Hill-climbing Search

Intermediate article

Bart Selman, Cornell University, Ithaca, New York, USA
Carla P Gomes, Cornell University, Ithaca, New York, USA

CONTENTS

Many computational and cognitive tasks involve a search through a large space of possible solutions. Hill-climbing, or local search, is one strategy for searching such a solution space.

LOCAL VERSUS GLOBAL SEARCH

There are many problems that require a search of a large number of possible solutions to find one that satisfies all the constraints. As a basic example consider the problem of composing a classroom schedule given a set of constraints on the availability of classrooms, and various time constraints involving students and lecturers. Even with a relatively simple set of constraints, one may be forced to search through many possible schedules in order to find one that satisfies all constraints. In certain cases, computer scientists have found clever algorithms to solve such computational problems without having to search through the space of all potential solutions. However, in many other cases, such a search cannot be avoided. We refer to the search space as a *combinatorial* search space. A key question is what is the best way of searching a combinatorial space. The answer often depends on the underlying problem to be solved. In order to illustrate the two main techniques for searching a combinatorial space, we consider the N-queens problem as an example. (*See* **Search**; **Constraint Satisfaction**; **Problem Solving**)

The N-queens problem is that of placing N queens on an N by N chess board, so that no two queens can attack each other. Carl Friedrich Gauss considered this problem in its original form on an 8 by 8 chess board (Campbell, 1977). Given that a queen can move horizontally in its row, it follows that we can have at most one queen in each row. In fact, because we need to place *N* queens, a solution will require us to place exactly one queen in each row. Similarly, because of the movement of a queen in its column, the placement of queens is such that there is exactly one queen in each column. What makes the problem difficult is the fact that queens can also move diagonally. Therefore, we have to find a placement with exactly one queen per row and column where no two queens share a diagonal.

To search for a solution to the N-queens problem, there are two fundamentally different techniques. Both techniques search through the space of placements of N queens on a chess board, but in dramatically different ways.

One strategy is referred to as a *global* (or *systematic*) search strategy. In this strategy, a solution is constructed incrementally. That is, we place one queen at a time, starting with one in the first row, then one in the second row etc. For each placement, we look for a position in the row that is not under attack from any of the previously placed queens. One difficulty is that after placing several queens, we may be unable find such an 'open' position in a row (i.e. one that is not being attacked). When we encounter such a situation, say in row i, we need to go back to row $i-1$, and shift the queen in that row to another open position in the row. We may find that even with the queen in the new position in row $i-1$, we still cannot place a queen in row i. We then again shift the queen in the $(i-1)^{\text{th}}$ row to another open position. We may, of course, run out of open positions to move to in row $i-1$, in which case we have to revisit the placement of the queen in row $i-2$, etc. The process of shifting previously placed queens is called *backtracking*. We literally revise or 'backtrack' on our earlier placement choices. Such a search technique will eventually search the full space of all possible placements of queens. So, if a solution exists, it will eventually be found. Unfortunately, the search space is exponentially large and a backtrack technique can therefore be quite inefficient. Using sophisticated heuristics to try the 'most promising' available positions first, one can solve the N-queens problem for up to around 100 queens using backtrack search (Stone and Stone, 1987).

Hill-climbing or *local search* provides a very different search strategy for this problem. In a local search approach, we first start with a 'random' guess at a solution, for example by placing a queen in each row, where the position within a row is chosen randomly. Because of the random placements, it is quite likely that we have one or more pairs of queens that can attack each other either because they share a column or a diagonal. This means that our initial placement does not yet give us a valid solution to the N-queens problem. We now select one of the queens on the board that is under attack from one or more other queens. We move this queen to another location in its row, giving preference to the positions that are not attacked by any other queen. If all positions in a row are under attack, we select a position that is under attack from the least number of queens. If after shifting a queen, which is referred to as a 'local move' or 'local modification', we still have queens under attack, we again select a queen randomly from the ones that are under attack and move that queen, each time trying to further minimize the number of conflicts on the board. This basic strategy is surprisingly effective. It can solve the N-queens problem for over a million queens in less than a minute on a standard PC (Sosic and Gu, 1991; Minton *et al*., 1992).

In general, the key benefits of hill-climbing search are that it requires only a limited amount of memory (only the current state is stored), it is easy to implement efficiently, and, if one or more solutions exists in the search space, hill-climbing search can be surprisingly effective at finding it. Perhaps the first successful use of a hill-climbing strategy was that of Lin and Kernighan (1973; Lin, 1965) for finding good solutions for the traveling salesperson problem. In this problem, the goal is to find the shortest route for a salesperson to take to visit a given set of cities. Starting with an arbitrary tour that visits the cities, Lin and Kernighan showed how one can quickly reduce the length of the tour by making a series of local changes to the route. Since the work by Lin and Kernighan, local search has become one of the main practical techniques for tackling combinatorial optimization problems (Papadimitriou and Steiglitz, 1982).

An important drawback of local search is its inability to detect the unsolvability of a problem instance. That is, if no solution exists, a local search method will simply continue to make local modifications indefinitely. (When dealing with an optimization problem, such as the traveling salesperson problem, the difficulty is that local search cannot be used to determine whether the solution found is globally optimal.) In principle, one could memorize all previously visited problem states and force the local search method to never explore states it has explored before. Provided the local modifications are general enough, such a search would eventually explore the full search space underlying the problem. However, given the combinatorial nature of these problems, this is not feasible on instances of practical interest. Interestingly, for global search techniques, there are memory-efficient ways of keeping track of the space explored so far. Therefore, global search techniques, in contrast to local search methods, can tell us that no solution exists after the method has explored the full search space and no solution has been found.

LOCAL SEARCH STRATEGIES

In hill-climbing search, we are optimizing a certain objective function. For example, in our N-queens problem, our objective function is $O(board) = (N^2/2) - L$, where L is the number of pairs of queens that attack each other on the current board. ($N^2/2$ is the number of pairs of queens.) A solution corresponds to finding a board with the largest possible value for the objective function, i.e. $N^2/2$.

In hill-climbing search, we select any local change that improves the current value of the objective function. *Greedy local search* is a form of hill-climbing search where we select the local move that leads to the largest improvement of the objective function. Traditionally, one would terminate hill-climbing and greedy search methods when no local move could further improve the objective function. Upon termination, the search would have reached a local, but not necessarily global, optimum of the objective function. For many optimization problems, such as the traveling salesperson problem, such a local optimum is quite acceptable, since it often is a reasonable approximation of the global optimum value. However, when a globally optimal solution is required – such as in our N-queens example – local optima present a serious problem for local search methods.

In recent years, it has been found, perhaps somewhat surprisingly, that simply allowing the local search to continue, by accepting 'sideway' or even 'downhill' moves, i.e. local moves to states with, respectively, the same or worse objective values, one can often eventually still reach a global optimum. For example, such 'non-improving' moves are a key component of local search methods for the Boolean satisfiability (SAT) problems, such as GSAT and Walksat (Selman *et al*., 1992, 1994; Gu, 1992). These local search methods for SAT outperform the more traditional backtrack search strategies for a number of SAT problem classes. Many combinatorial problems can be encoded effectively as Boolean SAT problems, so these local search algorithms provide a general mechanism for solving combinatorial problems.

The local search framework allows for a large number of different realizations, and literally hundreds of variants have been explored over the years. The main variants are tabu search, simulated annealing, and genetic algorithms.

Simulated annealing (Kirkpatrick *et al*., 1983) is another example of a local search technique that incorporates sideway and downhill moves. In simulated annealing, downhill moves are accepted with a probability based on the size of the change in the objective function, with the 'worst' moves becoming the least likely. The fraction of accepted local changes is also controlled by a formal parameter, called the 'temperature'. At a high temperature, almost any possible local move is accepted. When lowering the temperature parameter, fewer moves are accepted that worsen the objective value, and the search starts to resemble a purely greedy strategy. In simulated annealing, inspired by the physical process of annealing, the temperature starts high and is slowly lowered during the search process. Simulated annealing has been successfully applied in a wide range of applications.

In tabu search (Glover, 1989), a 'tabu' list is added to the local search method. The tabu list contains the most recent local moves; the local search method is prevented from undoing those modifications. The tabu list is of limited length, and is updated after each local move. Therefore, any particular change is only prevented for a limited period of time. A tabu list provides a powerful way of forcing a local search method to explore a larger part of the search space.

Another form of local search can be found in so-called genetic algorithms (Holland, 1992). Genetic algorithms are inspired by the natural selection process encountered in evolution. A genetic algorithm can be viewed as a strategy for running a large number of local searches in parallel. Aside from local modifications, the states are also modified through a process called 'crossover', in which states from different local searches are combined to provide a (hopefully) better starting point for a new local search.

Hill-climbing search also has a long tradition in the area of continuous optimization. In continuous search spaces, one generally uses the gradient of the objective function to take local steps in the direction of the greatest possible improvement. There are many refinements on the basic gradient search techniques, such as dynamically varying the local step size during the search (Miller, 2000).

In conclusion, hill-climbing search provides a powerful strategy for exploring combinatorial search spaces. In many applications, hill-climbing outperforms global search. In recent work on randomizing backtrack search methods (Gomes *et al*., 2000), we see how some of the ideas from local search are being used to boost the performance of global search methods.

References

Campbell PJ (1977) Gauss and the eight queens problem: a study in miniature of propagation. *Historia Mathematica* **4**: 397–404.

Glover F (1989) Tabu search – part I. *ORSA Journal on Computing* **1**(3): 190–206.

Gomes C, Selman B, Crato N and Kautz H (2000) Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *Journal of Automated Reasoning* **24**(1/2): 67–100.

Gu J (1992) Efficient local search for very large-scale satisfiability problems. *Sigart Bulletin* **3**(1): 8–12.

Holland JH (1992) *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press.

Kirkpatrick S, Gelatt CD and Vecchi MP (1983) Optimization by simulated annealing. *Science* **220**: 671–680.

Lin S (1965) Computer solutions of the traveling salesman problem. *BSTJ* **44**(10): 2245–2269.

Lin S and Kernighan BW (1973) An effective heuristic for the traveling-salesman problem. *Operational Research* **21**: 498–516.

Miller RE (2000) *Optimization*. New York, NY: John Wiley.

Minton S, Johnston M, Philips AB and Laird P (1992) Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence* **58**: 161–205.

Papadimitriou CH and Steiglitz K (1982) *Combinatorial Optimization*. Englewood Cliffs, NJ: Prentice-Hall Inc.

Selman B, Kautz HA and Cohen B (1994) *Noise Strategies for Improving Local Search*. AAAI-94, Seattle, Washington, pp. 337–343.

Selman B, Levesque HJ and Mitchell D (1992) *A New Method for Solving Hard Satisfiability Problems*. Proceedings AAAI-92, San Jose, California, pp. 440–446.

Sosic R and Gu J (1991) 3,000,000 queens in less than one minute. *Bulletin* **2**(2): 22–24.

Stone HS and Stone JM (1987) Efficient search techniques – an empirical study of the *N*-queens problem. *IBM Journal of Research and Development* **31**: 464–474.

Further Reading

Aarts EHK and Lenstra JK (1997) *Local Search in Combinatorial Optimization*. London, UK: John Wiley.

Garey MR and Johnson DS (1979) *Computers and Intractability, A Guide to the Theory of NP-Completeness*. New York, NY: WH Freeman.
Gent I, van Maaran H and Walsh T (eds) (2000) *SAT 2000*. IOS Press.
Papadimitriou C and Steiglitz K (1998) *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications.
Rayward-Smith VJ, Osman IH and Reeves CR (eds) (1996) *Modern Heuristic Search Methods*. London, UK: John Wiley.
Sait SM and Youssef H (1999) *Iterative Computer Algorithms with Applications in Engineering*. 1EEE Computer Science Press.

Hippocampus

Intermediate article

John O'Keefe, University College London, UK

CONTENTS

The hippocampus is a cortical structure located in the temporal lobes of the brain. Its role in memory has been extensively investigated in humans and other animals.

INTRODUCTION

There is universal agreement that the hippocampus, a cortical structure located in the temporal lobes, is involved in memory. Theories of hippocampal function are based on studies of amnesic patients and experimental animals. The cognitive map theory is the most clearly specified and is supported by the most evidence, but the declarative memory and flexible relational theories also have their proponents.

ANATOMY

Historically, the hippocampal formation has been considered to be a part of the limbic system, a set of structures traditionally believed to be constituents of the emotional brain. It was also included in the Papez circuit, a network of structures envisaged by Papez to subserve emotion. As we shall see, there is good evidence that the hippocampal formation and the Papez circuit are not involved in emotion but instead are part of the memory system of the brain. In humans, the hippocampus is an elongated structure lying on the medial surface of the temporal lobe (Figure 1a). In rats, it is more banana-shaped and extends dorsally upwards from the temporal lobes to lie beneath the parietal cortex (Figure 1b). There are two major pathways by which the hippocampus communicates with the rest of the brain: the fornix–fimbria fiber bundle which connects it with the septal nuclei and the hypothalamus, and the perforant path which connects it with the cerebral cortex. The fornix–fimbria conveys information to the hippocampal formation about the animal's movements, attentional and bodily states and takes information from the hippocampus to control movements, in particular those which take the animal towards goals or away from punishments. An important function of the fornical input is to provide both the cholinergic and γ-aminobutyric acid (GABA) inputs which are a necessary condition for the theta electroencephalographic (EEG) state of the hippocampus (see below). Sensory information about the external world enters the hippocampus through the perforant path, which arises in the entorhinal cortex. The entorhinal cortex in turn receives information from other parts of the temporal lobe, in particular the perirhinal and parahippocampal cortices and ultimately from many of the sensory analysing regions of the cerebral cortex. These areas in the parahippocampal gyrus appear to be responsible for at least some aspects of the