

FPGA_REPORT

方法主要是參考助教提供的reference那篇論文，但是論文並沒有針對FPGA的capacity做限制，而且對於Algo1，會用到floyd warshall來找各個node之間的距離，一開始本來有寫，但看到測資很大，十分鐘感覺是做不出來，所以後來就只針對fixed node的鄰居更新candidates，也就是fixed node鄰居的candidates會跟fixed node所放置的FPGA以及放置FPGA的鄰居做交集，來盡量減少candidates。

而針對Algo2，我主要利用了兩個priority queue，分別是Q及R，Q是來選擇哪個node先放，我用了兩個參數做排列，首先是candidates數量少得優先，若一樣則再用nets少的為優先(nets舉例來說，如果給2 3 5、2 8，那nets[2]就會存3、5、8)，先把nets數量少的先放，比較有可能產生0個external_degree，接著R則是看要放在哪個FPGA中，這邊也用了兩個參數來做排列，首先是看放入哪個FPGA產生的external_degree最少(external_degree計算，若node 1放入FPGA 0，則看nets[1]中已經固定的node，若固定的node與node 1的FPGA不同則external_degree[1][0]++)，若相同，則再看哪個fpga_capacity_predict較少(fpga_capacity_predict是我用來存，當前有多少node可以放在這個FPGA，如node 1的candidate為2、3，node 2的candidate為3，則fpga_capacity_predict[3]=2)，越多代表越多node能放在這個FPGA，那我就盡量不要放，越低則較少node要放在這個FPGA，那我就令越低的FPGA優先權較高，希望利用這個參數來盡量減少無法放入的問題。

但目前這個方法只能用在node數量為20000以下，20000以上我是用，先固定fixed node，再照順序的方式將node放進FPGA(從FPGA 0開始放，滿了就換FPGA 1，以此類推)，但盡量讓nets中的node放在同一個FPGA(舉例來說，若node 1放在FPGA 0，那我就讓nets[1]中的node在不超過當前FPGA capacity的前提下放在FPGA 0)，來盡量減少external_degree，這方式比完全照順序放可以減少快一倍。