Correctness and Working of Conflict Resolution

Our Fisrt Step is adding the minimum edge of each node to the MST.
This step is trivially correct.

Second step we get the conflicts (*where my minimum has me as its minimum*)
These are the number of edges we will need to add.
These are like the groups in Boruvka's algorithm.
They are unconnected by logic;
If a node(1) connects to another node(2),
and node(2) connects to yet another node(3).
Then the edge between (2) and (3) is smaller than
the edge between (1) and (2).
This inequality continues for the whole chain.
Stopping at the point of conflict where,
(1) and (2) connect to eachother.
So, no edge can form between such groups, as each node can connect to only one other node.

Third step we give chains and groups to the node.
Group number is the conflict location.
Chains are there to speed up processing.

Fourth step we repeat till no conflicts.
Each group(a) will try to reach another group(b) by its smallest
external edge.
This can now be seen as node(a) connecting to node(b).

So like in mergesort, we stop when at one element.

This algothrim's correctness is based on not adding any edge outside the mst.
I we consider only one group, it will be Prim's algorithm.

Example execution.

Consider the graph,   from geeks_for_geeks

0 – 1 (2) 3 (6)
1 – 2 (3) 3 (8) 4 (5)
2 – 4 (5)
3 – 4 (9)

First step ->   MST has 0-1 , 2-1 , 3-0 , 4-1.
conflict   1-0 ,
Giving answer in first step.

Now, consider new edge   3-4 (1)

First step ->   MST has 0-1 , 2-1 , 3-4
conflict   1-0 ,   4-3

Second step is getiing two groups as 1 and 4.
In group 1 minimum to another group is   1 – 4(5)
In group 4 minimum to another group is   4 – 1(5)

we get new edge for MST   1 - 4
and one conflict   1 – 4

giving final MST   0-1, 1-2, 3-4, 1-4.