University of Milan, Computer Science department

Computing the number of Stable Partitions in a disjoint union of cliques

Gabriele Civitarese, Francesco Giorgio

January 14, 2014

1 Introduction

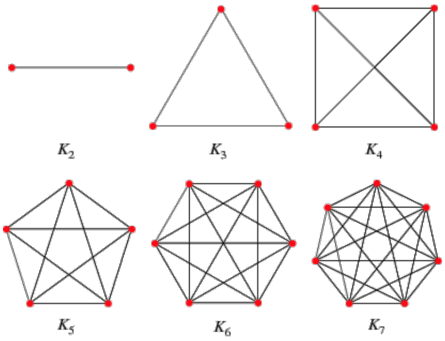
1.1 Abstract

The purpose of this document is to explain how to calculate the number of stable partitions in a disjoint union of cliques. We will provide a general formula and a detailed explanation about this result. We also propose an implementation of this problem.

1.2 Theoretical background

1.2.1 Clique

A *graph* is a pair $<V, E>$, where V is the set of *nodes* and $E \subseteq V\text{x}V$ is the set of *edges* that connect pair of nodes. Let's indicate with $V(G)$ the set of nodes and $E(G)$ the set of edge for a certain graph G. A *clique* K is an undirected graph (all the edges are bidirectional) such that $\forall\ v_1, v_2 \in V(K)$, $\exists\ (v_1, v_2) \in E(K)$. Roughly speaking, every pair of nodes is connected by an edge. We indicate with K_n the clique with n nodes.



A *disjoint union* of n cliques is a graph made by n disconnected cliques.

1.2.2 Stable Partitions

A *partition* π of a set S is a collection of subsets of S called *blocks*: $\pi = \{B_1, B_2, \dots, B_n\}$ such that $\forall i$:

- $B_i \neq \emptyset$
- $\bigcup B_i = S$
- $B_i \cap B_j = \emptyset$ if $i \neq j$

Let's indicate with $B[i]$ the block that contains $i \in S$. A **stable partition** π of a graph G is a partition of $V(G)$ such that if $B[v_1] = B[v_2]$, then $(v_1, v_2) \notin E(G)$. Notice that a clique with m nodes admits just one stable partition in exactly m blocks, for the definition of clique. So, in a disjoint union of cliques a block contains only nodes from different cliques, and the number of blocks is between $max(m_1, m_2, \dots, m_N)$ and $m_1 + m_2 + \dots + m_N$ (where m_i is the number of nodes of the i-th clique, and N is the total number of cliques).

1.2.3 Falling Factorial

It is helpful to explain the definition of *falling factorial*:

$$(n)_k = n(n-1)\dots(n-k+1)$$

2 The problem

From now on, we'll indicate *Stable Partitions* with *SP*. The general problem is to find a formula that calcuates the total number of *SP* of a disjoint union of cliques. Let's say that we have the disjoint union of N cliques:

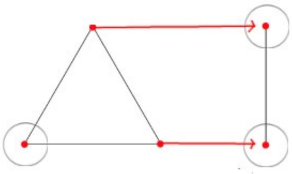
$$K_{m_1}, K_{m_2}, \dots K_{m_N}$$

where m_i is the number of nodes in the i-th clique.

In the next sections we explain the process in different steps: we first analyze the case of the disjoint union of two cliques, then the case of the disjoint union of three cliques, and finally we generalize the problem to an arbitrary number of cliques.

2.1 Disjoint union of two cliques

Let's start by "taking" a certain number of nodes from K_{m_2} and coupling them with as many nodes from K_{m_1}. We call this number i_2, because we will **inject $\mathbf{i_2}$** elements from K_{m_2} to K_{m_1}. By *injecting* we mean that we're going to include i_2 nodes from K_{m_2} and K_{m_1} in the same blocks, so that we have i_2 blocks with 2 nodes. The other blocks will be *singletons*. For example, the following picture shows the disjoint union of K_3 and K_2 where $i_2 = 2$ nodes of K_3 are injected into the only *SP* of K_2.



Let's notice how it's impossible to inject K_{m_1} nodes in K_{m_1} due to the fact that it's a clique and how is useless to inject K_{m_1} nodes in K_{m_2} since injection is commutative. Now it's easy to see that the total number of blocks is $m_1 + m_2 - i_2$, with $0 \leq i_2 \leq m_2$. It's obvious that, regardless the value of i_2, the total number of blocks can't overtake the limits stated before.

Now we can answer the question: *what is the number of* SP *in $m_1 + m_2 - i_2$ blocks of a 2-cliques disjoint union?*
The answer is given by the following formula:

$$SP(K_{m_1} \cup K_{m_2}; m_1 + m_2 - i_2) = \binom{m_2}{i_2} (m_1)_{i_2}$$

In fact:

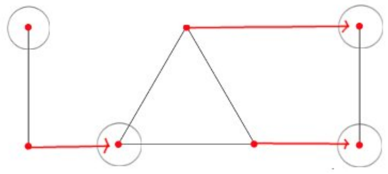
- $\binom{m_2}{i_2}$ represents how many ways we have to choose i_2 nodes from m_2.
- $(m_1)_{i_2}$ represents how many ways we have to inject the i_2 nodes in (m_1).

Now we want to know the **total** number of *SP*, regardless the number of blocks. To do this we iteratively sum the precedent formula over all possible i_2 values and consequently over all possible numbers of blocks:

$$SP(K_{m_1} \cup K_{m_2}) = \sum_{i_2=0}^{m_2} \binom{m_2}{i_2} (m_1)_{i_2}$$

2.2 Disjoint union of three cliques

What happens if we consider a third clique? The main reasoning is essentially the same: supposed that the new clique K_{m_3} has m_3 nodes, we select a certain number among them, let's say $0 \leq i_3 \leq m_3$ and try to inject them. The difference is that now we don't have single nodes to inject into, but the SP of $K_{m_1} \cup K_{m_2}$. In the following picture we recall the previous example. In this case we have the disjoint union of K_2, K_3 and K_2. We choose $i_3 = 1$ node from the first K_2 and we inject it in one of the *SP* of $K_3 \bigcup K_2$ (the one that we discussed in the precedent section).



In fact, by injecting i_3 nodes in this way, we'll find the number of SP of $K_{m_1} \cup K_{m_2} \cup K_{m_3}$ in $m_1 + m_2 - i_2 + m3 - i_3$ blocks, because also the number of SP of $K_{m_1} \cup K_{m_2}$ is calculated given a certain i_2 representing how many K_{m_2} nodes have been injected into K_{m_1}:

$$SP(K_{m_1} \cup K_{m_2} \cup K_{m_3}; m_1 + m_2 - i_2 + m_3 - i_3) =$$
$$\binom{m3}{i_3} SP(K_{m_1} \cup K_{m_2}; m_1 + m_2 - i_2)(m_1 + m_2 - i_2)_{i_3}$$

In fact:

- $\binom{m3}{i_3}$ represents how many ways we have to choose i_3 nodes from m_3.

- $SP(K_{m_1} \cup K_{m_2}; m_1 + m_2 - i_2)(m_1 + m_2 - i_2)_{i_3}$ represents how many ways we have to inject the i_3 nodes in the $SP(K_{m_1} \cup K_{m_2}; m_1 + m_2 - i_2)$.

Like before, to know the **total** number of SP, regardless the number of blocks, we must sum this formula over all possible i_2 and i_3:

$$SP(K_{m_1} \cup K_{m_2} \cup K_{m_3}) = \sum_{i_3=0}^{m_3} \sum_{i_2=0}^{m_2} \binom{m_3}{i_3} \binom{m_2}{i_2} (m_1)_{i_2} (m_1 + m_2 - i_2)_{i_3}$$

Notice that we have replaced the $SP(K_{m_1} \cup K_{m_2}; m_1 + m_2 - i_2)$ member.

2.3 GENERALIZING

Needless to say that the same reasoning with 4 cliques will lead us to the following formula:

$$SP(K_{m_1} \cup K_{m_2} \cup K_{m_3} \cup K_{m_4}) =$$

$$\sum_{i_4=0}^{m_4} \sum_{i_3=0}^{m_3} \sum_{i_2=0}^{m_2} \binom{m_4}{i_4} \binom{m_3}{i_3} \binom{m_2}{i_2} (m_1)_{i_2} (m_1 + m_2 - i_2)_{i_3} (m_1 + m_2 - i_2 + m_3 - i_3)_{i_4}$$

It's easy to see a pattern in the general solution and we can now generalize to n cliques. With an arbitrary number of cliques, the approach is iterative: we first inject in all possible ways the nodes of K_{m_2} in the m_1 blocks of the only *SP* of K_{m_1}. Then we inject in all possible ways the nodes of K_{m_3} in all the blocks of all the *SP* of the disjoint union of K_{m_1} and K_{m_2} , and so on: we iteratively keep on doing this for all the cliques of the disjoint union. This iterative process lead us to the following general formula:

$$SP(\bigcup_{i=0}^{n} K_{m_i}) = \sum_{i_2=0}^{m_2} \sum_{i_3=0}^{m_3} ... \sum_{i_n=0}^{m_n} \prod_{j=2}^{n} \binom{m_j}{i_j} (\sum_{k=1}^{j-1} m_k - i_k)_{i_j}$$

3 Implementation

In this section we explain how to implement the formula to calculate the total number of *SP* in a disjoint union of cliques. We propose two algorithms: the first one calculates exactly the number of *SP*, the second one calculates the number of *SP* for each number of blocks.

3.1 Algorithm to calculate SP

The first algorithm that we propose returns the total number of *SP* of an arbitrary disjoint union of cliques. These are the input of the algorithm:

- N is the number of cliques in the disjoint union.
- A vector $m = [m_0, m_1 \dots m_N]$ where m_i is the number of nodes of the i-th clique.
- A vector $c = [c_0, c_1 \dots c_N]$ where c_i is the counter of the i-th sum (initialized to 0).

So, we define this recursive procedure *totSP* that takes one parameter n, that indicates which sum we are considering. The first call will be *totSP(1)* because in the formula we never use the first sum (c_0 will always be 0).

Algorithm 1 $totSP(n)$

```
if $N \leq 1$ then
  $sum = 1$
else
  if $n == N$ then
    $sum \leftarrow sum + \prod_{i=2}^{N} \binom{m_i}{k_i} (\sum_{j=1}^{i-1} m_j - k_j)_{k_i}$
  else
    for $c_n = 0$ to $m_n$ do
      $sum \leftarrow sum + totSP(n+1)$
    end for
  end if
end if
return $sum$
```

3.1.1 Calculating the number of *SP* in each number of blocks

Now we propose a second algorithm very similar to the first. This time the procedure returns a map between the number of blocks and the relative number of *SP* in that number of blocks. With $map[k] = v$ we mean that key k is mapped to value v.

Algorithm 2 $SPBlocks(n)$

```
if $N == 0$ then
  $map[0] = 1$
else
  if $N == 1$ then
    $map[m_0] = 1$
  else
    if $n == N$ then
      $map[\sum_{i=0}^{N-1} m_i - c_i] = \prod_{i=2}^{N} \binom{m_i}{c_i} (\sum_{j=1}^{i-1} m_j - c_j)_{c_i}$
    else
      for $c_n = 0$ to $m_n$ do
        $SPBlocks(n+1)$
      end for
    end if
  end if
end if
```