Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Add links

Min-conflicts algorithm

From Wikipedia, the free encyclopedia
(Redirected from Min conflicts algorithm)

In computer science, the **min conflicts algorithm** is a search algorithm or heuristic method to solve constraint satisfaction problems (CSP).

Given an initial assignment of values to all the variables of a CSP, the algorithm randomly selects a variable from the set of variables with conflicts violating one or more constraints of the CSP.[1] Then it assigns to this variable the value with that minimizes the number of conflicts. If there is more than one value with a minimum number of conflicts, it chooses one randomly. This process of random variable selection and min-conflict value assignment is iterated until a solution is found or a pre-selected maximum number of iterations is reached.

Because a CSP can be interpreted as a local search problem when all the variables have an assigned value (called a complete state), the min conflicts algorithm can be seen as a repair heuristic[2] that chooses the state with the minimum number of conflicts.

Algorithm [edit]

```
 algorithm MIN-CONFLICTS
     input: csp, a constraint satisfaction problem
            max_steps,the number of steps allowed before giving up
            current_state, an initial assignment of values for the variables in the
csp
     output: a solution set of values for the variable or failure
   for i=1 to max_steps do
       if current_state is a solution of csp then return current_state
       var <-- a randomly chosen variable from the set of conflicted variables
CONFLICTED[csp]
       value <-- the value v for var that minimizes CONFLICTS(var,v,current,csp)
       set var = value in current_state
   return failure
```

Although not specified in the algorithm, a good initial assignment can be critical for quickly approaching a solution. Use a greedy algorithm with some level of randomness and allow variable assignment to break constraints when no other assignment will suffice. The randomness helps min-conflicts avoid local minima created by the greedy algorithm's initial assignment. In fact, Constraint Satisfaction Problems that respond best to a min-conflicts solution do well where a greedy algorithm almost solves the problem. Map coloring problems do poorly with Greedy Algorithm as well as Min-Conflicts. Sub areas of the map tend to hold their colors stable and min conflicts cannot hill climb to break out of the local minimum. The CONFLICTS function counts the number of constraints violated by a particular object, given that the state of the rest of the assignment is known.

History [edit]

Although Artificial Intelligence and Discrete Optimization had known and reasoned about Constraint Satisfaction Problems for many years, it was not until the early 1990s that this process for solving large CSPs had been codified in algorithmic form. Early on, Mark Johnston of the Space Telescope Science Institute looked for a method to schedule astronomical observations on the Hubble Space Telescope. In the process, he created a neural network capable of solving a toy N-queens problem (for 1024 queens). Steven Minton and Andy Philips analyzed the neural network algorithm and separated it into two phases: (1) an initial assignment using a

greedy algorithm and (2) a conflict minimization phases (later to be called "min-conflicts"). A paper was written and presented at AAAI-90; Philip Laird provided the mathematical analysis of the algorithm.

Subsequently, Mark Johnston and the STScI staff used min-conflicts to schedule astronomers' experiment time on the Hubble Space Telescope.

Example [edit]

Min-Conflicts solves the N-Queens Problem by randomly selecting a column from the Chess board for queen reassignment. The algorithm searches each potential move for the number of conflicts (number of attacking queens), shown in each square. The algorithm moves the queen to the square with the minimum number of conflicts, breaking ties randomly. Note that the number of conflicts is generated by each new direction that a queen can attack from. If two queens would attack from the same direction (row, or diagonal) then the conflict is only counted once. Also note that if a queen is in a position in which a move would put it in greater conflict than its current position, it does not make a move. It follows that if a queen is in a state of minimum conflict, it does not have to move.

This algorithm's run time for solving N-Queens is independent of problem size. This algorithm will even solve the *million-queens problem* on average of 50 steps. This discovery and observations lead to a great amount of research in 1990 and began research on local search problems and the distinctions between easy and hard problems. N-Queens is easy for local search because solutions are densely distributed throughout the state space. It is also effective for hard problems. For example, it has been used to schedule observations for the Hubble Space Telescope, reducing the time taken to schedule a week of observations from three weeks to around 10 minutes.[3]

See also [edit]

- Warnsdorff's algorithm
- Eight queens Puzzle
- Guided Local Search

References [edit]

1. ^ Minton, Steven; Mark D. Johnston; Andrew B. Philips; Philip Laird (1990). "Solving Large-Scale Constraint Satisfaction and Scheduling Problems Using a Heuristic Repair Method" (PDF). *Eighth National Conference on Artificial Intelligence (AAAI-90), Boston, Massachusetts*: 17–24. Retrieved 27 March 2013.
2. ^ Minton, Steven; Mark D. Johnston; Andrew B. Philips; Philip Laird (1992). "Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems" (PDF). *Artificial Intelligence* **58** (1): 161–205. doi:10.1016/0004-3702(92)90007-k. Retrieved 27 March 2013.
3. ^ Stuart Russell, Peter Norvig, "Artificial Intelligence: A Modern Approach (3rd Edition)", pp. 220-222, December 11, 2009.

- Stuart J. Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*

External links [edit]

- A visualization of the N-Queens puzzle solved using the min-conflicts algorithm. Created by Yuval Baror.
- [1] The min-conflicts heuristic microform : experiment and theoretical results / Steven Minton ... [et al.]. NASA, Ames Research Center, Artificial Intelligence Research Branch. Distributed to depository libraries in microfiche.

Categories: Constraint programming