Introduction

The terms ‘search’ and ‘search problem’ arise in a number of problems that appear in different fields of applied mathematics and engineering. In its literal meaning, the notion of search problem corresponds to the class of situations in which an agent (a searcher) is looking for a hidden object (either one of them can be physical or abstract) by screening a certain defined area. The search problem is formulated under various restrictions on the considered search implementation as well as on the agent and the target functionalities.

To illustrate the potential complexity that might be considered in a search problem let us start with a simple search example and gradually add to it various assumptions and conditions. The figure below presents a simple schematic view of a search problem where the target (a vehicle) is located at some point in a given space, while the searcher or searchers (aircraft) are looking for it. An initial classification of the problem depends on the definition of the sample space, which can be either discrete or continuous. In this book we mainly consider the former case, which implies that the target and the searcher move to well-defined points in the space. These discrete positions can also be modeled by nodes in a graph. This type of presentation is popular in the artificial intelligence (AI) literature where cost parameters or weights are often added to the edges of the graph, such that the overall cost of the search is obtained by accumulating these costs along the search path over the edges. When the weights are distributed unevenly, the search procedure can account for different considerations, such as non-homogeneous search distances or search efforts. We consider some of these cases. A second critical feature of the problem is related to the ability of the target to move in the space. In the case of a moving target, several versions exist for representing its type and path. Some of the most popular schemes are random moves, Markovian moves, and Brownian moves. We address these assumptions in the relevant sections. In general, optimal solutions exist for static search problems but they often do not exist for dynamic search problems with a moving target. In this book we consider both approaches, and in particular we propose a general search scheme that applies to both cases. A third feature of the search is related to the information available to the searcher. If the location of the target is known, then a complete-information search
problem can be mapped to a relatively simpler path-planning or chase-planning problem. These types of problems often appear in the operations research literature.

The origin of these deterministic search problems for a moving target was the *pursuit problem* that was formulated in the eighteenth century. This class of problem is computationally tractable and often focuses on capturing the target with a minimal number of search moves. In this book we focus almost entirely on the *incomplete-information search*, where the exact location of the target is generally unknown to the searcher. Note that there are several methodological concepts for addressing the incomplete-information search, for example, by relying on rough-set theory, fuzzy logic, or on probability theory. We follow the latter *probabilistic search* approach, modeling the incomplete information on the target location by a function that quantifies the probability of the target to be located at any point in the space (we call it the *location probability*). The search then becomes a probabilistic search and in many cases an adaptive one, where the results of the search up to a certain time are used to update the location probability distribution over the space, often by using a Bayesian statistics approach as we do here. A popular extension of the pursuit problem with incomplete information is known as the *search and screening problem*, formulated during World War II by Bernard Koopman. The problem is probabilistic not only with respect to the location of the target, but also with respect to the distribution of the *search efforts* that are applied in a continuous manner by the searcher to the search space.

![Diagram of search and screening problem](image)

We follow this approach, although we do not use the notion of distributed efforts. Instead, we assume that the search can be applied to discrete points of the search space. An important extension of the distributed search efforts, under a consideration of a discrete search space, is the *group-testing* search. In group testing the searcher can look for the target in a subspace of the search space and obtain an indication of whether the target is located somewhere in this subspace. The size of the allowed subspace is treated as an input parameter, while the search terminates if the subspace contains only a single point, thus representing complete information on the target location. In this book, we explicitly consider methods of group testing. If the target is static the search can be modeled by a coding theory process (where a code represents the location of the target in the space), while the coding procedures can be easily mapped to obtain the optimal search policy. These cases are often represented by decision trees that have become extremely popular in data-mining applications. In dynamic search, when the target is moving, such isomorphism between coding theory and search is no longer valid, so we propose a methodology that can be extended also to these cases.
There are several variants of the incomplete-information search. We often assume that the target is unaware of the search agent. If this is not the case, then the search process becomes a *search game* relying on some game theory concepts. We will shortly address these search games. Another conventional and realistic assumption is that the searcher’s observations are prone to some *observation errors*. In these cases, two types of statistical errors have to be taken into account – either missing the target even though the searcher has searched the right point (a false negative error), or falsely indicating that the target has been found at a certain point (a false positive error). Of these two errors, the false negative one is much more popular, and we consider a few such cases in later chapters.

Another version of the incomplete-information search, which is also considered in this book, addresses the situation of *side information*, where the searcher obtains some (incomplete) indication during the search of where the target is located. Another natural extension to all of the above methods is obtained when assuming that there is more than one target or one searcher in the search space. In such a case the question regarding the amount of cooperation among the targets or among search agents arises. A simple example of such cooperation is information sharing between the searchers in order to better estimate the location probability distribution and better utilize the joint search efforts. These extensions are discussed in the book as well.

We must stress the fact that the general formulation of the search problem as presented in this book does not distinguish between search for existing physical objects, such as cell (mobile) phones, people, and devices, and search for abstract entities, such as records in a database, an e-commerce customer on the Internet, a targeted customer type, or a search for feasible solutions of a given problem within a predefined solution space. Some popular tools for such search procedures can be found in the data-mining and statistics literature. We draw clear lines of similarities between search procedures for physical entities and those found in *problem-solving* procedures, typically in *stochastic local search* methods that are used to obtain feasible solutions to a given schematic problem.

In any case, even from the above simple example it can be understood that the search problem in its general form can be very complex, highly variant, and call for different solution schemes, which are partially covered in this book.

In summary, it is important to note that despite the similar properties of the above-mentioned variants of the search problem, there is no formal and unified search theory that captures all these points. Instead, one can find different search procedures and considerations in various research areas, such as operations research, coding theory, information theory, graph theory, computer science, data mining, machine learning, statistics, and AI. We do not claim to present such a unified search theory in this book, but we do try to bridge some of these gaps by formalizing the main properties, procedures, and assumptions that are related to many of these search problems and their variants.

In this chapter we start by discussing the motivation and applications of the search problem. This is done through a brief historical note on the origin of the search problem and its development. Then, we provide a general description of the search problem, focusing on three research fields in which it appears, namely, *search and screening*, *group testing*, and *stochastic local search*. We specifically discuss the core features of the stochastic local search, which provides the main insights on the proposed information-based search approach, and attempt to bridge some of the gap between different search theories. We end the chapter by discussing the structure of the book.
1.1 Motivation and applications

The problem of search is one of the oldest mathematical problems which were originated by the first applications of calculus in the seventeenth century. The problem arose by considering an agent which is required to determine the location of a hidden object and to intercept it. In the second half of the seventeenth century, the problem was explicitly proposed by Claude Perrault to Leibniz and studied in cooperation with Huygens. Perrault himself associated this problem with earlier studies conducted by Fermat and Newton [1].

The first essentially mathematical formulation and analysis of the search problem, according to Nahin [1], relates to the paper of Pierre Bouguer, which was submitted to the French Academy on 16 January 1732. This paper considered the pursuit of a merchant ship by a pirate ship. This is the reason why nowadays such a problem is often called the \textit{pure pursuit problem}. In it the pursuing agent has complete information about the current location of the pursued object, and the task is to find the agent’s path which leads to a guaranteed interception of the object in minimal time. In Bouguer’s original pursuit problem, it was assumed that the merchant’s movement was directed in a straight line. In 1748 a required generalization of the problem in which the pursued object moved in circles was formulated and published in the English \textit{Ladies’ Diary Journal} by John Ash in the form of a puzzle.

Formal mathematical considerations of the pursuit problem for various paths of the pursued object and different characteristics of the pursuer were proposed in a review published by John Runkle (1859) in \textit{Mathematical Monthly}. Following this work, the pursuit problem was considered as an application of the theory of differential equations, and after publication of the classical monographs by Isaacs [2] and Pontryagin \textit{et al}. [3], it was often considered as a variant of the optimal control problem.

Nowadays, the terms ‘search’ and ‘search problem’ arise in a number of problems that appear in different areas of applied mathematics and engineering. Some of the problems deal with essentially discrete models that can be described by walks over graphs, while other problems address continuous models that require the application of differential equations and certain variation principles. The problem of search in its literal/physical meaning appears in military applications (see, e.g., [4–6]) as the search for an object hidden by the enemy, in geological tasks as a search for minerals, as well as in other fields such as in quality control as the search for faulty units in a production batch [7, 8]. In such tasks, the searcher’s detection abilities are restricted by sensor quality, and the order of search actions is often restricted by the physical abilities of the searcher.

Other variants of the search problem are obtained in search for intangible (abstract) items, such as in medical procedures when searching for the right diagnosis using decision trees, in cellular network paging in which the network controller looks for a mobile device address to redirect incoming calls [9], or in search for specific records in a file or database. In such tasks, the search can be modeled by relying on fault detection and group-testing principles [10, 11], by which the searcher is able to test parts of the system in a different order. Note that similar search properties appear in the tasks of data mining and classification, where the target of the search is defined as a certain property or data record.

Contrary to these indicated fields, in the tasks of optimization and operations research the search is considered as a method of problem solving, where the target is defined as a required solution that depends on a certain problem. In particular, the method of local
search [12, 13] for the solutions of NP-hard problems is based on sequential consideration of the solutions that are obtained at each step of the search while choosing the next solution from a close neighborhood of the current solution. Moreover, during recent decades search has obtained a similar meaning in the studies of AI [14, 15], where, for example, the A* family of algorithms is being intensively considered and used to solve related problems.

For each of these approaches to the search problem, certain algorithms and results were obtained over the years. Nevertheless, in spite of the clear association and analogies in such problems and methods, the connections between heuristic search methods (including the AI version), search and screening methods, and group-testing methods have not been fully established to date. The present work attempts to close the gap between these approaches to the search problem.

### 1.2 General description of the search problem

Search problems appear in different applications of invariant methods, formulations, and considerations. In the most general formulation, the problem includes a *searcher* that moves over a search space and looks for a *target* that is located somewhere in the search space. In this context, the searcher and the target can be considered either as physical agents so that the goal of the searcher is to find the target in minimal time (or minimal search effort), or as an abstract object, such as searching for the right medical diagnosis, where the searcher is considered as a problem-solver while the target is considered as a required solution of the problem.

The initial formulation of the problem dealt with the search for a hidden object by a physical agent acting in a *continuous space* and continuous or discrete time. In such a formulation, the problem was studied within the framework of stochastic optimization and control theory [16–18], and for most cases of search for a static target, exact solutions of the problem in terms of shorter search path or minimal search time were obtained. When considering search for a moving target, the problem was often addressed by using different approaches, including optimization methods [5, 6, 19] and game theory [20–22].

In contrast to search by physical agents, the abstract formulation of the search problem often considers the actions of the problem-solver over a *discrete space* of candidate solutions, where at each search step a single solution is obtained. In such a formulation, the problem appears both in the models of search over a gridded plane and in the tasks of AI [15, 23] and optimization theory [12], including group-testing methods for the processes of quality control and inspection [7, 8]. Moreover, in many studies [10], a group-testing formulation of the search problem has been considered, while in other applications the problem has been referred to as a problem of screening and detection [6].

A general formulation of the search problem in the group-testing form allows the implementation of a wide set of assumptions regarding the searcher and the target that relate to different tasks. The main assumption corresponds to the *searcher's knowledge* regarding the target, which determines the nature of the search space. In most cases, it is assumed that the search space consists of abstract states that represent either target locations or candidate solutions, and that the probability distribution over the search space is defined. If the probabilities of the states are not available to the searcher, then it can be assumed that this distribution is uniform or that the states are equally probable. This assumption stems naturally from the well-known maximum entropy principle [24].
Other assumptions that correspond to the searcher’s knowledge deal with detection and decision making. The first assumption is addressed by a detection function that defines the probability of errors of the first and second types. The Type I error corresponds to an erroneous declaration that the target has been found at a certain searched point. The Type II error corresponds to an erroneous declaration that the target is not located at a certain searched point. In most applications, it is assumed that the search is conservative – thus the search is prone to errors when checking a state where the target is located, while the detection of the absence of the target is errorless. Detection results, which can be represented either by Boolean ‘true’ or ‘false’ values or by detection probabilities, are used by the searcher to make decisions about the next state of the target and the next search action. In search by a physical agent, the detection function often corresponds to the properties of the available sensors, for example, when considering a car equipped with radar and cameras to avoid collisions with objects.

In the abstract version of the search problem, the detection function is often defined on the basis of the search space and its metric or topological properties. For example, when considering a problem-solving approach, a set of constraints can define the feasible search space.

Decision making and selection of the next search action usually depend on the detection result and on the information that is available to the searcher. In most optimization and group-testing algorithms, decision making is based on the probabilities of the states and on the search cost of the candidate decisions. The probabilities and costs are updated by the searcher during detection and support the decision making in further stages of the search [25]. In some AI methods [15], in contrast, the search is not defined by probabilistic terms – thus location probabilities are not modeled and decision making is mainly based on the distances (costs) between the states. In the popular A* algorithm of problem solving [14], the searcher uses side information about the required solution and applies an estimated distance (cost) to it, while choosing the next solution from the neighboring candidates.

Depending on the nature of the search problem, two main types of solution neighborhoods are defined. In the first type, the neighborhood includes all feasible solutions that are known at the current search step, and the choice of candidate solution is conducted over this set of feasible solutions. In such a case, some solution of the search problem can be obtained and there exist a number of algorithms, for example, Huffman search [26] or dynamic programming search [27, 28], that obtain the required solution. Nevertheless, these algorithms are mainly applicable to the search for a static target that corresponds to the search for a solution under the assumption that the uncertainty regarding the solution depends on the searcher’s actions and not on external factors. In the second solution type, the neighborhood includes a restricted subset of a search space so that the selection procedure requires a small number of calculations and comparisons, and a complete solution is obtained by sequential decision making and acceptance of a number of intermediate solutions that converge to the required solution. Such a search method is referred to as a local search and can be implemented in the search both for static and for moving targets. In terms of problem solving, the search for a moving target means that the uncertainty regarding a required solution changes over time and that the solution depends on some external factors beyond the control of the searcher. With some additional assumptions, such search procedures can be associated with a process of sequential decision making in game theory [29, 30] and, in particular, in search games [20–22], where the physical target agent can
move over the space according to its internal rule, while the goal of the search agent is to find the target in minimal time or with minimal effort.

Finally, the formulation and methods of consideration of the search problem depend on the number of search agents and on the number of targets. The abstract search problems usually deal with a single decision-maker looking for a single solution or a group of solutions that represent this solution. In contrast, the search problems that consider the actions of physical agents depend on the number of search agents and the number of target agents and their movement. In such problems, additional questions arise regarding collective decision making, shared (or common) information, as well as communication between the searchers. Similar questions are considered regarding the target agents and their responses to the searchers’ actions.

Based on the type of search problem and implemented assumptions, the problem can be addressed and solved by using different methods. Some of these methods are applicable to the problem of search in which physical agents act in a physical space; other methods are used in an abstract search for a solution of optimization or AI problems. In the next section, we consider some of the main approaches to problem-solving search that provide a context for the local search methods considered in the book.

### 1.3 Solution approaches in the literature

Methods for solving the search problem depend on its formulation and on the implemented assumptions and applied restrictions. For example, in the original pursuit problem or in its game theory formulation, called the pursuit-evasion problem, the pursuer has complete information on the position of the pursued object at any time. Accordingly, the goal is to determine the path of the pursuer given the velocity of the pursuer, the velocity and trajectory of the pursued object, and the governing rules of its movement. In such a formulation, the solution can be obtained by using suitable variation principles and, in most cases, it has a well-defined structure.

A variant of the pursuit problem with incomplete information – the so-called ‘dog in a fog’ problem – was initiated in 1942 by the US Navy’s Antisubmarine Warfare Operations Research Group. In a focused report [4] on the problem and later in a book [19], Koopman named this problem the search and screening problem. As indicated in a later report by Frost and Stone [31], the considered theory

is the study of how to most effectively employ limited resources when trying to find an object whose location is not precisely known. The goal is to deploy search assets to maximize the probability of locating the search object with the resources available. Sometimes this goal is stated in terms of minimizing the time to find the search object.

The objective in the search and screening problem, as stated in [31], is ‘to find methods, procedures, and algorithms that describe how to achieve these goals,’ where the pursuer is called the searcher and the pursued object is the target. It is assumed that the searcher acts under uncertainty, and that the available information on the target location changes during the search. The dependence between the amount of available information and the locations of the searcher and the target is defined by a detection function. This function defines the probability of detecting the target corresponding to the locations of the searcher and the
target. In the original pursuit problem, the function is reduced to a value that unambiguously
determines the location of the target.

The effects of the searcher’s path on the amount of information collected on the target
gives rise to the following optimization problem. On the one hand, the goal of the searcher
is to intercept the target with minimal resources or in minimal time. On the other hand, the
searcher’s knowledge about the target location often increases as the search continues. The
optimal path thus balances these two terms, collecting information while finding the target
in a minimum number of search steps.

An optimal solution of the search and screening problem for a Markovian target and
an exponential detection function was obtained in 1980 by Brown [32]. His work initiated
discussions and formulations of the search problem from various viewpoints and stimu-
lated further studies. Despite the fact that a complete formal theory of search and screening
has not been created, ongoing research work has covered many reasonable applications and
aspects of this problem. For reviews of the search and screening theory and various military
applications, see [5, 6, 33–36]. Recent results of the theory are also presented in [31, 37].
Further developments of the problem as formulated by Brown [32] are presented in [38].

In search and screening theory, it is assumed that the searcher has such available
resources that the reaction of the target to the searcher’s actions may be omitted. In special
cases, such a reaction is represented by the detection probability, which is defined by a
detection function [33]. If the behavior of the target strongly depends on the searcher’s
actions, then the search process is named the pursuit-evasion game, which is also called a
search game.

A search game known as ‘the Princess and the Monster’ is a pursuit-evasion game
that was formulated in 1965 by Isaacs [2]. In this game, the goal of the searcher is to
find the target, whereas the target is informed about the searcher’s behavior and tries to
escape interception. In the simplest case, the game is formulated as a hidden-object problem,
which applies to a search for a static target. In the case of a moving target, a game theory
aspect appears in the form of a rationality assumption that the target at each step of the
game follows the best actions to avoid interception. An overview of the methods and
results obtained in this game theory search are presented in the monographs in [20–22]
and in [30].

Another formulation of the search problem has its origins in group testing for cases
where the searcher can observe simultaneously more than a single point in the space at
each search action. Such a solution assumes that the costs of the searcher’s actions do not
depend on the size of the chosen subsets and do not change over time.

Originally group-testing procedures focused on diagnostic tasks and quality inspections.
Group testing was proposed during World War II by Dorfman [39] as a testing strategy to
identify defective units. The strategy involves testing a set of units simultaneously, taking
into account that probably most of these units are acceptable. Therefore, there is a high
probability of approving many units by a single group test. If, however, the test indicates that
there is a faulty unit in the set, then the set is partitioned into subsets, to each of which finer
group testing is applied. This iterative procedure of partitioning and testing terminates when
the faulty unit is found. The main motivation of the strategy is to minimize testing efforts
in terms of minimum average number of tests in a kind of zooming-in search procedure,
as indicated by Dorfman [39]:
Often in testing the results of manufacture, the work can be reduced greatly by examining only a sample of the population and rejecting the whole if the proportion of defectives in the sample is unduly large. In many inspections, however, the objective is to eliminate all the defective members of the population.

The goal of these studies, according to Dorfman [39], is ‘under certain conditions, [to] yield significant savings in effort and expense when a complete elimination of defective units is desired.’

The development of group-testing theory is closely related to other statistical theories, such as Wald’s theory of statistical decision functions [40], the theory of statistical games [29], and Raiffa’s Bayesian decision analysis [41]. An overview of the main methods and results of the theory are presented in the monographs in [42, 43].

Group-testing theory deals mainly with the problem of statistical decisions. According to Wald [40], this problem can be formulated as selecting the best action under uncertainty with regard to the state of nature given payoffs, or rewards for possible actions. In the simplest case, the decision-maker chooses from continuing or stopping the testing process. Another decision concerns the size of the test samples, which may depend on the test results.

In relation to the considered search problem, the group-testing formulation has the following decision process [10]. It is assumed that there is a set of possible target locations in the search space. At each step, the action available to the searcher is to choose a subset from the set of locations and check whether the target is located somewhere within this subset or not. The procedure terminates if the searcher finds the target in a single-point subset. In other words, the main considerations concentrate on determining the size and location of the samples, given a constant or variable detection function.

The implementation of group testing to a problem of search for a static target was suggested in 1959 by Zimmerman [26]. In contrast to conventional statistical methods at that time, Zimmerman’s procedure is directly related to methods of information theory, and particularly the Huffman coding procedure [44], to solve the search problem.

The analogy between testing (e.g., finding a defective unit) and coding relies on the fact that each unit in the set can be decoded by the sequenced testing outcomes of subsets to which it belongs [7]. The length of the code is thus analogous to the length of the testing procedure required to identify the defective unit. Zimmerman’s offline method is constructed by using the probability of each unit as the defective one. An optimal testing tree is constructed offline by the Huffman coding procedure, and then the test procedure follows this testing tree, assuming that no new information is revealed. Since the Huffman coding algorithm is optimal in the sense of minimal average code length, the analogous search procedure is optimal in the sense of minimal average search length [26].

Other group-testing methods that are based on coding procedures have been suggested over the years. In 1994, the Zimmerman procedure with the same assumptions was generalized by Abrahams [45] for a Huffman search by multiple searchers, and for additional requirements regarding the connectivity of the set of locations to a search based on the Hu–Tucker coding procedure [46]. Ben-Gal [7] analyzed the popular weight balance tree (WBT) group-testing algorithm, which is based on the simple principle of successive partitions of the search set into equiprobable subsets (known also as Shannon–Fano coding). At each step of the algorithm, the subset which contains the searched item (i.e., a faulty unit) is partitioned into equiprobable sub-subsets and the process repeats itself until the searched
item is found. Herer and Raz [8] considered a similar group-testing search for a defective product in a lot produced by a process with a constant failure rate.

Group-testing methods of search based on coding procedures essentially rely on the possibility of checking a whole subset in one search act. Nevertheless, for certain cases of location probabilities, these methods solve the problem in the form of sequences of single-point subsets. For such cases, the methods yield optimal solutions of the hidden-object problem indicated above.

The absence of a general optimal solution for the hidden-object problem contributed to the application of dynamic programming methods, as initiated in 1969 by Ross [47]. In these cases, it is assumed that, similar to the theory of statistical decision functions and hypothesis testing, there is a probability that the target will not be detected at its real location. Thus, equivalent to the Type I error or false negative testing, the hypothesis on the location of the target is correct, although the test statistic leads to the wrong decision. In further studies [48, 49], this problem was formulated as a Markov decision process (MDP), yet the assumption of an imperfect detection, as it is often called, remains and is used for estimating the convergence of the search process.

The first solution for the simplest variant of the considered search problem with a moving target was published in 1970 by Pollock [50]. He considered a search for a target moving over two boxes (or two points in the search space) according to known probabilities and presented an optimal threshold policy for the searcher moves in order to find the target in minimal time. Approximate computation of the values of the thresholds was given in [51]. In later studies [52–57], a number of attempts were made to generalize the Pollock model to an arbitrary number of boxes, but a general solution has yet to be found. The search and screening problem and the group-testing search were also formulized and addressed by operations research techniques. However, for both problems, general optimal solutions were not obtained. Hence, for these cases, Ross’s statement that the searching for a moving target is an open problem [28] remains valid and actual.

Another approach to the search problem was initiated by Dechter and Pearl in 1985 [58] in the field of AI. In these studies, a solution for learning the optimal path toward a hidden object with equal location probabilities was studied. The method, called the A* algorithm, is considered a good and general strategy for the optimal path-learning problem. In the introduction to their work, Dechter and Pearl write [58] (authors’ italics):

One of the most popular methods of exploiting heuristic information to cut down search time is the informed best-first strategy. The general philosophy of this strategy is to use the heuristic information to access the ‘merit’ latent in every candidate search avenue exposed during the search and then continue the exploration along the direction of highest merit.

The success of such an approach and its relevance and association to optimization methods gave rise to considerations of different heuristics in the search problem. In 1990, Korf [59] suggested a modification of the A* algorithm by introducing heuristic learning abilities; the obtained algorithm was named the Learning Real-Time A* (LRTA*) algorithm [59]. In this algorithm, the searcher moves over the vertices of a graph until the static target in one of the vertices is found.

In 1991, a generalization of Korf’s algorithm for an application of search for a moving target was suggested. The algorithm, called the Moving Target Search (MTS) algorithm, was presented by Ishida and Korf [60] (see also [61]). Similar to Korf’s algorithm, the
MTS algorithm does not use any probabilistic information about the target location and is based on a pessimistic assumption (a worse-case scenario) regarding the target moves.

Further studies in the field addressed the implementation of different heuristics and assumptions in the MTS framework. In 1995, Koenig and Simmons [62] presented the first variant of the algorithm acting on a graph with changing probabilities of the target location and using a constant search strategy [63]. In 2006, Koenig and Likhachev [64] presented a revised version of the Koenig and Simmons algorithm, which allows for changing the strategy during the search. An overview of algorithms with non-constant search strategies, as well as their results, is presented in the paper [65] by Shimbo and Ishida. Variants of A* search algorithms were introduced in a unified framework [66–67] and considered to be associated with the theory of heuristic search on the graphs.

1.4 Methods of local search

One of the most studied heuristics for searching for an optimal solution is called the local search approach. Such a search, which is in essence over a virtual domain, acts over a graph or a discrete space of possible solutions called the search space. The above-mentioned A* and LRTA* algorithms [58, 59] follow this approach for the tasks of AI [15], while the stochastic local search algorithms [12, 13] implement the local search approach for solving various optimization problems.

The main idea of the local search and Stochastic Local Search (SLS) algorithms has been described in Hoos and Stützle [12]. The searcher starts with some solution of the problem that is represented by an initial location in the search space and sequentially moves from the current location to a neighboring location. The process continues until a location that represents a suitable solution of the problem is reached. The neighborhood of each location includes a set of candidate solutions, thus the decision at each step is obtained on the basis of local information only, which reduces the search space immensely, leading to computational tractability. The local search algorithms, in which the decision making is based on randomized choices of candidate solutions from the neighboring set, are called Stochastic Local Search algorithms.

Formally, the local search algorithms are defined for an instance of the optimization problem and use the following objects [12]:

- a set of candidate solutions of the problem called search positions, locations, or states;
- a set of feasible solutions that is a subset of the set of candidate solutions;
- a neighborhood relation that determines the neighboring solutions for each candidate solution;
- a step function that, given a current solution, defines the choice of the next solution taken from the neighboring solutions; and
- a termination condition that interrupts or ends the search process.

For the SLS algorithms, in addition, an initialization function that specifies an initial probability distribution over a set of candidate solutions is often defined. In this case, the
step function defines a probability distribution over the neighbors of the current solution so that the decision making is conducted by using this distribution solely over the neighbors.

As indicated above, the same local search approach is popular in AI algorithms of solving problems by search [15]. Similar to classical optimization problems, in such AI algorithms the set of states (or solutions) with an initial state and a termination condition, which is often called the goal test, are defined. In AI algorithms, the neighborhood relation is determined by a set of actions that can be conducted by the search agent for any state. In addition to the defined components, the AI algorithms implement a path cost that measures the performance of the search process and a step cost that is used for decision making while choosing an action.

Based on these principles, a local search algorithm implements the following general procedure [12, 15]:

Given a set of feasible solutions:

- Set the current solution as an initial solution.
- While the termination condition for a current solution does not hold, do the following:
  - Determine neighboring solutions or possible actions for a current solution.
  - Choose a next solution from the set of neighboring solutions or choose and apply an action from the set of possible actions.
  - Set the current solution as an obtained solution.
- End while.
- Return current solution.

An implementation of the local search algorithms that follows this outlined procedure depends on the definition of a neighboring relation and on the principles of decision making regarding the next solutions or applied actions. In the optimization tasks [12], the neighboring solutions can be generated from a current solution by changing it according to a certain perturbation/updating rule, often related to the function gradient if it exists. In AI tasks [15], the neighboring solutions are usually obtained by constructive extensions of the current solution. Such an approach, as well as a search over a discrete set of solutions (states), can define a graph search that is represented by a walk over the graph of solutions from an initial solution to the final solution. In the graph search, the goal of the searcher is to find a path over the graph vertices such that the path cost is minimal. The path cost is defined as the sum or combination of step costs that depend on the distances between the chosen solutions. The distances are calculated by using suitable metrics or pseudo-metrics that represent the differences between the solutions in the context of the considered problem.

A number of approaches to decision making exist, such that for a given current solution, they provide a choice (set of possible selections) of the next solution from the neighboring solutions. Selection of an action results in the next solution with a new neighborhood and, accordingly, a new set of possible selections. The method of decision making essentially depends on the nature of the problem and available data. However, a general decision-making procedure with solutions and action choices, both in the SLS and in the AI local search, follows the framework of a MDP. As indicated, this framework provides a general description of sequential decision making [49, 68].
Within this framework, for each state that represents a candidate solution of the problem, a set of actions is defined, and an action is chosen independently of the decisions that were obtained for the previous solutions. In other words, the current solution includes by definition all the required information on previously obtained solutions.

The decision making in MDP is conducted on the basis of a certain probabilistic rule and probability updating. Following the SLS approach [12, 13], these probabilities are defined by an *initialization function* that specifies initial probabilities for the candidate solutions that aim to solve the problem, and then the probabilities are updated according to some reasonable probabilistic or informational scheme. In the group-testing tasks [10], such a rule is provided by the Bayesian approach, while in the other cases, non-Bayesian methods are applied. In the AI local search [15], these probabilistic methods are not implemented directly, since a probability function is not defined explicitly over the search space. However, the considered tasks can be formulated by using probabilistic or, more generally, measure theory forms, and can be studied following the unified framework that includes the methods of both SLS and AI local search. Within this approach, we present in this book the *Informational Learning Real-Time A*\(^*\) (ILRTA\(^*\)) and the *Informational Moving Target Search* (IMTS) algorithms [69, 70].

Similar to LRTA\(^*\) and MTS, the proposed ILRTA\(^*\) and IMTS algorithms operate on a graph. However, unlike the former methods, in the ILRTA\(^*\) and IMTS the graph vertices represent partitions. These partitions stand for the possible results of the searcher’s group tests. For example, if the searcher tests a subset of points where the target might be located, the relevant partition of points following the test is represented by the corresponding vertex in the graph. The searcher’s decision to move to a certain vertex in the graph represents the selection and execution of a specific group test. The target location can be represented by the *target partition* that corresponds to the target location point and the complementary subset of all the other points. The search is terminated when the searcher selects a partition which is either identical or refined with respect to the target partition; we thus call it the *final partition* since the search ends at that stage. Reaching such a partition implies that the searcher obtains the necessary information regarding the target location, as a result of the group test. Similar to the LRTA\(^*\) [59] and MTS [60] algorithms, the edges between the vertices in ILRTA\(^*\) and IMTS are weighted by the corresponding distance measures. However, since these distance measures are between pairs of partitions, the ILRTA\(^*\) and IMTS algorithms apply specific information theory distance measures, and, in particular, they use the Rokhlin distance metric which is based on the relative entropies of these partitions [71].

As will be further explained in later chapters, similar to other SLS procedures (e.g., [62, 64]), in the ILRTA\(^*\) and IMTS algorithms the partitions with known distances to the current searcher’s partition define a *neighborhood*. The distances to all the other points in the graph have to be estimated and refined when new information is obtained from the next group tests. The iterative ILRTA\(^*\) and IMTS algorithms seek to find the shortest path between the searcher partition and the target partition, and thus select a path that correspond to the group test results. In summary, as in other search algorithms, the proposed ILRTA\(^*\) and IMTS implement a heuristic search based on distance estimates between the target partition and feasible search partitions. The uniqueness of these methods is that they use informational metrics and non-binary partitions to represent a group-testing search process over a graph of partitions (see also [72, 73]).
1.5 Objectives and structure of the book

This book describes various problems and solution methods that are related to the probabilistic search problem. It aims at providing a somewhat unified understanding of how to use stochastic local search principles combined with information theory measures to address the search problem in general. In the book we describe the main methods that have been used in the context of search problems over the years. We address the problem of search for static and moving targets in the context of group-testing theory and apply the methods and measures of information theory to different A*-type heuristics. It is assumed that the target is located or moves within a discrete set of possible locations. The action available to the searcher is checking a subset of locations (points) to determine whether the target is somewhere in this subset or not. In some cases the subset includes only a single point to represent a regular search. The procedure terminates if the searcher finds the target in a subset that contains only one point. The movements of the target and the searcher are considered in discrete time, and the searcher’s goal is to choose a sequence of subsets such that the search procedure terminates in a minimal expected number of steps.

The research aims to suggest a unified framework for a search for both static and moving targets, and to present novel algorithms of search for static and moving targets. The approach is based on the following key ideas that distinguish it from previous known methods.

A key feature of the proposed search procedure is that it acts on the search space that is a set of partitions of the sample space, and not on the set of locations. In contrast to known models of search in the form of MDPs, the decision-maker in the proposed search procedure takes into account both the chosen subset of the sample space and a set of subsets, which complete the chosen subset for the full sample space.

Using the set of partitions of the sample space enables different distance measures to be applied between candidate solutions in the proposed search scheme. In particular, we apply the Rokhlin metric [71] as a distance measure between partitions, and in certain case we use the Ornstein metric [74] as an estimation of the Rokhlin distance. The implementation of such metrics allows us to associate the considered search procedures within a broader context of general ergodic theory and dynamical systems theory [71].

Since the Rokhlin metric is essentially based on the relative entropies of the considered partitions, its application clarifies the connection between search procedures in information theory and coding procedures. This, in turn, allows us to apply information theory methods in considerations of the search procedures.

The rest of the book is organized as follows:

Chapter 2. In this chapter we describe the search problem as it appears in different applications. We start with the search and screening problem and describe the solution methods that use global optimization techniques and local search methods. For the group-testing theory that represents a search with unrestricted searcher abilities, we describe basic information theory methods and combinatorial methods. The search over graphs provides the most general search heuristics as they are implemented in optimization tasks and in AI. Finally, we provide notes on search games over continuous domains and over graphs.

Chapter 3. In this chapter we describe mathematical methods and models that are applied to the search problem. We start with the framework of MDPs and continue with the
methods of search and screening and group-testing search. Finally, we discuss some game theory models of search.

Chapter 4. This chapter focuses entirely on methods of information theory search. We then develop a search procedure that is based on the partitions space, as obtained by the searcher and the target moves. For this purpose we use the Rokhlin distance between the partitions as distance measures. We formulate an algorithm of search for static targets, which is based on the LRTA* algorithm, and prove its main properties. We then consider the informational algorithm of search for a moving target, which is based on the MTS algorithm. Finally, we present certain generalizations on the basis of the ILRTA* and MTS algorithms, such as search with multiple searchers for both cooperative and non-cooperative behavior.

Chapter 5. This chapter considers some related applications and perspectives of the search problem. We address a known data-mining problem of data classification and present an algorithm for constructing classification trees by using a recursive version of the ILRTA* algorithm. Another application presents a real-time algorithm of search for static and moving targets by single and multiple searchers.

Chapter 6. This chapter concludes the book with some general observations and future research directions.

References


