(67686) Mathematical Foundations of AI May 14, 2008

Lecture 5

Lecturer: Ariel D. Procaccia *Scribe: Rivka Oster, Noam Aigerman, Dan Friedman*

1 Constraint satisfaction problems: the structure of the constraint graph

Constraint satisfaction problems (CSPs) are problems in which variables must be assigned values subject to specific constraints. We define them formally using as a triplet $\langle X, D, C\rangle$, where X is a set of variables, D is a domain of values, and C is a set of constraints $C_1(S_1)..C_n(S_n)$ where each S_i is a set of variables. A constraint C_i is a combination of valid values for the variables S_i. A solution to the CSP is an assignment of values to all the variables that satisfies all constraints.

1.1 Example

Let the triplet $\langle\{x, y, z\}, \{2, 4, 5, 6, 10\}, C\rangle$, where the constraints can be described as $x \in \{5, 2, 4, 6\}$, $y \in \{10, 4, 6, 2\}$, $z \in \{5, 2, 4, 6\}$, $z|x$, $z|y$.

2 Solutions to the problem :

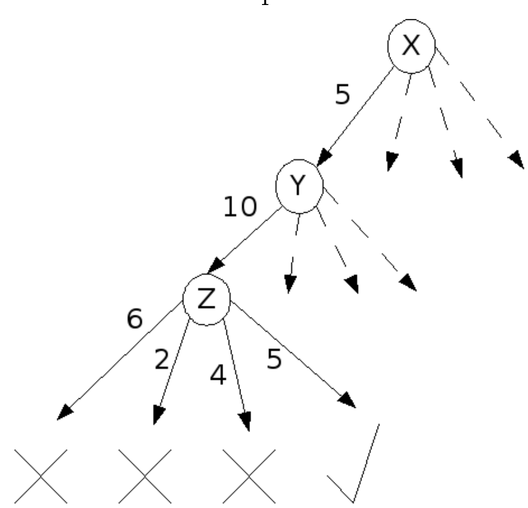

Figure 1: Recursively assign values to x, y, z in that order, until we find a solution which satisfies all the constraints.Notice that in this example numerous redundant backtrackings are made.

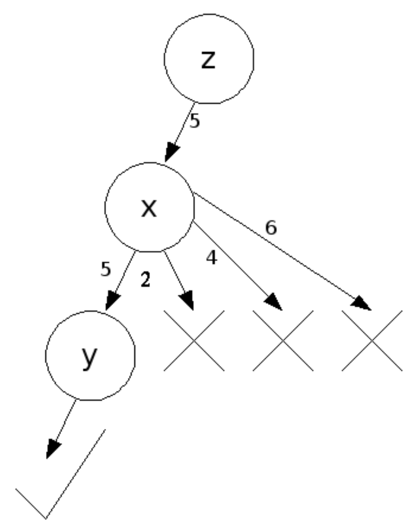

Figure 2: We execute the same recursive solution, but this time assigning the values in the order z, y, x. Notice how the different ordering of moves, results in a more efficient solution to the problem.

Definition 1 *A **vertical order** is the order in which variables are chosen for instantiation.*

Definition 2 *A **horizontal order** is the order in which values are tested for a given variable.*

Definition 3 *A **constraint graph** for such a problem is an undirected graph where the nodes represent variables and two nodes are linked to represent the existence of a constraint which involves these variables (and possibly others). At most a single edge will join two nodes, even if they are related by more than one constraint.*

Definition 4 *An **ordered constraint graph** arranges the nodes in a linear order. (Notice that the number of ordered constraint graphs is $n!$ where n is the number of variables).*

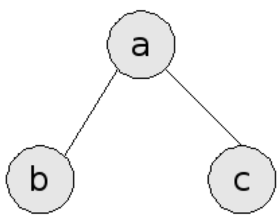

Figure 3: The original graph

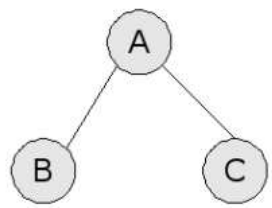

Figure 4: The constraint graph

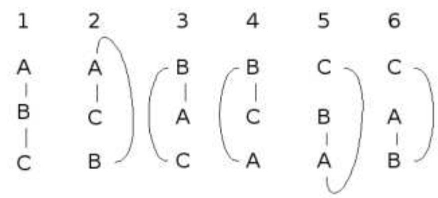

Figure 5: The ordered constraint graphs

Observe the graph in figure 3, applied to a graph 2-coloring problem. Let A, B, C be the variables that represent the color of the nodes a, b, c respectively. The constraint graph of this CSP is shown in figure 4, and the 3! different ordered constraint graphs are shown in figure 5.

Definition 5 *The **width of a vertex** in an ordered constraint graph is defined to be the number of edges from this vertex to its predecessors in that order. For example, the width of the vertex A in the fifth ordered constraint graph in figure 5, is 2.*

Definition 6 *The **width of an ordered constraint graph** is the maximum of the widths of its vertices. For example, the width of the first ordered constraint graph in figure 5 is 1.*

Definition 7 *The **width of a constraint graph** is the minimum of the widths of all the possible orders. For example, the width of the graph in figure 4 is 1.*

Definition 8 *A vertical order is **backtrack-free** if it guarantees a backtrack-free search regardless of the horizontal order of search.*[1]

Definition 9 *A CSP is **k consistent** if every legal partial assignment of $k-1$ variables can be expanded to a legal assignment of k variables, for any additional variable.*

[1]Notice that an order that is globally consistent is also backtrack-free, but not necessarily the other way around.

Definition 10 *A CSP is* ***strongly k consistent*** *if it is i consistent for every $i \leq k$.*

Definition 11 *A CSP is* ***globally consistent*** *if it is k consistent for every k.*

Theorem 1 *Given a constraint satisfaction problem:*

1. *A vertical search order is backtrack-free if the level of strong consistency is greater than the width of the corresponding ordered constraint graph.*

2. *There exists a backtrack-free vertical search order for the problem if the level of strong consistency is greater than the width of the constraint graph.*

Proof

1. Let o be a vertical order, and let the width of the constraint graph be k. Let v_i be the i'th variable in o. Notice that v_i is constrained by no more than k of the previously assigned variables (Because the ordered constraint graph has width of k). The problem is at least strongly $k+1$-consistent, and therefore every legal assignment to k or less variables can be completed to a legal assignment to those variables and to v_i. It follows by induction that o is backtrack-free.

2. Let the the level of strong consistency be greater than k, and let the width of the constraint graph be k. By definition, it means that there is an order o such that o's width is k. According to (1) o is backtrack-free.

■

2 The relation between the width and the connective structure of a graph

2.1 Definitions

Definition 12 *The* ***linkage*** *of a subgraph H is the maximal k such that every vertex has a degree of at least k.*

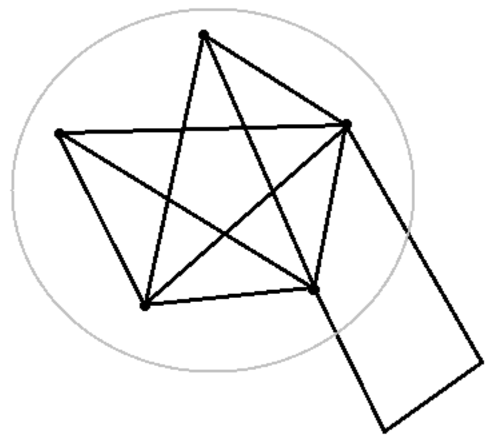

Figure 6: The subgraph defined by the circle, has a linkage of 3, since all vertices in the subgraph are connected to at least 3 other vertices

Theorem 2

The width of a constraint graph is equal to the maximal linkage of its subgraphs.

Lemma 1

Let w be the width of the constraint graph G. $\forall k \; w \geq k \iff$ there exists a subgraph of G with linkage $l \geq k$.

Proof of Lemma

$\Longleftarrow$

Let H be a subgraph of G with linkage $l \geq k$. For a given order O of the vertices, Let v be the last vertex in O. All other vertices are ordered higher up, and v is connected to at least k of them. Therefore, v is a vertex with width w such that $w \geq k$. This can be shown for every possible ordering, and thus the total width of the graph is at least k.

$\Longrightarrow$

We assume for contradiction that every subgraph of G has a linkage $< k$. The complete graph G is a subgraph of itself and thus also has a linkage $< k$. There exists a vertex v that is connected to less than k other vertices in G. Let O be an ordering in which v is the vertex ordered last. Let $G' = G \backslash \{v\}$. From the assumption of the proof, G' also has a linkage $< k$, thus it also has a vertex v' that is connected to less than k of the vertices in G'. We place v' before v in O, and remove v' from G'. Repeating this process $|G|$ times, will yield an ordering that has a width of less than k, in contradiction with the assumption that G has a width of at least k.

■

Proof

$\Longrightarrow$

We conclude from the lemma, that if a graph G has a width of k, then there exists a subgraph

G' with a linkage of at least k. If there had been a subgraph with a linkage larger than k, then from the lemma we would conclude that G also has a width larger than k, which leads to a contradiction since G has a width of k. Thus we conclude that the maximal linkage of a subgraph of G is k

$\Longleftarrow$

Similarly, let G' be the subgraph of G that has the maximum linkage, and let its linkage be k. From the lemma, we conclude that G has a width of at least k. G can't have a width of more than k, since from the lemma this would entail that there exists a subgraph with a linkage larger than k, contrary to the assumption. Thus the width of G is exactly k. ■

We describe an algorithm that given the width k of the constraint graph, finds an ordering with a width of k.

Algorithm 1 Find-Ordering(k)

for $i = n$ to 1

1. find a vertex that is connected to at most k other vertices (One must exist according to the theorem).
2. put the vertex in the i'th place in the ordering, and remove it from the graph.

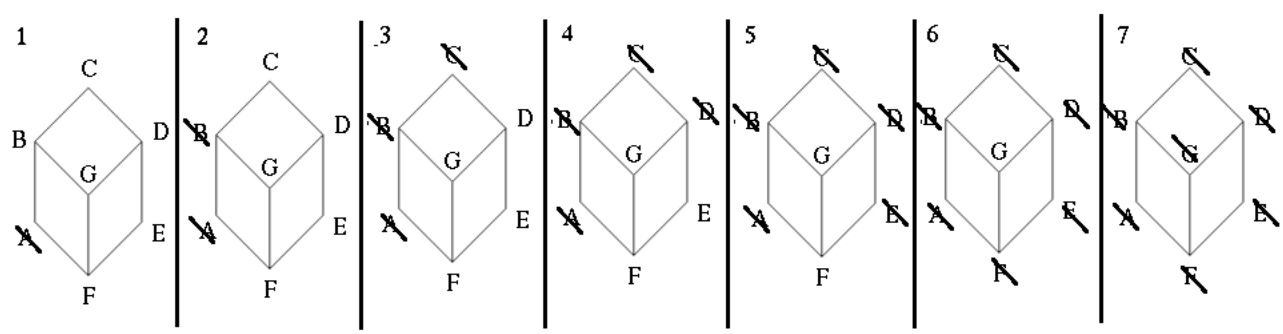

Figure 7: An example for Algorithm 1. Each figure corresponds to an iteration of the algorithm. The final ordering can be seen in figure number 8

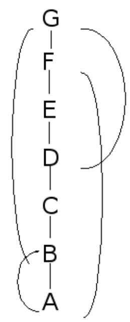

Figure 8: The resulting ordering achieved

We describe an algorithm that given graph G, finds the subgraph of G with the maximal linkage.

Algorithm 2 Find-Maximal-Linkage(G)

1. $k \leftarrow 0$
2. Remove from the graph all vertices of degree 0
3. **while** There are still vertices in the graph
   $k = k + 1$
   while There are still vertices in the graph with degree $< k + 1$
   remove them.
4. return k.

Theorem 3 *For a given constraint graph G, the width of the graph $\leq 1 \Leftrightarrow$ the graph is a forest.*

Proof A forest is a graph without cycles. According to theorem 2, it is enough to prove that the maximal linkage of all the subgraphs of $G \leq 1 \Leftrightarrow$ there are no cycles.

$\Rightarrow$ Assume that a cycle exists in G. The cycle itself is a subgraph with a linkage of 2, thus the maximal linkage of the graph > 1

$\Leftarrow$ Assume that the maximal linkage of any subgraph of $G \geq 2$ Let H be a subgraph with linkage ≥ 2, and let $v_1 \in H$. v_1 is necessarily connected to another vertex, $v_2 \in H$. v_2 is connected to necessarily connected to another vertex, $v_3 \in H$, such that $v_1 \neq v_3$. We continue with this procedure, building a chain in H. Since the size of H is finite, we would have to eventually complete a cycle. ■

We return to example described at the beginning of the article: $x \in \{5, 2, 4, 6\}$, $y \in \{2, 4, 6, 10\}$, $z \in \{5, 2, 4, 6\}$. We are looking for an assignment of x,y,z such that $z|x \wedge z|y$. There is an ordering, $z - x - y$, with width of 1. The CSP as presented is strongly 2 consistent. Thus, from Theorem 1 , we conclude that the ordering $z - x - y$ is free from backtracking.

3 Usages in coloring of graphs

Theorem 4 *Every planar graph has a width of at most* 5.

Proof It is known that in every planar graph, there is a vertex with degree ≤ 5. We try to find an ordering with width ≤ 5: We remove the node with degree ≤ 5, and insert it last in our ordering. We are left with a planar graph, and necessarily there exists another vertex with degree ≤ 5. We continue with this algorithm, until there are no vertices left in the graph. We have created an ordering in which every vertex is connected to at most 5 vertexes which are ordered before him. Thus we conclude that the overall width of the graph is at most 5. ■

Corollary 1 *It is possible to color any planar graph in 6 colors, without backtracking, since if there are 6 colors, than the CSP is strongly 6-consistent.*

Corollary 2 *The width of the problem* $+1$ *is an upper bound on the chromatic number of the graph, where the chromatic number of a graph is the minimal number of colors needed to perform a coloring. This is true, since if there are* l *colors, then the problem is obviously at least strongly* l *consistent.*

References

[1] E.C. Freuder, A sufficient condition of backtrack-free search. *Journal of the ACM 29 (1982)*, pp. 24-32.