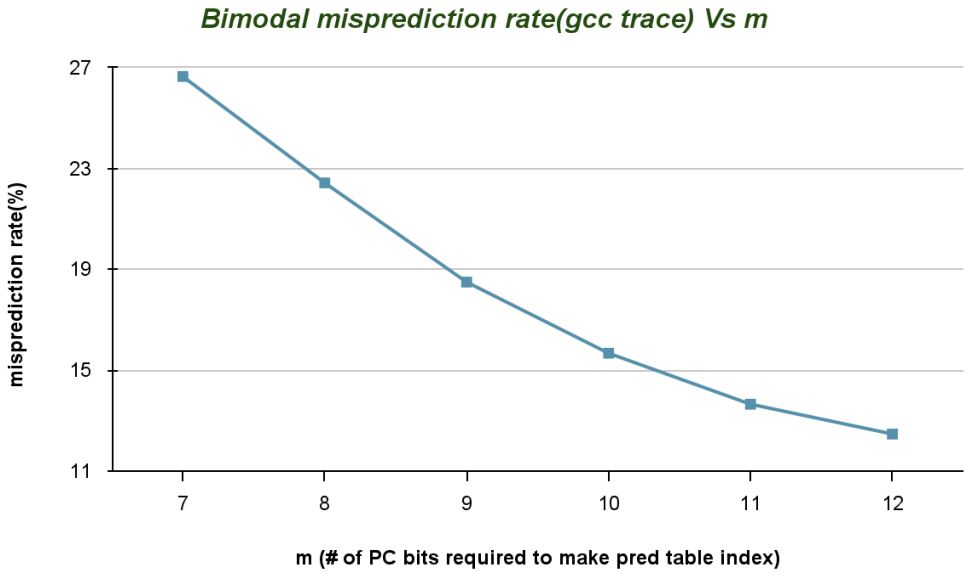
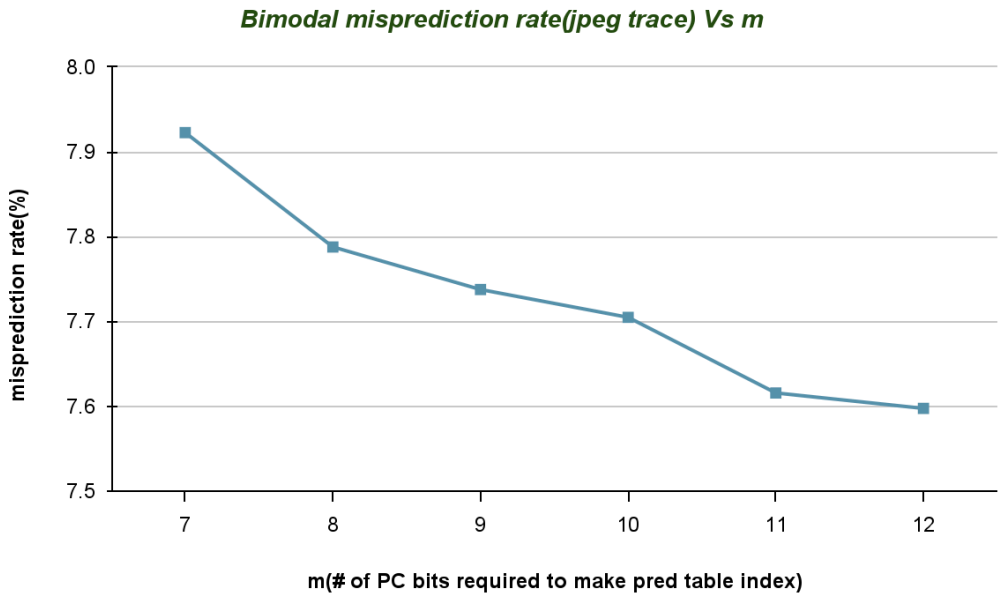
BRANCH PREDICTOR SIMULATOR REPORT

Bimodal Predictor

Plot 1(a) : Misprediction rate Vs m





Similar trends:

For both the traces, misprediction rate decreases as 'm' increases.

When branches are indexed into larger prediction table, branch collisions of different PCs decrease and can store more different branch patterns. Due to this, there can be a decrease in misprediction rate for most of the general branches.

However after a certain point of increase of m, there is diminishing returns of decrease in misprediction rate as table may contain more rarely used branch entries.

Differences:

Both the gcc and jpeg files have the same number of branches in its trace. However, the misprediction rate is higher in gcc compared to jpeg trace. When m is increasing, percentage of decrease of misprediction rate is higher in gcc compared to jpeg showing that it is benefitting more due to larger prediction table.

In jpeg trace as said earlier, misprediction rate is almost similar when m is increasing implying that jpeg trace has more predictable branch behaviour of its branches.

When m=7, mispred rate = 7.79%

When m=12, mispred rate = 7.597%

Smaller table is able to capture most of its predictions in jpeg trace unlike gcc trace.

Question 1(b):

m	Misprediction rate % (gcc)
7	26.6485
8	22.4286
9	18.4933
10	15.6714
11	13.6502
12	12.4672

m	Misprediction rate % (jpeg trace)
7	7.9227
8	7.78785
9	7.73765
10	7.70485
11	7.6157
12	7.5975

Prediction table stores 2-bit counters for each entry.

So size of table = # of entries * 2 = 2^m * 2 bits = 2^{m-2} bytes $\leq$ 16KB (=2^{14}B)

$\Rightarrow$ m - 2 $\leq$ 14 $\Rightarrow$ m $\leq$ 16 So all m values from 7 to 12 are possible with 16KB storage space.

But we need to reduce the misprediction rate with reasonable cost. For m increase by 1, storage space increases by 50% (doubles).

For gcc trace, there is a significant decrease in mispred rate till m = 10. After m = 10, rate of decrease in misprediction rate is less. (Storage space is proportional to cost)
For m = 10, storage space = 2^8 B = **256 B**
For m = 11, storage space = 2^9 B = 512 B
Bimodal predictor design for gcc trace could be m = 10 with misprediction rate = 15.67%

For jpeg trace, misprediction rates are almost the same for all m. However opting for a somewhat bigger table is not costly as when m = 9 is considered. (**128B** table)
Even though m = 10 to m = 11, there is a decrease of 0.09 it is not really required to consider 512B(m=11) for a 0.12 decrease in misprediction rate from m = 9.
Bimodal predictor design for jpeg trace could be m = 9 with misprediction rate = 7.74%

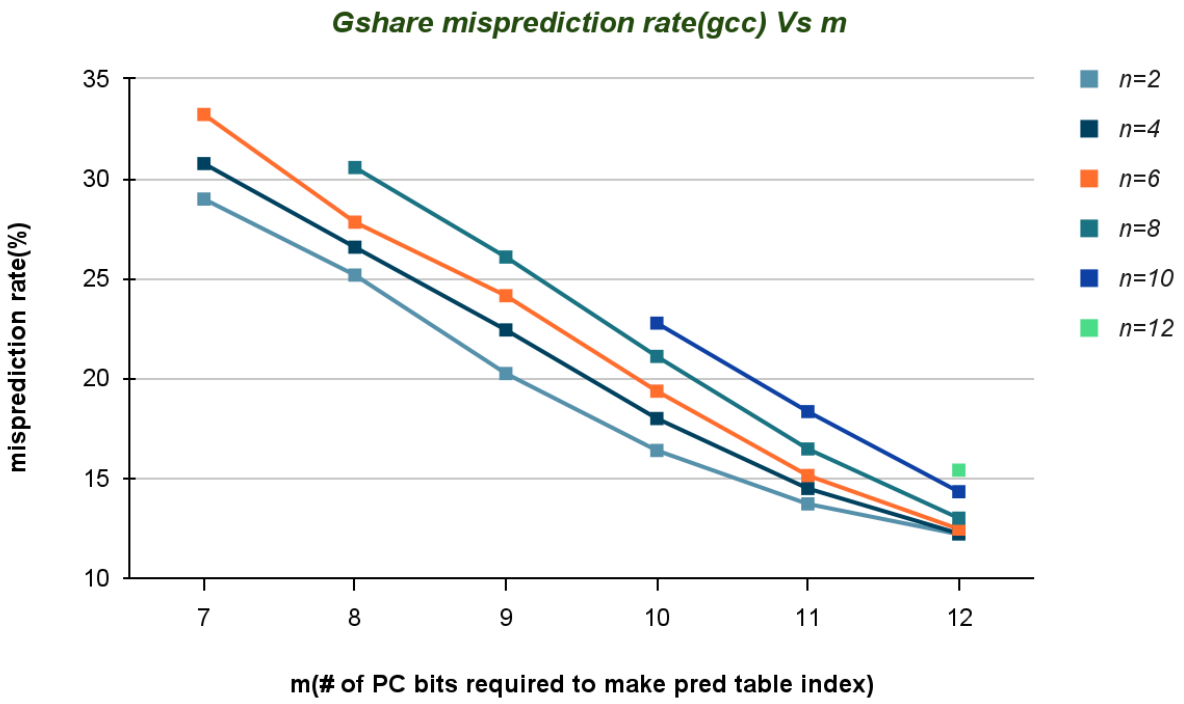
Gshare Predictor

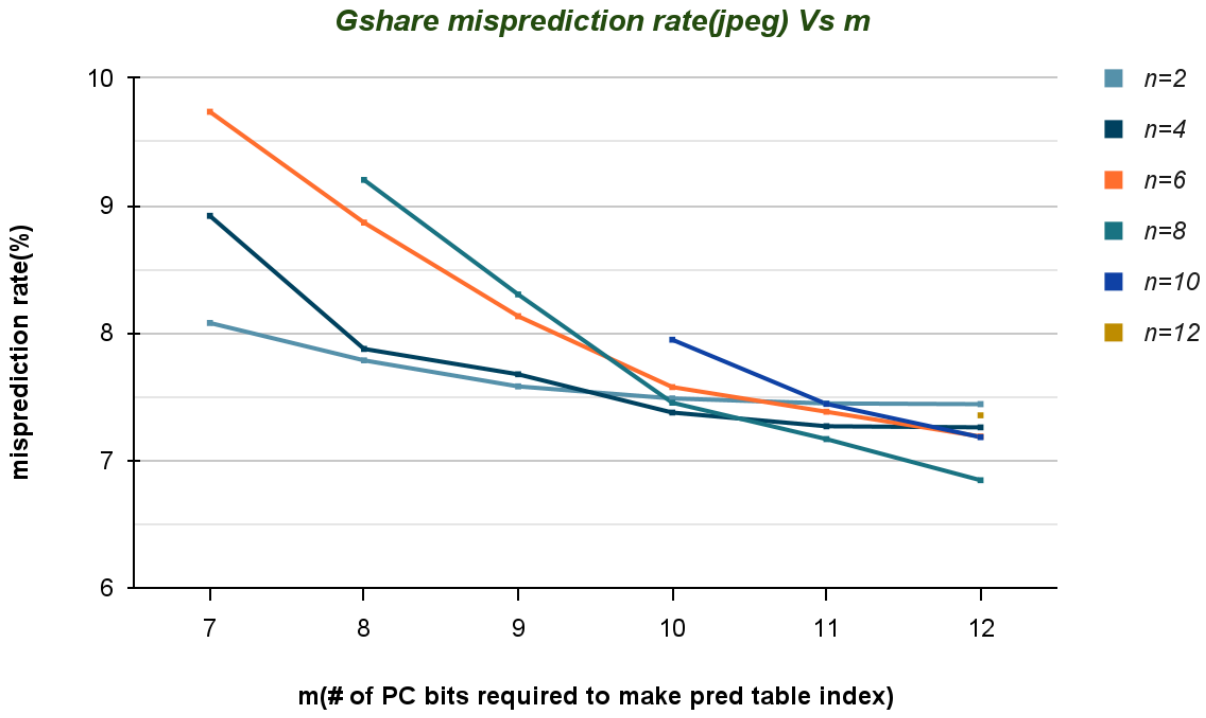
Misprediction rate(in %) for gcc trace:

m	n=2	n=4	n=6	n=8	n=10	n=12
7	28.9838	30.7588	33.2203			
8	25.1783	26.5729	27.8238	30.5616		
9	20.2474	22.4297	24.1433	26.0793		
10	16.3925	17.9925	19.3631	21.101	22.7695	
11	13.7148	14.4884	15.138	16.4682	18.3384	
12	12.1959	12.2337	12.4562	13.0021	14.3262	15.4049

Misprediction rate(in %) for jpeg trace:

m	n=2	n=4	n=6	n=8	n=10	n=12
7	8.07905	8.9198	9.73675			
8	7.7854	7.87525	8.86685	9.2021		
9	7.58095	7.67595	8.13185	8.3029		
10	7.4864	7.3757	7.57565	7.45305	7.94755	
11	7.44735	7.2681	7.38195	7.1674	7.4433	
12	7.44175	7.25965	7.1876	6.84465	7.18235	7.3536





Larger prediction table is benefitting both traces for a fixed number of global branch history. (i.e. for fixed n, as m increases misprediction rate decreases)

However for gcc trace, when m is held constant, as n increases misprediction rate is increasing. Indexing using more global branch history is not benefitting.
And sometimes, smaller prediction tables with few global branch history, larger tables with large global branch history yield the same prediction rate. Smaller tables can be considered to reduce cost for storage.
While for jpeg trace, when m is held constant, as n increases misprediction does not follow an increasing trend for m ≥ 10.

Question 2(b):

For gcc trace, diminishing returns of misprediction rate occurs when m is increased from 11 to 12. (m = 11 ⇒ storage space = 512B, m = 12 ⇒ storage space = 1024B = 1KB)
And as n value is increasing, gcc trace sees the trend of increasing misprediction rates but it's better to consider n larger than minimal value like 2 since it's a gshare branch predictor.
Gshare predictor design for gcc trace could be m = 11 (with storage space 512B) and n value can be 4 / 6 as misprediction rates are not differing from m = 11, n = 2 with the same cost.
For jpeg trace, gshare predictor design could be m = 11 (with storage space 512B) and n value can be either 4 or 8.