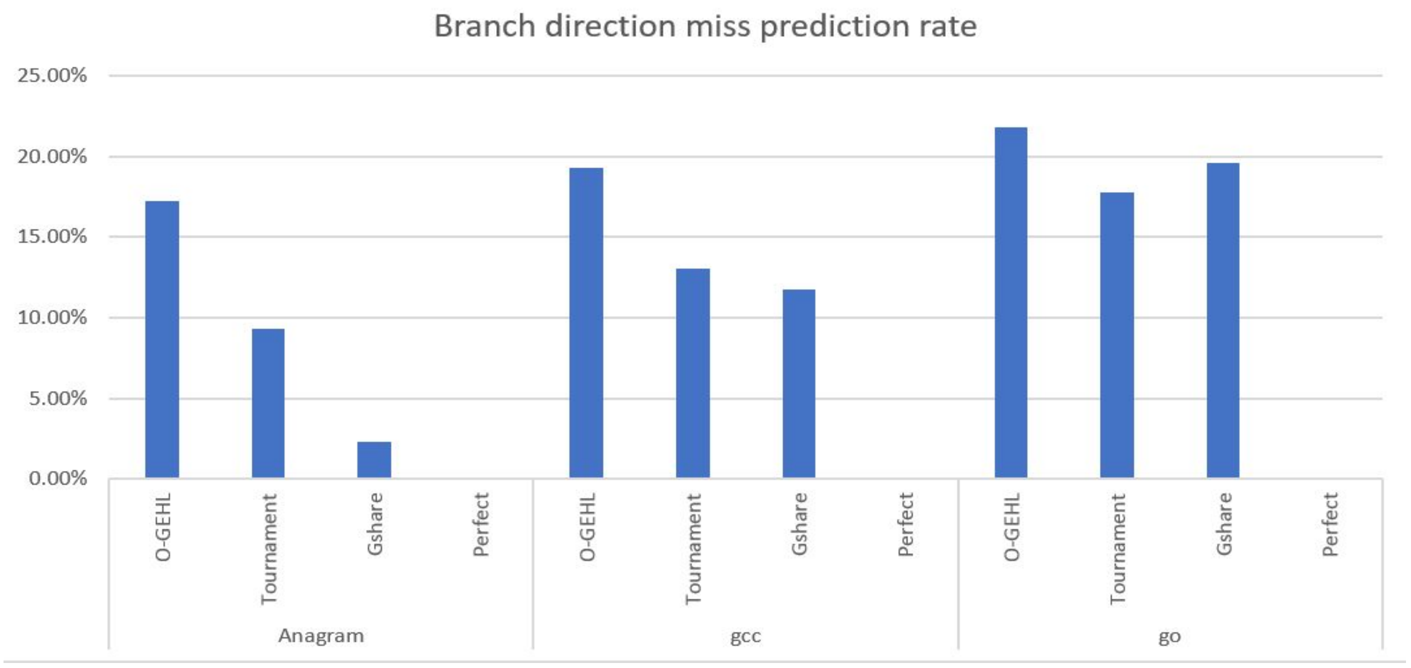
Report

Pragna Munagala
ASU ID: 1211111639

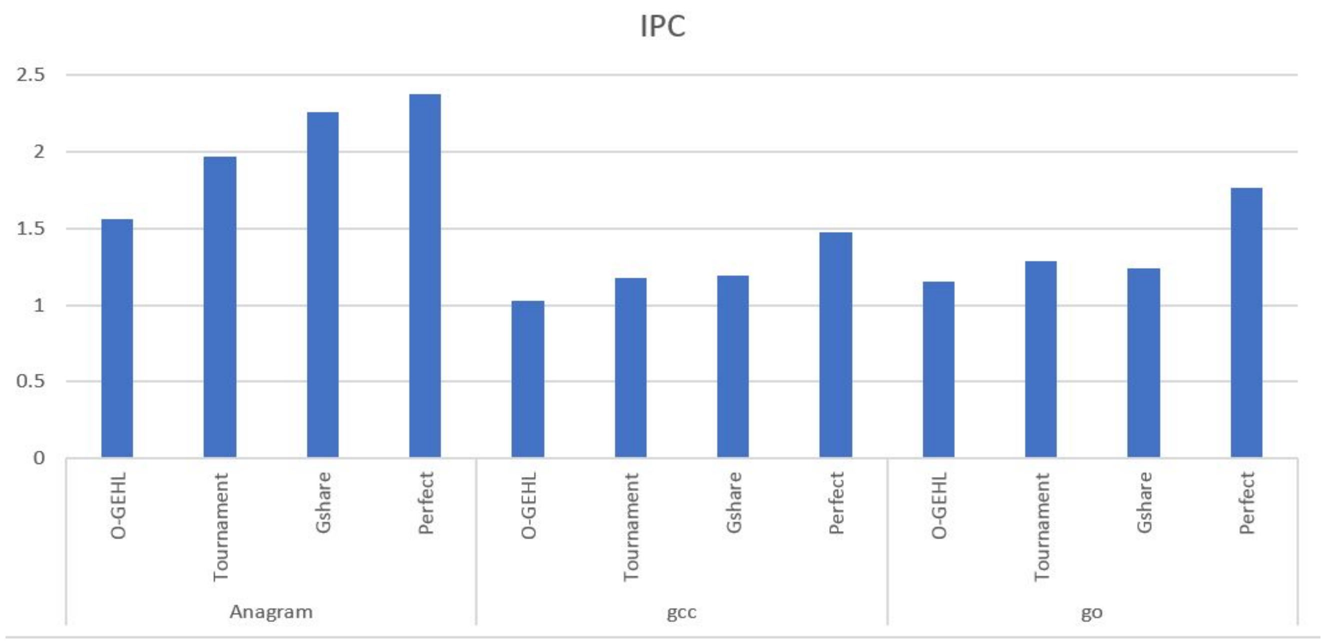
The branch direction miss rate of Tournament, O-GEHL, GShare and Perfect predictors for various benchmarks is as shown below:

Anagram				**gcc**				**go**			
O-GEHL	**Tournament**	**Gshare**	**Perfect**	**O-GEHL**	**Tournament**	**Gshare**	**Perfect**	**O-GEHL**	**Tournament**	**Gshare**	**Perfect**
17.22%	9.34%	2.33%	0.00%	19.27%	13.03%	11.79%	0.00%	21.78%	17.78%	19.59%	0.00%



The IPC of Tournament, O-GEHL, GShare and Perfect predictors for various benchmarks is as shown below:

Anagram				**gcc**				**go**			
O-GEHL	**Tournament**	**Gshare**	**Perfect**	**O-GEHL**	**Tournament**	**Gshare**	**Perfect**	**O-GEHL**	**Tournament**	**Gshare**	**Perfect**
1.5602	1.9674	2.2582	2.376	1.0267	1.1775	1.1911	1.4753	1.1565	1.2849	1.2364	1.7619



Analysis:

1) **Anagram** - Total number of instructions that are committed are approximately 25 million and total execution time - 11 seconds and the number of instruction per branch are approximately 6.
   gcc - Total number of instructions that are committed are approximately 337 million and total execution time - 276 seconds and the number of instruction per branch are approximately 5.
   go - Total number of instructions that are committed are approximately 545 million and total execution time - 1770 seconds and the number of instruction per branch are approximately 7.

2) The miss prediction rate of O-GEHL should be less than tournament predictor but in my analysis, I got the other way round. The reasons for this might be:
   a) We are not considering long branch history lengths because of which the miss prediction rate might be higher (although this depends on the benchmark).
   b) We are also not considering the adaptive threshold fitting and adaptive branch history lengths as mentioned in the paper reference given.

3) IPC is inversely proportional to branch miss prediction rate. Because of miss predictions, the pipeline has to be flushed and loaded with required instructions which will consume the time equal to length of the pipeline. The concept of branch prediction is to save the cycles because we cannot afford stalling the pipeline until the execution (**EX**) stage of conditional branches. But if we make a miss prediction about the next instructions that are to be fetched, then we have to pay more and more penalty in order to get the actual instructions which indeed makes the IPC go high. So the IPC calculation also depends on total number of instructions, time of access to the hardware on which branch predictors (The Impact of Delay on the Design of Branch Predictors by Daniel A. Jimenez) are designed and length of pipelines. If the pipeline is very deep, then penalty is very high making the IPC go low.