Experiments:

To find the optimal configuration for each trace in the pruned search space, the parameters of number of Load/Store Units, MUL units, reservation stations per FU, ROB entries and fetch width, retire width and data cache size were varied and the corresponding output statistics were written to a CSV file for further analysis. Only the configurations in the reduced search space were considered for the trace files.

Since the optimal configuration with the least resources utilized had to have an IPC within 90% of the maximum IPC obtained as the threshold, those configurations with an IPC lesser than the threshold were filtered out. Before, we can choose the configuration we also need to define a priority and weight for each of the parameters.

Using a direct mapped cache of a very large capacity would utilize more power or add latency to the processor. Hence, it was decided that limiting the cache capacity would be given main precedence. Since, using an extra execution unist has a heavy cost in terms of hardware and adding extra read ports, the next priority was to reduce the number of execution units used. Reservations stations add extra bits to store the necessary information and we would need to perform a fully associative lookup each time an instruction is retired to update the appropriate reservation station. Hence using a scheduling queue of a bigger size would add increase the power consumption significantly or limit the execution speed of the processor. So, minimizing this is given precedence next. However, we also need to keep enough instructions ready to fire, so that should not cause a bottleneck on the instructions completed per cycle. The retire width also needs to be kept to minimum since this also adds a hardware complexity of more result buses that the scheduling queue needs to check to update its ready status. We keep the fetch width minimum within the pruned search space, but if increasing the fetch width alone can increase the IPC significantly, it would be given consideration. Since ROB requires lesser number of bits for each entry ROB size is given lesser precedence. Among the pruned search space all configurations require three ALU units, hence increasing or decreasing number of ALU units was not considered.

Parameter	Priority
Cache size	1
L/S units	2
MUL units	3
Fetch width	4
Scheduling queue entries	5
Retire width	6
ROB entries	7

Approximate formula: 2.5*S*(A+M+L) + 0.7*R + 16*F + 12*W + 4*2^(C-5) + 20*A + 30*M + 60*L

Although, this weight priority is set initially to get an IPC within 90% of the maximum IPC obtained, other statistics were also analyzed to gauge the effect of each parameter's effect on getting a better performance. If the priority had to be broken so that we can utilize lesser resources on multiple fronts and get better auxiliary statistics like lesser No fire cycles and maximum Dispatch Queue Usage, and close to maximum usage of the scheduling queue and the ROB most of the time, then such configurations were selected as the optimum configuration by providing appropriate reasoning.

Bwaves trace:

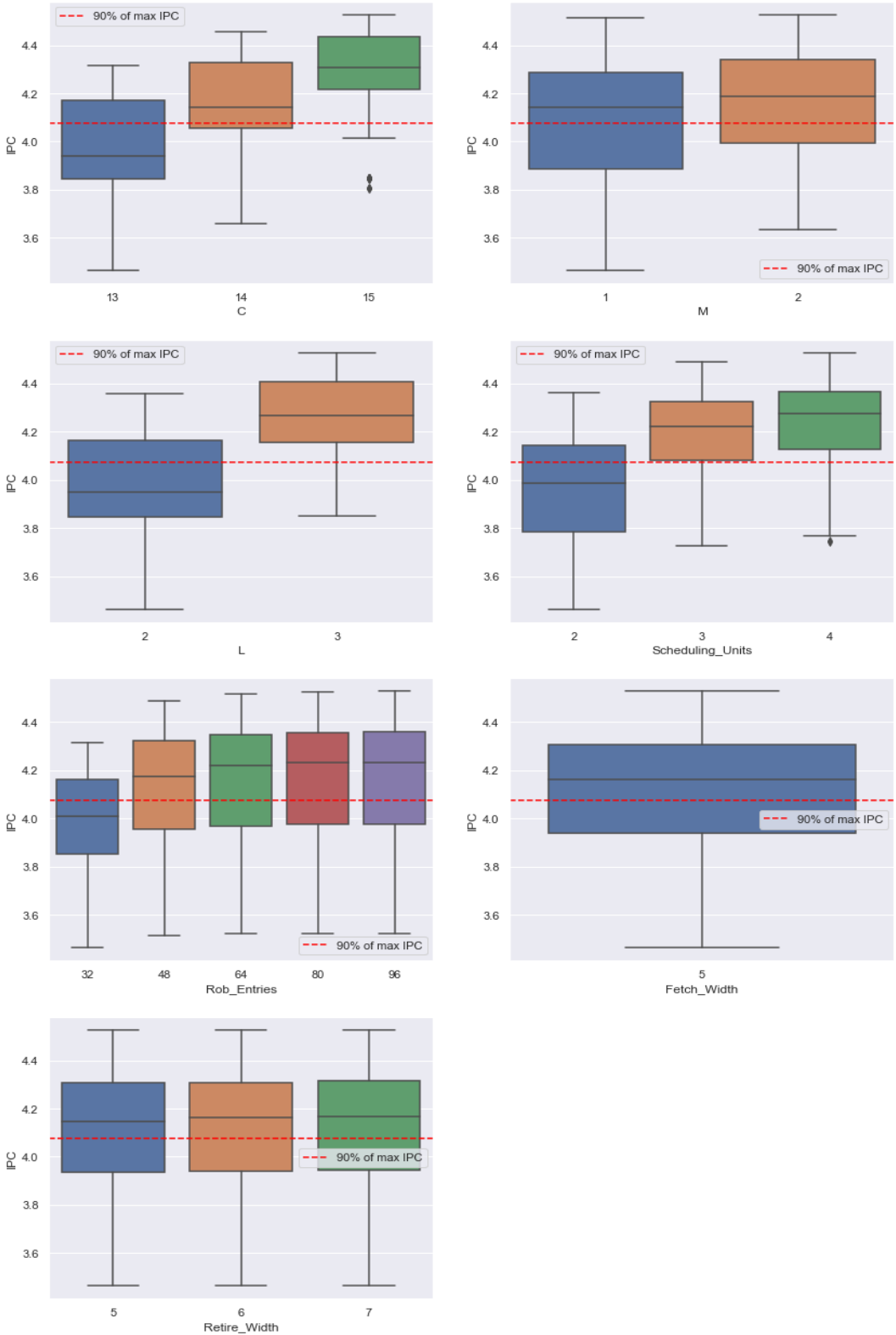

Figure 1.1: Box plots of IPC versus each parameter for bwaves trace

From the above plots, we can infer the effect of each parameter on getting a high IPC. The red dotted line denotes the 90% threshold of IPC that the configuration needs to achieve. We can observe that that increasing the number of load units from a value of 2 to 3 has a substantial increase in the number of configurations that perform better than the threshold. However, we also notice that increasing the scheduling units per FU also provides us a considerable performance boost. The objective would be to increase the performance by increasing the minimum number of parameters and thus utilize minimum amount of resources in terms of hardware cost and power consumption.

C	14
A	3
M	1
L	2
S	3
R	64
F	5
W	5
IPC	4.08695
Dispatch Queue Max Usage	26456
Scheduling Queue Average Usage	17.93
No fire cycles	14152
ROB Average Usage	15.852435

Figure 1.2: Optimal configuration for bwaves trace

Another configuration that provided a slightly better performance in terms of IPC, Dispatch Queue Max Usage and no-fire cycles was:

C	14
A	3
M	1
L	2
S	4
R	48
F	5
W	5
IPC	4.120763
Dispatch Queue Max Usage	25767
Scheduling Queue Average Usage	22.81563
No fire cycles	12402
ROB Average Usage	17.05343

However, this configuration uses more Scheduling Units and lesser ROB entries. This could also be a candidate for the optimum configuration but had a slightly higher cost score on the weights assigned.

Gcc trace:

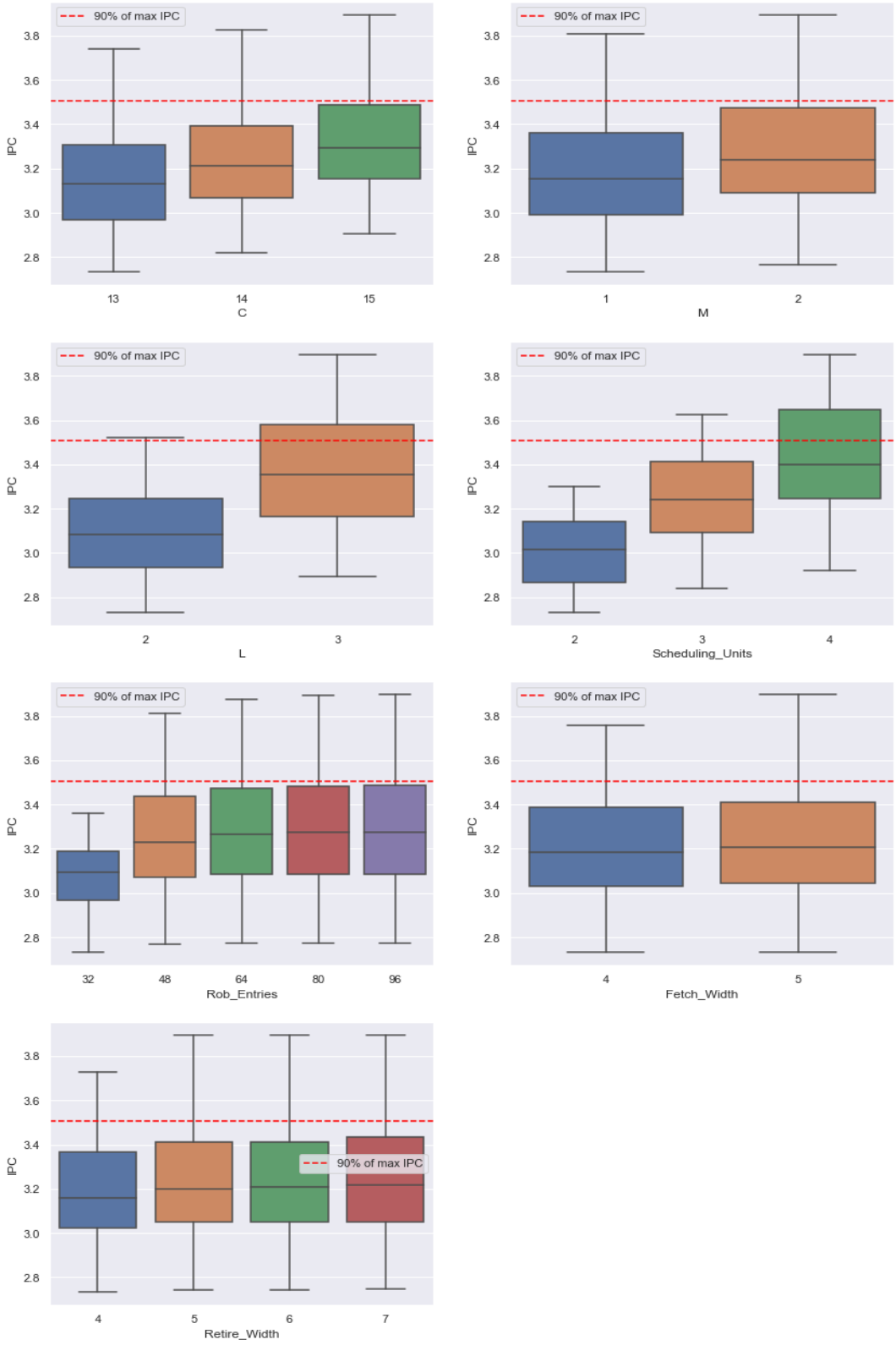

Figure 2.1: Box plots of IPC versus each parameter for gcc trace

C	13
A	3
M	1
L	3
S	4
R	64
F	4
W	4
IPC	3.525273
Dispatch Queue Max Usage	27327
Scheduling Queue Average Usage	22.86624
No fire cycles	9500
ROB Average Usage	29.7125

Figure 2.2: Optimal configurations for gcc trace

The 2 configurations had the same score on the weights assigned and the first configuration had a better IPC, so the configuration was chosen as the optimum. From figure 2.1, we can also observe that increasing the number of Load/Store Units and the number of scheduling units per FU has the most effect on IPC. Also, increasing the ROB size over 64 has diminished improvements.

Mcf trace:

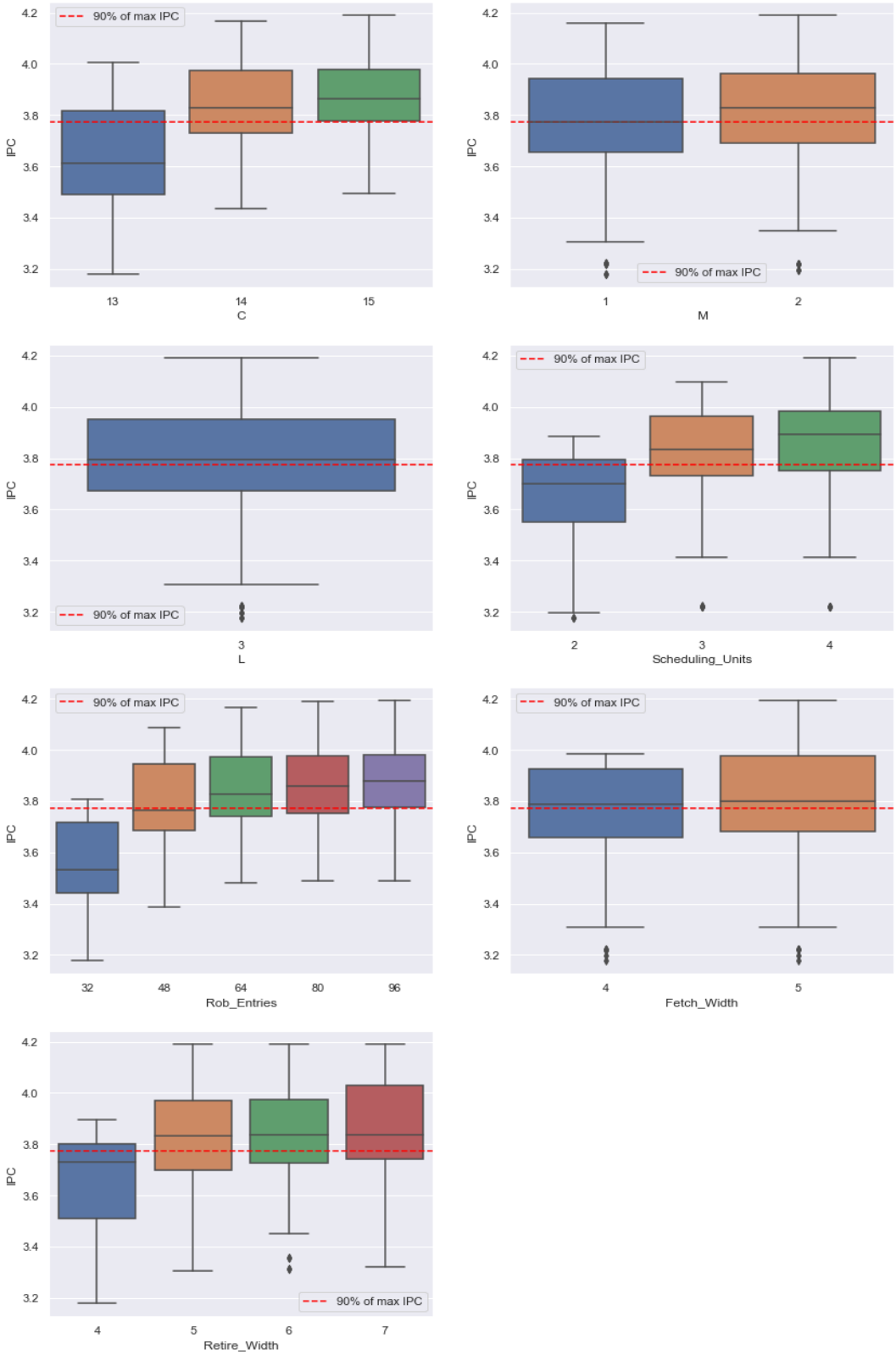

Figure 3.1: Box plots of IPC versus each parameter for mcf trace

C	13
A	3
M	1
L	3
S	3
R	80
F	4
W	5
IPC	3.816463
Dispatch Queue Max Usage	9098
Scheduling Queue Average Usage	20.69357
No fire cycles	8124
ROB Average Usage	26.02636

Figure 3.2: Optimal configurations for mcf trace

The configuration listed above had the same score with the cost weighting system pre-determined. We can observe from figure 3.2 that increasing the scheduling units above 3 and retire width greater than 5 has diminishing returns.

Perlbench trace:

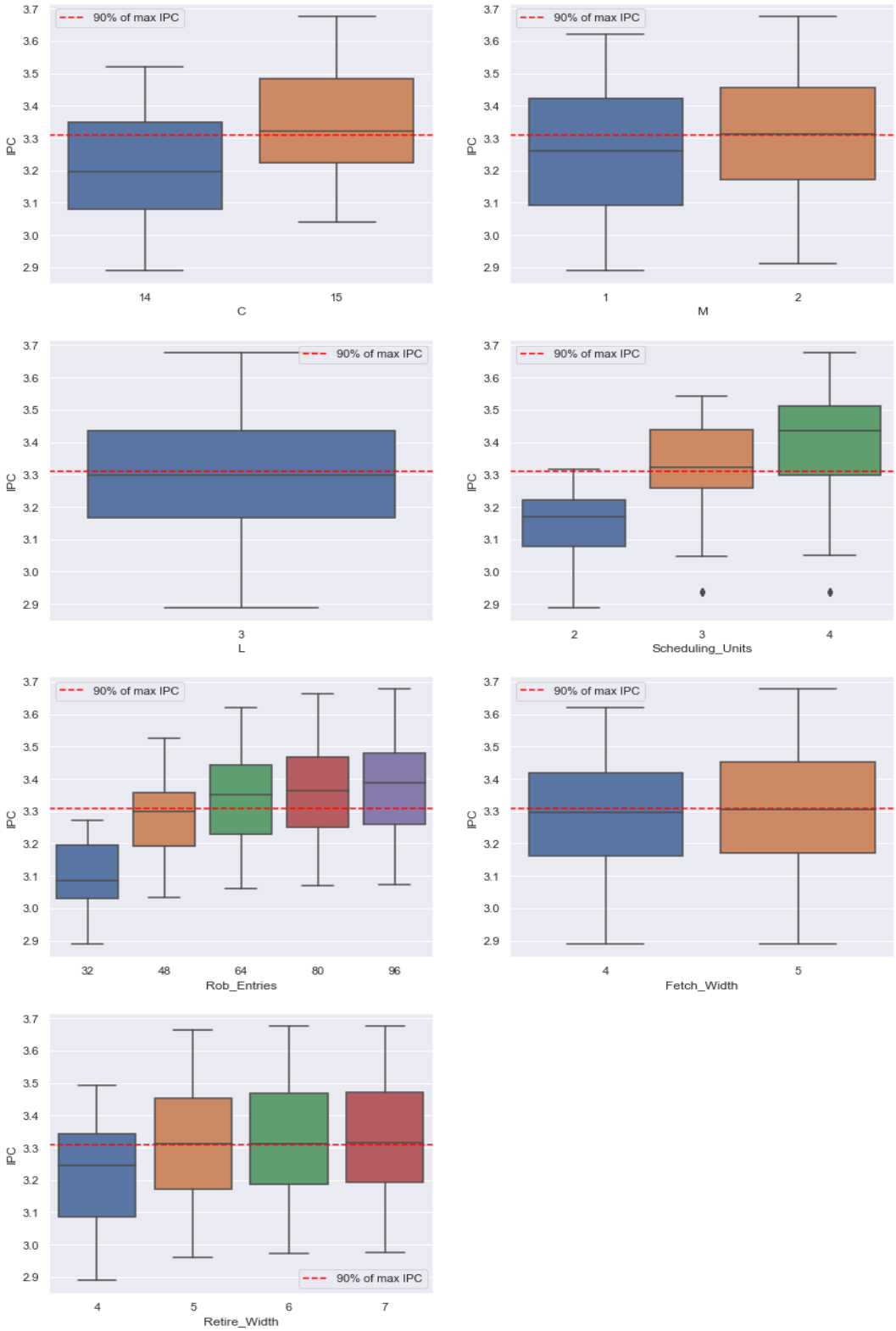

Figure 4.1: Box plots of IPC versus each parameter for perlbench trace

C	14	14
A	3	3
M	1	1
L	3	3
S	4	4
R	64	80
F	4	4
W	5	4
IPC	3.3945	3.336434
Dispatch Queue Max Usage	12731	12591
Scheduling Queue Average Usage	25.45979	23.4811
No fire cycles	12000	10870
ROB Average Usage	26.879	42.86478

Figure 4.2: Optimal configurations for perlbench trace

The above 2 configurations got a similar score under our weighting method. The first one has a better IPC, and higher average scheduling queue usage, however the retire width is higher which adds a hardware complexity. The second configuration has a lower IPC but has lesser no fire cycles and maximum dispatch queue usage size.