The SAT-solver used as part of this investigation is called MiniSAT, designed by Niklas Eén and Niklas Sörensson. MiniSAT is a very popular open source SAT solver that has risen to fame in the early 2000s due to its simple, well documented implementation. It contains certain features such as incremental SAT-solving and a well-defined interface for general boolean constraints. MiniSAT is considered a modern SAT-solver since it implements the David-Putnam-Logemann-Loveland (DPLL) procedure, backtracks by conflict analysis and clause learning and propagates boolean constraints using watched literals.[1]

Since SAT solvers require its input to be in conjunctive normal form (CNF), the vertex cover problem needed to be encoded to fit the input criteria for MiniSAT. The vertex cover problem was reduced to CNF form as follows:

Given a pair (G, k), where $G = (V, E)$, let $|V| = n$. Create $n * k$ atomic propositions, denoted $x_{i,j}$, where $i \in [1, n]$ and $j \in [1, k]$. A vertex cover of size k is a list of k vertices. Let $x_{i,j}$ be true if and only if vertex i of V is the jth vertex in that list. The clauses are as follows:

- At least one vertex in the ith vertex in the vertex cover:
  $$\forall i \in [1, k], (x_{1,i} \vee x_{2,i} \vee \cdots \vee x_{n,i})$$
- No one vertex can appear twice in a vertex cover:
  $$\forall m \in [1, n], \forall p, q \in [1, k] \text{ with } p < q, (\neg x_{m,p} \vee \neg x_{m,q})$$
- No more than one vertex appears in the mth position of the vertex cover:
  $$\forall m \in [1, k], \forall p, q \in [1, n] \text{ with } p < q, (\neg x_{p,m} \vee \neg x_{q,m})$$
- Every edge is incident to at least one vertex in the cover:
  $$\forall \langle i, j \rangle \in E, (x_{i,1} \vee x_{i,2} \vee \cdots \vee x_{i,k} \vee x_{j,1} \vee x_{j,2} \vee \cdots \vee x_{j,k})$$

[1] For a more in-depth description of miniSAT, visit http://minisat.se/downloads/MiniSat.pdf