Universidad de Cádiz
Escuela Superior de Ingeniería
Computer Science Engineering

Comparative study: local search, simulated annealing and taboo search

Intelligent Systems

Lucía Atienza Olmo

Cádiz
March 2022

Abstract

The n-Queens problem is to place n chess queens on an n by n chessboard so that no two queens are on the same row, column or diagonal. This puzzle dates to 1848, and only two years later a variant was introduced by Nauck [1850] in which some number of queens are pre-placed and the solver is asked to place the rest, if possible. The n-Queens problem has been very widely used as a benchmark in Artificial Intelligence(AI). However, because of the ease of finding a solution, the n-Queens problem has been subject to repeated controversy in AI over whether it should be used as a benchmark at all.

Introduction

The purpose of this practice is to analyze the behavior of n queens with different search algorithms. This problem consists in putting n queens on an n x n board with no two queens on the same row, column, or diagonal. The search algorithms used to test the performance have been:

- Local Search
- Simulated Annealing
- Taboo Search

It consists of looking for a solution to the problem with different implementations (the above mentioned) and see how the puzzle behaves. The objectives of the practice are to learn how local search, simulated anneling and taboo search work, learn how to implement them and analyze the results obtained.

For all random generations one seed has been used, using `rng(n_iter)`, with $n\_iter >= 1 \wedge n\_iter <= 1000$

Local Search

1. Start with initial configuration. In our case, we start from a random configuration by using `randperm`.
2. Repeatedly search successors and select the best neighbor as candidate
3. Apply a cost function (fEval) and accept candidate if it is better than current
4. Stop if quality is sufficiently high or if no improvement can be found

Simulated Annealing

As in the previous section, we start with a random initial configuration. However, contrary to local search, simulated annealing does not choose the best successor, but a random one. For this algorithm it is necessary to control the temperature and for this we have used Cauchy

Taboo Search

The essence of this algorithm is: in each iteration the best successor (no taboo) is selected. Now, the current is included into the taboo list for a certain number of iterations (4 have been chosen).

And if a successor is taboo, then we take the next one. In the code you can see that we continue searching among the successors of the current state as long as new states are being generated by using the boolean variable new_state.

Results and conclusion

In the following page we can see the results obtained after having performed 1000 iterations with each algorithm using 12 and 24 queens states. I honestly don't understand how taboo can come up with a solution to the problem in such a short time and iterations.

We can see that, for example, local search is faster than annealing an taboo, as theory tells us. It should be noted that annealing loses many hits as we increase the size of the problem, as well as how long it takes to finish. The latter may be due to a wrongly chosen temperature.

	N=12			N=24		
	Local	Annealing	Taboo	Local	Annealing	Taboo
N times	111	223	1000	20	1	1000
Time (mean)	0.6080678	6.855447	1.4260908	2.9434521	16.388241	8.2719315
Iterations (mean)	27.715	934.123	7.5450	57.023	1000.531	9.329