Reduction From Clique To Set Packing

1. Problem Definitions (from Karp)

Karp's 21 problems refer to a set of computational problems that Richard Karp identified as NP-complete in his influential 1972 paper *"Reducibility Among Combinatorial Problems."* These problems are all decision problems that belong to the complexity class NP, and each was shown to be as hard as any other problem in NP through polynomial-time reductions from the Boolean satisfiability problem (SAT), the first known NP-complete problem.
Using the variable names from Karp's paper, the **Clique** (Problem #3) and **Set Packing** (Problem #4) problems are defined as follows:

Clique:
INPUT: graph G, positive integer k
PROPERTY: G has a set of k mutually adjacent nodes.
(Karp, p 94.)

A clique is a subset of vertices in a graph where every vertex is connected to every other vertex in that subset. The goal of the CLIQUE problem is to determine whether a graph contains a clique of size k, where k is provided as part of the input. Additionally, cliques do not require self-loops (edges from a node to itself), which may be present in complete graphs.

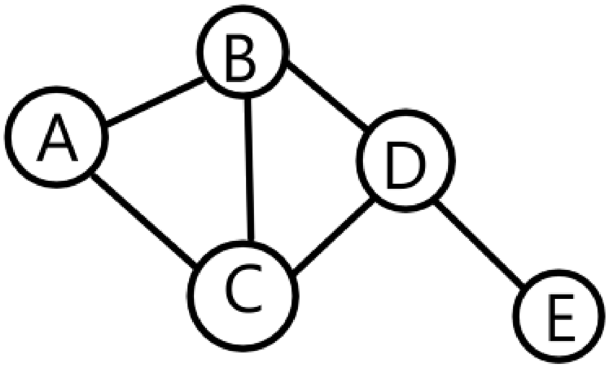
Set Packing:
INPUT: Family of sets $\{S_j\}$, positive integer l
PROPERTY: $\{S_j\}$ contains l mutually disjoint sets.
(Karp, p 94.)

Set Packing is the problem of determining whether a given collection of sets contains a specified number of mutually disjoint sets. The input is a family of sets and a positive integer, and the goal is to find that many sets in the collection such that no two share any common elements.

2. Problem Instances with Explanation

Clique Example:

Vertices: {A, B, C, D, E}
Edges: (A,B), (A,C), (B,C), (C,D), (D,B) (D,E)

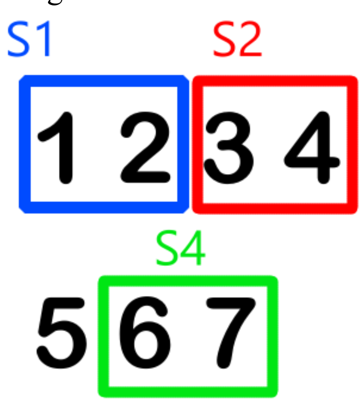


The graph consists of five vertices: A, B, C, D, and E, with edges connecting (A,B), (A,C), (B,C), (C,D), (D,B), and (D,E). Within this graph, the subset {A, B, C} forms a clique because all three vertices are connected to each other by edges. The presence of other vertices and connections does not prevent a clique from existing as long as the subset itself is fully connected.

Set Packing Example:

Set Collection:
S1 = {1, 2}
S2 = {3, 4}
S3 = {2, 5}
S4 = {6, 7}

Target number of sets to select: 3



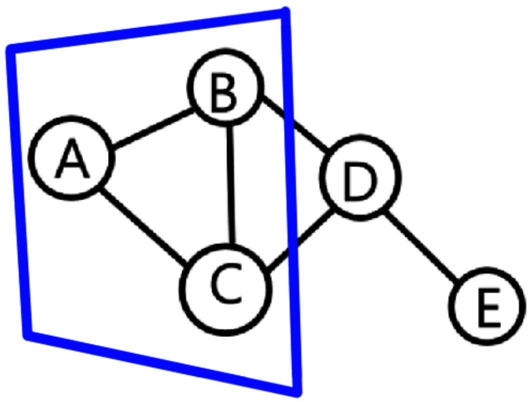
The sets S1, S2, and S4 can be selected because they are pairwise disjoint, no elements are shared between any two of them. This satisfies the condition for set packing, where the goal is to choose a specified number of sets from the collection such that none of the selected sets have any elements in common.

3. Reduction Diagram and Explanation (Clique to Set Packing)

To construct the Clique to Set Packing instance, create a set for each **edge** in the graph. Each set contains the two **vertices** that the edge connects. These sets will form the collection for the Set Packing problem.

Vertices: A, B, C, D, E
Edges: (A,B), (A,C), (B,C), (C,D), (D,B), (D,E)

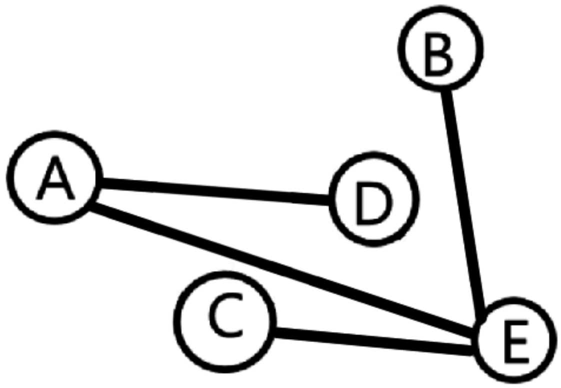
Target: Find a clique of size 3



The resulting sets are:

Edge	Set
(A, B)	{A, B}
(A, C)	{A, C}
(B, C)	{B, C}
(C, D)	{C, D}
(D, E)	{D, E}

However, in this construction, sets that represent edges of a clique, such as S1, S2, and S3 (which represent the edges of clique {A, B, C}) — all share vertices and therefore cannot be selected together in a Set Packing solution.



(Complement graph)

To fix this and make the reduction work, the Set Packing instance must be constructed from the complement graph of the original graph. In the complement graph, edges connect non-adjacent vertices. By constructing sets based on non-connected pairs, a solution to Set Packing (selecting disjoint sets) will correspond to a clique in the original graph.

4. Web Source

Algorist. "Set Packing." *The Algorithm Design Manual*, https://www.algorist.com/problems/Set_Packing.html.

Karp, Richard M. "Reducibility Among Combinatorial Problems." *Complexity of Computer Computations*, edited by R. E. Miller and J. W. Thatcher, Plenum Press, 1972, pp. 85–103.
https://www.cs.purdue.edu/homes/hosking/197/canon/karp.pdf

Roberts, Eric. "Clique Problem." *DNA Computing*, Stanford University, https://cs.stanford.edu/people/eroberts/courses/soco/projects/2003-04/dna-computing/clique.htm.