Many problems that are NP-Hard for arbitrary graphs can be solved efficiently for chordal graphs. Here we give some examples of these problems. Most of the algorithms use either the perfect elimination ordering or the representation of the chordal graph as the intersection graph of subtrees of a tree.

The four most basic problems are chromatic number, maximum independent set, maximum clique and minimum clique cover. These can be solved in $O(n+m)$ time using the perfect elimination ordering.

Assume the vertices of the chordal graph are numbered 0 to $n-1$ in the perfect elimination ordering. Let $N^+[i]$ denote the neighbors of $i$ that are greater than $i$ and $N^-[i]$ the neighbors that are less than $i$.

**Maximum Independent Set**

The algorithm can be described as:

$I = \emptyset$
for $i = 0$ to $n-1$
if $(N^-[i] \cap I = \emptyset)$ then $I = I \cup \{i\}$.

The proof of this follows the standard pattern of proof for greedy algorithms. Suppose there exists a maximum independent set $X$ that agrees with the greedy solution for vertices $\{0,\dots,i-1\}$. Then show that there exists a maximum independent set that agrees with the greedy solution for vertices $\{0,\dots,i\}$.

If $i$ was not included in the greedy solution, some vertex in $N^-[i]$ is in the greedy solution and hence also in $X$. Thus $i \notin X$ and $X$ agrees with greedy for vertices $\{0,\dots,i\}$.

If $i$ was included in greedy, then no vertex in $N^-[i]$ is in greedy, and hence there is no such vertex in $X$ either. If $i \in X$, then $X$ agrees with greedy for vertices $\{0,\dots,i\}$. If $i \notin X$, some vertex in $N^+[i]$ must be included in $X$, otherwise $X \cup \{i\}$ is an independent set, contradicting the fact that $X$ is a maximum independent set. Since the ordering is a perfect elimination ordering, any two vertices in $N^+[i]$ are adjacent. Thus there is exactly one vertex $j \in X \cap N^+[i]$. Replacing $j$ by $i$ in $X$ gives another maximum independent set that agrees with greedy for vertices $\{0,\dots,i\}$.

An easier way to prove is to simultaneously construct a clique cover of the same size as the independent set. This implies that the independent set has maximum size and the clique cover has minimum size.

$I = \emptyset$
for $i = 0$ to $n-1$
if $(N^-[i] \cap I = \emptyset)$ then $I = I \cup \{i\}$ and add a new part $\{i\}$ to the partition else add $\{i\}$ to a part containing any vertex in $N^-[i] \cap I$.

Since a vertex in $I$ is the smallest vertex in its part, and all vertices in its part are adjacent to it, the perfect elimination property implies that each part induces a complete subgraph. Thus the number of cliques in the clique cover equals the size of the independent set.

A similar algorithm works for the chromatic number and the maximum clique.

for $i = n-1$ to 0
assign vertex $i$ smallest possible color that is not assigned to any vertex in $N^+[i]$.

This gives a coloring with minimum number of colors. If a vertex $i$ is assigned color $k$, then it must have neighbors in $N^+[i]$ already assigned colors 1 to $k-1$. Thus there is a complete subgraph of size $k$. The number of colors used is thus at most the size of the maximum clique and hence is minimum.

For weighted versions of these problems, the representation as the intersection graph of subtrees of a tree is needed. However, this is also implicitly given by the perfect elimination ordering, where the parent of vertex $i$ is the smallest vertex in $N^+[i]$. The subtree corresponding to $i$ contains $i$ and all vertices in $N^-[i]$. The underlying tree is assumed to be rooted at vertex $n-1$. The descendants of $i$ in the tree are all those vertices that are reachable from $i$ by a monotonically decreasing path. Let $G[i]$ denote the subgraph of $G$ induced by descendants of $i$ and let $S_1[i]$ be the weight of maximum weight independent set in $G[i]$ that excludes $i$ and $S_2[i]$ the weight of maximum weight independent set including $i$. Let $S[i]$ be the maximum of $S_1[i], S_2[i]$. A child of $i$ is a vertex $j < i$ that is adjacent to $i$ but not to any vertex $k, j < k < i$. Let $C[i]$ denote the set of children of $i$.

Then we have the recurrences:

$$
\begin{aligned}
S_1[i] &= \sum_{j \in C[i]} S[j] \\
S_2[i] &= wt(i) + S_1[i] + \sum_{j \in N^-[i]} (S_1[j] - S[j]) \\
S[i] &= \max(S_1[i], S_2[i])
\end{aligned}
$$

If $i$ has no children the sums are assumed to be zero.

This can be easily computed in $\mathrm{O}(n+m)$ time.

The same idea can be used for many other problems. A generalization of the above problem is to find for a fixed $k$, a maximum weight subset of vertices that induces a subgraph with maximum clique size at most $k$. The independent set problem is the case $k = 1$. In this case we compute $S[i, A]$ for all subsets $A \subseteq N^+[i]$ with $|A| \leq k$. Here $S[i, A]$ is the maximum weight subset $X$ of descendants of $i$ such that $X \cup A$ induces a subgraph with maximum clique at most $k$. The recurrence is

$$
\begin{aligned}
S[i, A] &= \textstyle\sum_{j \in C[i]} S[j, A \cap N^+[j]] && \text{if } |A| = k \\
&= \max\left(\textstyle\sum_{j \in C[i]} S[j, A \cap N^+[j]],\quad wt(i) + \textstyle\sum_{j \in C[i]} S[j, (A \cup \{i\}) \cap N^+[j]]\right) && \text{if } |A| < k
\end{aligned}
$$

A similar idea works for graphs with bounded treewidth. A graph is said to have treewidth at most $k$ if it is a subgraph of a chordal graph with maximum clique size $k+1$. Given a graph $G$ of treewidth $k$, we can add edges to it and get a chordal graph with maximum clique size $k+1$. Consider a perfect elimination ordering of the resulting chordal graph and its representation as the intersection graph of subtrees of a tree, as defined earlier. The only difference now is that some of the edges may not be original edges in the graph, but if treewidth is small, we can get polynomial-time algorithms. Denote by $N_G[i]$ the set of actual neighbors of $i$ in $G$, while $N^+[i]$ and $N^-[i]$ refer to the neighbors in the chordal graph containing $G$.

Here is an algorithm for finding maximum weight independent set in a graph of treewidth at most $k$. The complexity of the algorithm is $O(2^k n)$. Compute $S[i, A]$, the weight of a maximum weight subset $X$ of descendants of $i$, such that $X \cup A$ is an independent set in $G$. Here $A$ is any subset of vertices of $N^+[i]$. Since the maximum clique size is at most $k+1$, $|N^+[i]| \leq k$, hence there are at most $2^k$ subsets $A$ for each $i$.

The recurrence now is

$$
\begin{aligned}
S[i, A] &= \textstyle\sum_{j \in C[i]} S[j, A \cap N^+[j]] && \text{if } N_G[i] \cap A \neq \emptyset \\
&= \max \left\{ \begin{array}{l} \sum_{j \in C[i]} S[j, A \cap N^+[j]] \\ wt(i) + \sum_{j \in C[i]} S[j, (A \cup \{i\}) \cap N^+[j]] \end{array} \right\} && \text{if } N_G[i] \cap A = \emptyset
\end{aligned}
$$

Many other problems can be solved efficiently for graphs with bounded treewidth. However, there are still many problems that are NP-Hard for chordal graphs. One example is the dominating set problem. Given a chordal graph $G$ and a number $k$, is there a subset of at most $k$ vertices in $G$ such that every vertex not in the subset is adjacent to some vertex in the subset. Dominating set for general graphs is known to be NP-Complete and we can reduce that problem to chordal graphs.

Given an arbitrary graph $H$, construct a chordal graph $G$ as follows. The vertex set of $G$ contains two copies $V_1$ and $V_2$ of the vertex set of $H$. $V_1$ induces a complete graph in $G$ while $V_2$ is an independent set. For each vertex $v$ of $H$, add an edge joining copies of $v$ in $V_1$ and $V_2$. For each edge $uv$ in $H$, add an edge joining the copy of $v$ in $V_1$ to the copy of $u$ in $V_2$, and also an edge joining the copy of $u$ in $V_1$ to the copy of $v$ in $V_2$. Then it is easy to argue that $H$ has a dominating set of size at most $k$ iff $G$ has one.