VEDANT PAITHANE, 2017MEB1247

Performance metrics:

	Backtrack				Min Conflict			
	start time	finish time	total time	nodes generated	start time	finish time	total time	nodes generated
Eg 1	158696 4141	158696 4141	0.00801 0149	94	158696 4180	158696 4180	0.09075 9993	384
Eg 2	158696 4454	158696 4454	0.02197 0034	911	158696 4486	158696 4487	0.52276 0153	2176
Eg 3	158696 5945	158696 5945	0.03719 9974	1619	158696 6043	158696 6044	0.42958 9987	1484
Eg 4	158696 7038	158696 7038	0.01599 0019	542	158696 7076	158696 7076	0.16778 9936	492
Eg 5	158696 7306	158696 7306	0.01137 9957	291	158696 7309	158696 7309	0.11868	1010

Analysis:

As such, the Backtracking algorithm is expected to perform better, since it completes the puzzle in sequential manner while the min conflict uses trial and error while reducing overall cost. This is also the case for this example. For simpler problems, backtracking arrives at the solution quicker, however it also depends on the initial solution selected by the min conflict algorithm. Also, as the problem becomes harder, min conflict starts to perform better, but still takes much more time than backtracking, while backtracking's time increases at a fairly linear rate. The same can be seen in the number of nodes generated as well.

Min Conflict is also expected to get stuck at local minimums in some cases. To counter that, I have used the random module to at a variability rate so that in case it gets stuck, the values can be changed until the algorithm reaches the global minimum. Thus, the algorithm reaches a solution in every case, albeit taking more time.

Test case 1:

		2	1		6			4
1					9			6
		4		5		1	8	3
	4							
	1		9	2		3	7	8
2	3		5		8	4		
			8	7	5	2	9	1
8	5		4	9	2	6		
		9			1	8		5

```
[3, 7, 2, 1, 8, 6, 9, 5, 4]
[1, 8, 5, 3, 4, 9, 7, 2, 6]
[6, 9, 4, 2, 5, 7, 1, 8, 3]
[9, 4, 8, 7, 1, 3, 5, 6, 2]
[5, 1, 6, 9, 2, 4, 3, 7, 8]
[2, 3, 7, 5, 6, 8, 4, 1, 9]
[4, 6, 3, 8, 7, 5, 2, 9, 1]
[8, 5, 1, 4, 9, 2, 6, 3, 7]
[7, 2, 9, 6, 3, 1, 8, 4, 5]
Backtrack
```

```
[3, 7, 2, 1, 8, 6, 9, 5, 4]
[1, 8, 5, 3, 4, 9, 7, 2, 6]
[6, 9, 4, 2, 5, 7, 1, 8, 3]
[9, 4, 8, 7, 6, 3, 5, 1, 2]
[5, 1, 6, 9, 2, 4, 3, 7, 8]
[2, 3, 7, 5, 1, 8, 4, 6, 9]
[4, 6, 3, 8, 7, 5, 2, 9, 1]
[8, 5, 1, 4, 9, 2, 6, 3, 7]
[7, 2, 9, 6, 3, 1, 8, 4, 5]
Min conflict solution
```

1) Both algorithms reach the solution.
2) Backtracking finishes the puzzle way faster than Min conflict and also uses lesser nodes and lesser space than min conflict, as expected.

Test case 2:

5	3		1	7				4
	6	1			9			
			5			7		
					3	8		
8	2	3	4					
1	7						4	2
		7			1	4	8	6
			6					
	1		7	5		2		

```
[5, 3, 2, 1, 7, 8, 6, 9, 4]
[7, 6, 1, 3, 4, 9, 5, 2, 8]
[4, 8, 9, 5, 2, 6, 7, 1, 3]
[9, 4, 5, 2, 1, 3, 8, 6, 7]
[8, 2, 3, 4, 6, 7, 9, 5, 1]
[1, 7, 6, 8, 9, 5, 3, 4, 2]
[2, 5, 7, 9, 3, 1, 4, 8, 6]
[3, 9, 4, 6, 8, 2, 1, 7, 5]
[6, 1, 8, 7, 5, 4, 2, 3, 9]
Backtrack
```

```
[5, 3, 8, 1, 7, 2, 6, 9, 4]
[7, 6, 1, 8, 4, 9, 5, 2, 3]
[2, 4, 9, 5, 3, 6, 7, 1, 8]
[4, 9, 5, 2, 6, 3, 8, 7, 1]
[8, 2, 3, 4, 1, 7, 9, 6, 5]
[1, 7, 6, 9, 8, 5, 3, 4, 2]
[9, 5, 7, 3, 2, 1, 4, 8, 6]
[3, 8, 2, 6, 9, 4, 1, 5, 7]
[6, 1, 4, 7, 5, 8, 2, 3, 9]
Min conflict solution
```

1) Both algorithms solved the problem.
2) Backtracking vastly outperformed min conflict here, performing about 25 times better while also using fewer nodes.

Test case 3:

					8		4	
	9		6			8		3
			1	5				
2			7	3			6	
4		6		1			7	
7		5			1			
8								
9	6	3	8			5		

```
[1, 2, 3, 9, 7, 8, 6, 4, 5]
[5, 9, 7, 6, 4, 2, 8, 1, 3]
[6, 4, 8, 1, 5, 3, 7, 2, 9]
[2, 5, 1, 7, 3, 4, 9, 6, 8]
[4, 8, 6, 5, 1, 9, 3, 7, 2]
[3, 7, 9, 2, 8, 6, 1, 5, 4]
[7, 3, 5, 4, 9, 1, 2, 8, 6]
[8, 1, 2, 3, 6, 5, 4, 9, 7]
[9, 6, 4, 8, 2, 7, 5, 3, 1]
Backtrack
```

```
[5, 3, 7, 9, 2, 8, 1, 4, 6]
[1, 9, 2, 6, 4, 7, 8, 5, 3]
[6, 4, 8, 1, 5, 3, 7, 2, 9]
[2, 5, 1, 7, 3, 4, 9, 6, 8]
[4, 8, 6, 5, 1, 9, 3, 7, 2]
[3, 7, 9, 2, 8, 6, 4, 1, 5]
[7, 2, 5, 3, 9, 1, 6, 8, 4]
[8, 1, 3, 4, 6, 5, 2, 9, 7]
[9, 6, 4, 8, 7, 2, 5, 3, 1]
Min conflict solution
```

1) Both algorithms reach the solution.
2) Backtracking takes much less time but creates more nodes as compared to the min conflict algorithm. One explanation could be that the algorithm could have been stuck on a local minimum, which consumed some of the time while not increasing the number of nodes by a

large amount as the initial solution could have been close to the output, but the algorithm would have been stuck on the minima before changing the values.

Test case 4:

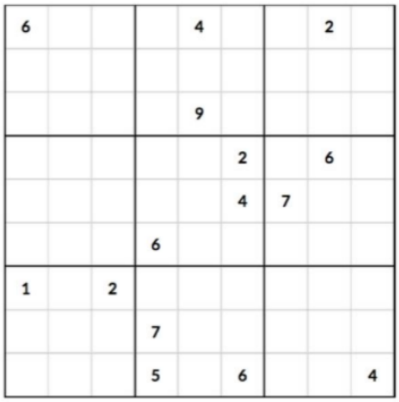
3			5		9			4
								6
5		1	4			7	3	
				8	5		4	
		7						
							9	3
	9	8						
6					7			

```
[3, 2, 6, 5, 7, 9, 1, 8, 4]
[4, 7, 9, 1, 3, 8, 2, 5, 6]
[5, 8, 1, 4, 2, 6, 7, 3, 9]
[1, 3, 5, 2, 6, 4, 9, 7, 8]
[9, 6, 2, 7, 8, 5, 3, 4, 1]
[8, 4, 7, 3, 9, 1, 5, 6, 2]
[7, 5, 4, 8, 1, 2, 6, 9, 3]
[2, 9, 8, 6, 5, 3, 4, 1, 7]
[6, 1, 3, 9, 4, 7, 8, 2, 5]
Backtrack
```

```
[3, 2, 6, 5, 7, 9, 1, 8, 4]
[4, 7, 9, 8, 3, 1, 2, 5, 6]
[5, 8, 1, 4, 6, 2, 7, 3, 9]
[9, 4, 5, 7, 1, 3, 8, 6, 2]
[1, 3, 2, 6, 8, 5, 9, 4, 7]
[8, 6, 7, 9, 2, 4, 3, 1, 5]
[7, 1, 4, 2, 5, 8, 6, 9, 3]
[2, 9, 8, 3, 4, 6, 5, 7, 1]
[6, 5, 3, 1, 9, 7, 4, 2, 8]
Min conflict solution
```

1) Both algorithms reach the solution.
2) Backtracking takes much less time but creates more nodes as compared to the min conflict algorithm. One explanation could be that the algorithm could have been stuck on a local minimum, which consumed some of the time while not increasing the number of nodes by a large amount as the initial solution could have been close to the output, but the algorithm would have been stuck on the minima before changing the values.

Test case 5:




```
[6, 1, 3, 8, 4, 5, 9, 2, 7]
[2, 4, 9, 1, 6, 7, 3, 5, 8]
[5, 7, 8, 2, 9, 3, 1, 4, 6]
[3, 5, 4, 9, 7, 2, 8, 6, 1]
[8, 2, 6, 3, 1, 4, 7, 9, 5]
[7, 9, 1, 6, 5, 8, 4, 3, 2]
[1, 6, 2, 4, 8, 9, 5, 7, 3]
[4, 3, 5, 7, 2, 1, 6, 8, 9]
[9, 8, 7, 5, 3, 6, 2, 1, 4]
Backtrack
```

```
[6, 7, 9, 1, 4, 5, 3, 2, 8]
[8, 4, 3, 2, 6, 7, 9, 1, 5]
[2, 1, 5, 8, 9, 3, 6, 4, 7]
[7, 9, 4, 3, 5, 2, 8, 6, 1]
[3, 6, 8, 9, 1, 4, 7, 5, 2]
[5, 2, 1, 6, 7, 8, 4, 9, 3]
[1, 3, 2, 4, 8, 9, 5, 7, 6]
[4, 5, 6, 7, 3, 1, 2, 8, 9]
[9, 8, 7, 5, 2, 6, 1, 3, 4]
Min conflict solution
```

1) Both algorithms reach the solution.
2) Backtracking finishes the puzzle way faster than Min conflict and also uses lesser nodes and lesser space than min conflict, as expected