A.R. 2016. Search for a stationary target, pp.9-48. In Stone, L.D., Royset, J.O. and Washburn, A.R., 2016. *Optimal Search for Moving Targets*, vol. 237. Springer, Cham. <https://doi.org/10.1007/978-3-319-26899-6-2>.
14. Trummel, K.E and Weisinger, J.R., 1986. The complexity of the optimal searcher path problem. *Operations Research*, vol. 34, no. 2, pp. 324-327.
15. Wegener, I., 1985. Optimal search with positive switch cost is NP-hard, *Information Processing. Letters*, 21,1, 49-52.
16. Kriheli, Boris, Eugene Levner, and Alexander Spivak, 2016. Optimal search for hidden targets by unmanned aerial vehicles under imperfect inspections." *American Journal of Operations Research* 6, no. 02 (2016): 153.
17. Kress, M., Lin, K.Y. and Szechtman, R., 2008. Optimal discrete search with imperfect specificity. *Mathematical Methods of Operations Research*, 68, 539-549. <http://dx.doi.org/10.1007/s00186-007-0197-2>
18. Steinberg, R. and Zangwill, W.I., 1983. The prevalence of Braess' paradox. *Transportation Science*, 17(3), pp.301-318.
19. Shan, Fei, Hui Li, Zhongren Wang, Ming Jin, and Dawei Chen, 2024. Optimizing rural highway maintenance scheme with mathematical programming. *Applied Sciences* 14, no. 18.
20. Schäfer, Benjamin, Thiemo Pesch, Debsankha Manik, Julian Gollenstede, Guosong Lin, Hans-Peter Beck, Dirk Witthaut, and Marc Timme, 2022. Understanding Braess' paradox in power grids. *Nature Communications* 13, no. 1: 5396.
21. Gershenson, Carlos, and Dirk Helbing, 2015. When slower is faster. *Complexity* 21, no. 2: 9-15.
22. Taitler, Boaz, and Omer Ben-Porat, 2024. Braess's Paradox of Generative AI. *arXiv preprint arXiv:2409.05506*.
23. Sorrell, Steve (2009). Exploring Jevons' Paradox. *Energy Efficiency and Sustainable Consumption: The Rebound Effect*. London: Palgrave Macmillan. pp. 136–164.
24. York, Richard; McGee, Julius Alexander (2 January 2016). "Understanding the Jevons paradox". *Environmental Sociology*. 2 (1): pp.77–87.
25. Spieksma, F.C.R. and Woeginger, G.J. 2005. The no-wait flow-shop paradox. *Operations Research Letters*, 33(6), pp603-608. <https://doi.org/10.1016/j.orl.2004.10.007>.

26. Charnes, A. and Klingman, D. 1971. The more-for-less paradox in the distribution model, *Cahiers du Centre d'Etudes de Recherche Opérationnelle*, 13, pp.11–22.
27. Graham, R.L., 1969. Bounds on multiprocessing timing anomalies, *SIAM J. Appl. Math.* 17 pp.416–429.
28. Ross, S.M., 1976. *A First Course in Probability*, New York: Macmillan.
29. Powell, W.B., 2016. Perspectives of approximate dynamic programming. *Annals of Operations Research*, Springer, vol. 298(1), pp. 95-109.
30. Zhang, T. and Mo, H., 2021. Reinforcement learning for robot research: A comprehensive review and open issues. *International Journal of Advanced Robotic Systems*, 18(3), 17298814211007305.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.