Software report

Buciu Ștefan, ICA 256

1. Sudoku

* Problem definition

TODO

* CSP representation

TODO

* Solution

TODO

1. Maximum Cut

* Problem definition

The Maximum Cut problem is NP-complete, and it aims to partition an undirected graph into two sets such that the number of edges that cross the cut is maximized (i.e., one end of the edge in each set). We will be trying to solve this problem version, however there is a more general version where the edges have a positive weight assigned.

* CSP representation

We can represent the MaxCut problem as follows:

* Variable: given a graph with a set of vertices , a variable will be assigned for each vertex.
* Domain: after a cut, a vertex can be part of a set , thus the domain for each variable is .
* Constraints: after a cut, a set of edges will be selected . Given a selected edge , then the vertices connected by it and must have variables assigned such that .
* Solution
* Backtracking:

We solve MaxCut with backtracking by iterating over the given edges and keeping track of the partitioning. If we select edge , then each vertex is added to a set and . However, since the graph is undirected, we check both configurations (i.e., adding to S and to T).

A solution is correct if the constraints are satisfied (i.e., ). If we have a solution, we then compute the size of the cut by counting how many edges have the first vertex in a set, and the second in the other set.

This method is implemented in the class *MaxCutBacktracking* in the Java source code provided.

* Problem Reduction ():
* Basic Search Strategy ():