

# Atividade 2

## Profiling

### Integrantes :

Felipe Seiji Momma Valente	12543700
Fernando Gonçalves Campos	12542352
Thiago Shimada	12691032

## Descrição da atividade

Foram desenvolvidos 8 experimentos de multiplicação de matrizes para se testar o impacto de diferentes técnicas de otimização de código, sendo eles, 4 códigos com alocação de memória dinâmica e 4 com memória estática. E em cada uma delas foram utilizadas 4 diferentes otimização de código, sendo elas : sem otimização nenhuma, loop unrolling, loop interchange e block tiling.

Após essa etapa, cada código foi executado vinte vezes para calcular as médias de tempo de execução, cache load, cache miss, branch instructions e branch misses, utilizando a ferramenta perf. Em seguida, um script em R foi aplicado para gerar os gráficos e analisar as influências desses resultados.

Foram realizadas duas análises:

- Comparação do tempo de execução entre todas as configurações
- Comparação de métricas de cache e branch para as seguintes configurações:
  - Alocação de memória estática ou dinâmica
  - Técnica de otimização block tiling ou sem otimização

## Resultados de execução

### Memória estática

Sem otimizações :

- Tempo de execução : 1,2812s (+- 0,00276105s)
- L1 dcache load : 8,747,199,563 (+- 2,683,491)
- L1 dcache load misses : 56,149,846 (+- 873,595)
- Branch instructions : 533,331,640 (+- 140.243)
- Branch misses : 683,512 (+- 539)

Loop interchange :

- Tempo de execução : 1,2136s (+- 0,00276105s)
- L1 dcache load : 8,747,201,206 (+- 3.066.847)
- L1 dcache load misses : 4,560,026 (+- 62.552)
- Branch instructions : 533,194,389 (+- 163.575)
- Branch misses : 683,096 (+- 628)

Loop unrolling :

- Tempo de execução : 1,3160s (+- 0,0054783s)
- L1 dcache load : 8,233,069,437 (+- 2.886.588)
- L1 dcache load misses : 54,958,048 (+- 2.131.610)
- Branch instructions : 277,196,898 (+- 194.375)
- Branch misses : 683,823 (+- 419)

Block Tilling :

- Tempo de execução : 0,8024s (+- 0,002016s)
- L1 dcache load : 7,989,460,285 (+- 4.551.913)
- L1 dcache load misses : 1,231,015 (+- 20.339)
- Branch instructions : 581,386,996 (+- 25.479)
- Branch misses : 5,932,148 (+- 33.277)

### **Memória dinâmica**

Sem otimizações :

- Tempo de execução : 1,3420s (+- 0,0020949s)
- L1 dcache load : 10,799,777,418 (+- 3.313.187)
- L1 dcache load misses : 323,641,510 (+- 4.382.842)
- Branch instructions : 553,130,568 (+- 266.657)
- Branch misses : 683,033 (+- 808)

Loop interchange :

- Tempo de execução : 1,0543s (+- 0,0023228s)
- L1 dcache load : 11,308,364,860 (+- 2.973.611)
- L1 dcache load misses : 4,110,699 (+- 47.201)
- Branch instructions : 533,844,121 (+- 140.378)
- Branch misses : 685,318 (+- 480)

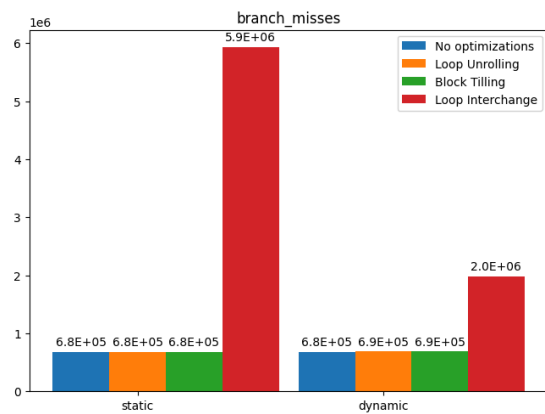
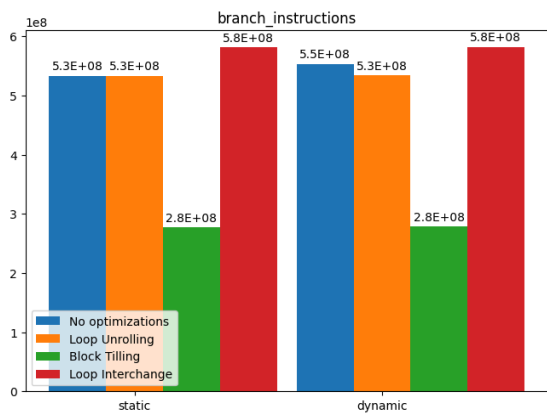
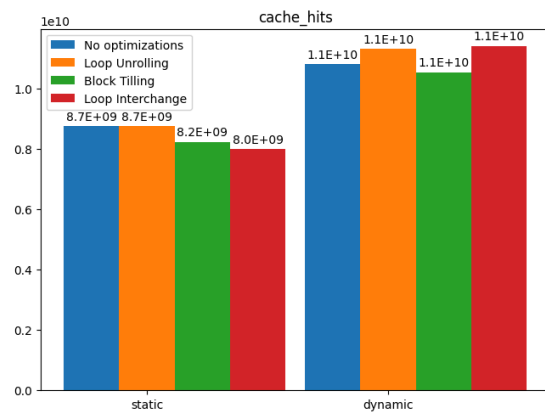
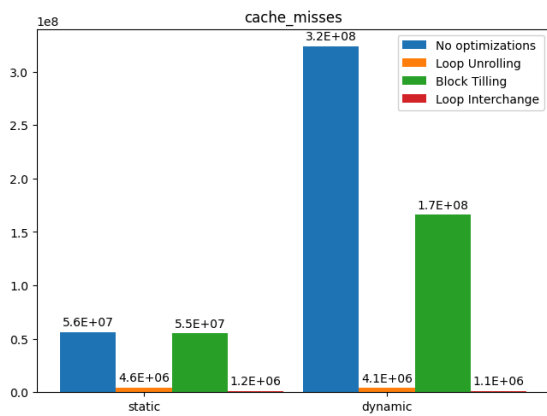
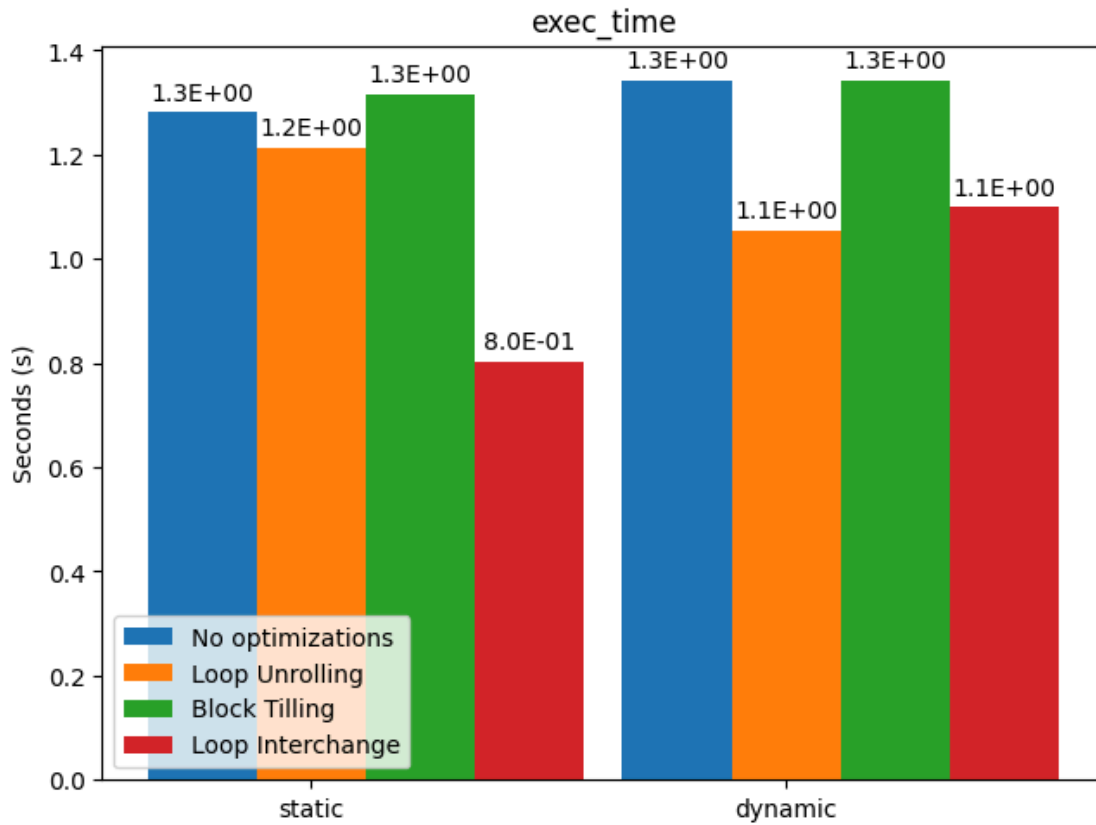
Loop unrolling :

- Tempo de execução : 1,3420s (+- 0,0050838s)
- L1 dcache load : 10,541,084,505 (+- 2.309.875)
- L1 dcache load misses : 165,996,945 (+- 181.875)
- Branch instructions : 278,541,227 (+- 134.281)
- Branch misses : 685,510 (+- 330)

Block Tilling :

- Tempo de execução : 1,0989s (+- 0,0026734s)
- L1 dcache load : 11,402,836,699 (+- 2.498.711)
- L1 dcache load misses : 1,082,665 (+- 2.277)
- Branch instructions : 581,611,942 (+- 101.959)

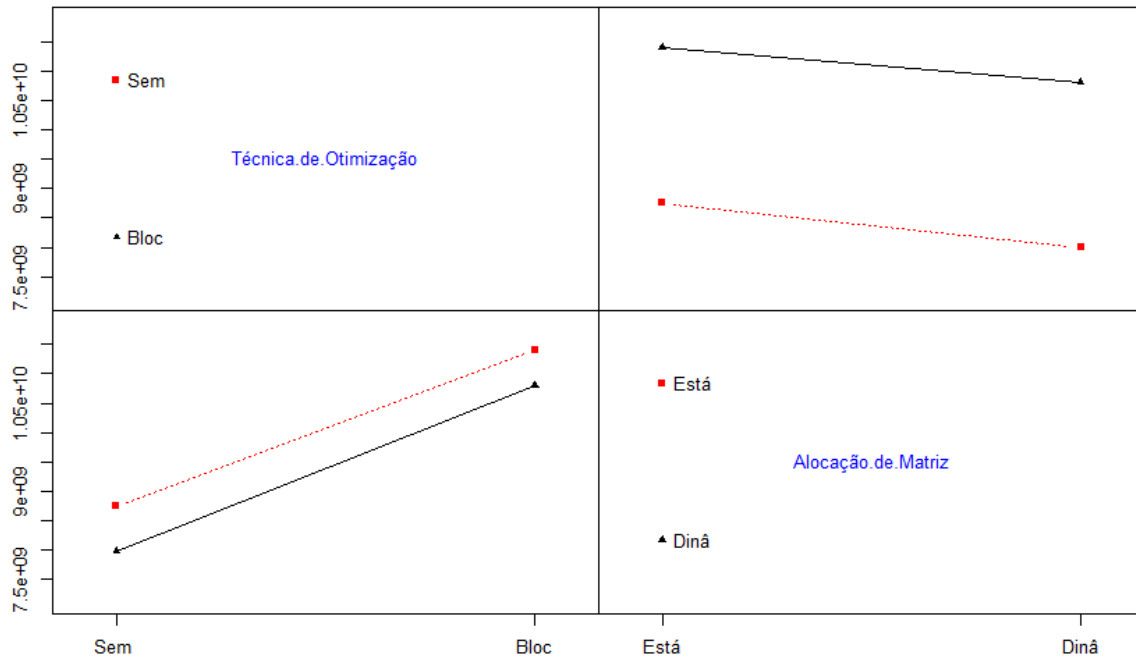
- Branch misses : 1,980,289 (+- 17.791)



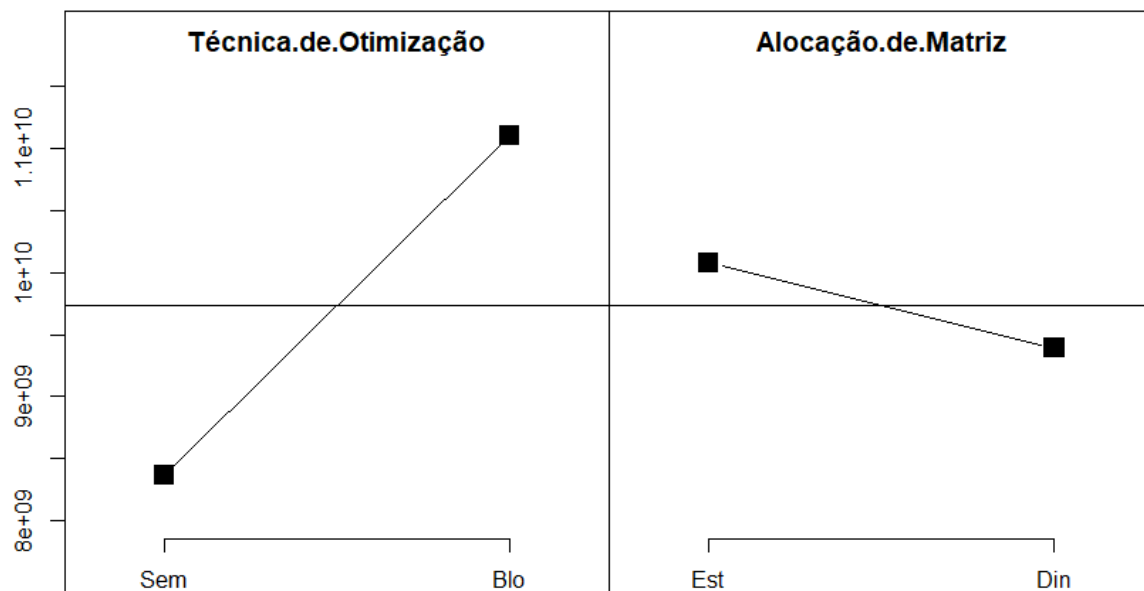
# Gráficos e Influências

Gráficos para métrica **L1 dcache load**

**Interaction plot matrix for results**



**Main effects plot for results**



Influência do fator técnica : 0.9409273

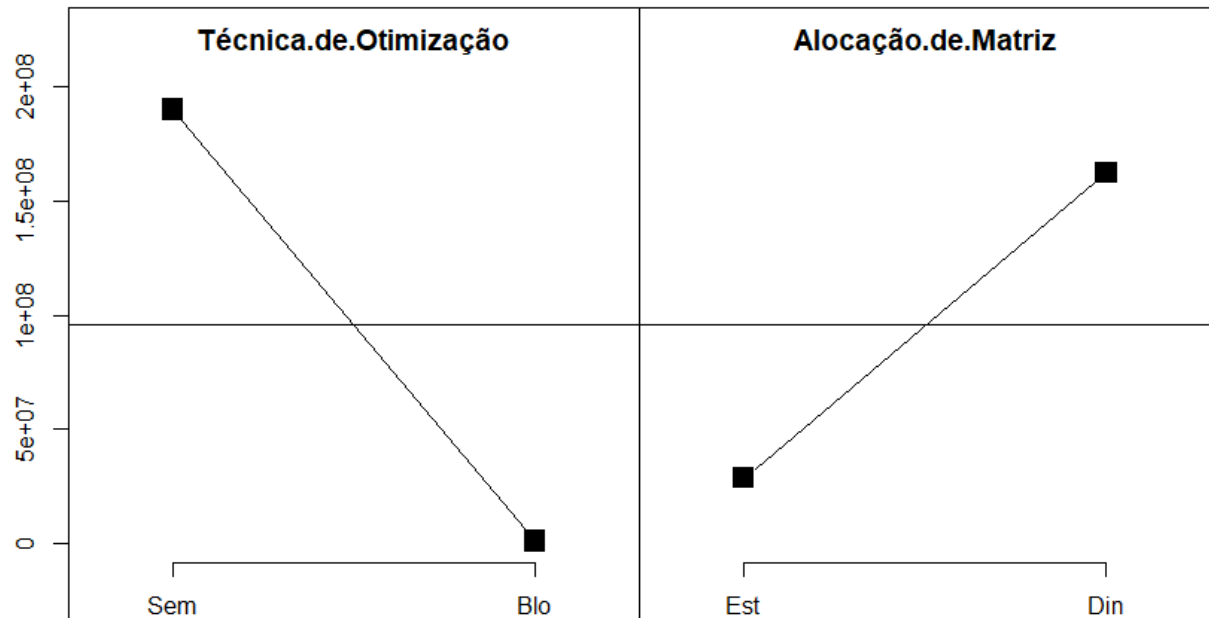
Influência do fator alocação : 0.05831923

Influência dos fatores técnica + alocação : 0.0007535158

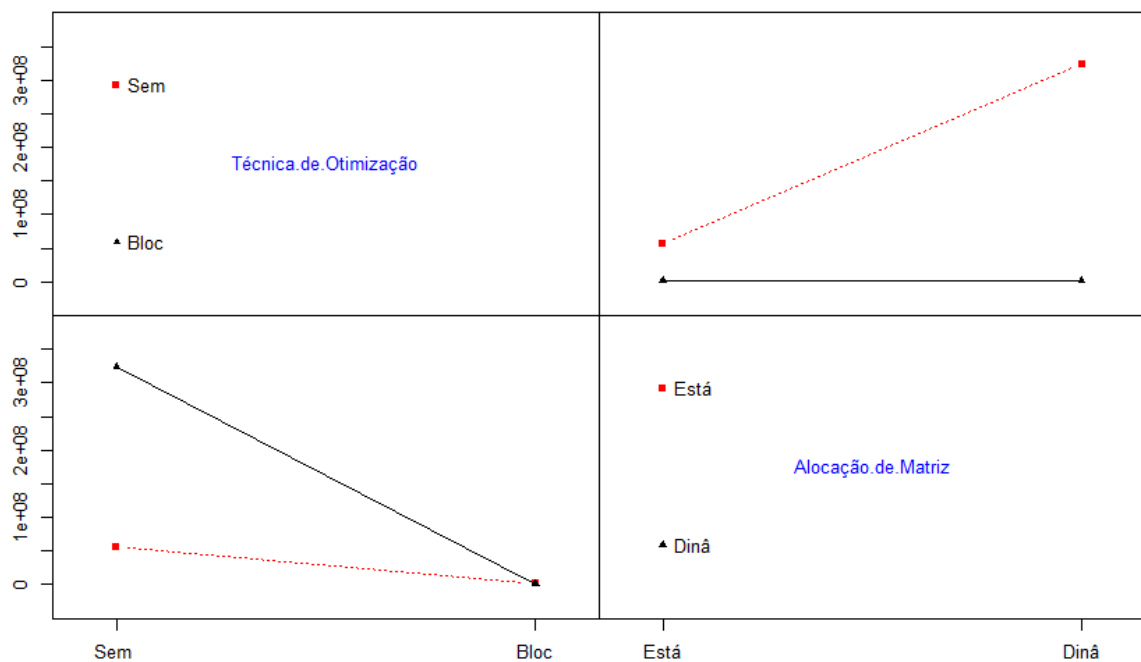
**Conclusão:** A escolha da técnica possui uma influência consideravelmente maior no número de acertos da cache do que o método de alocação das matrizes.

Gráficos para métrica **L1 dcache load misses**:

### Main effects plot for results



### Interaction plot matrix for results



Influência do fator técnica : 0.4989246

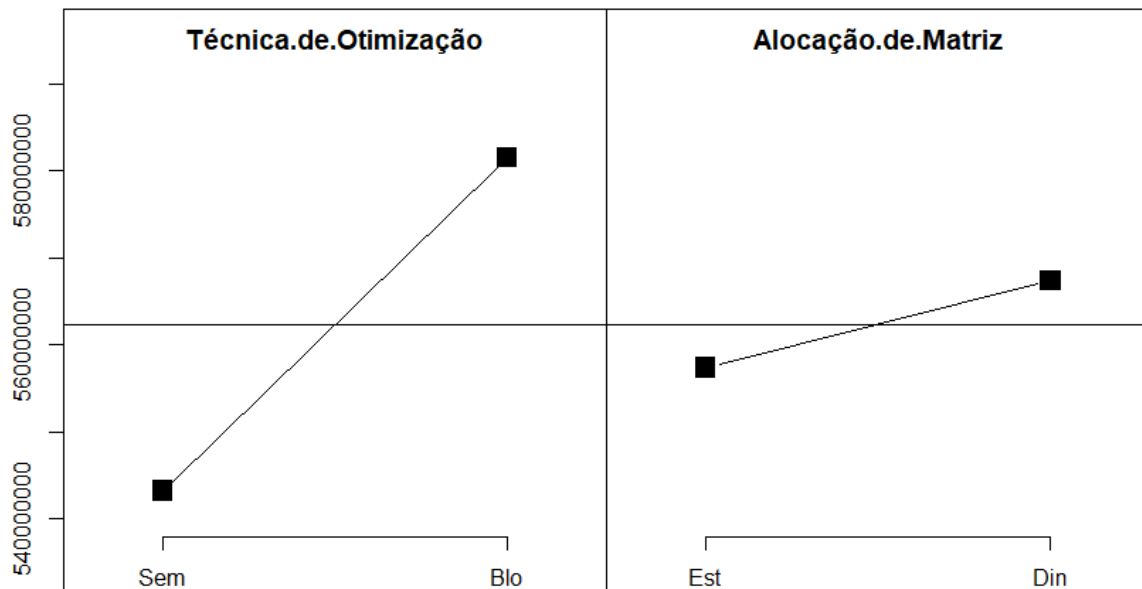
Influência do fator alocação : 0.2502598

Influência dos fatores técnica + alocação : 0.2508156

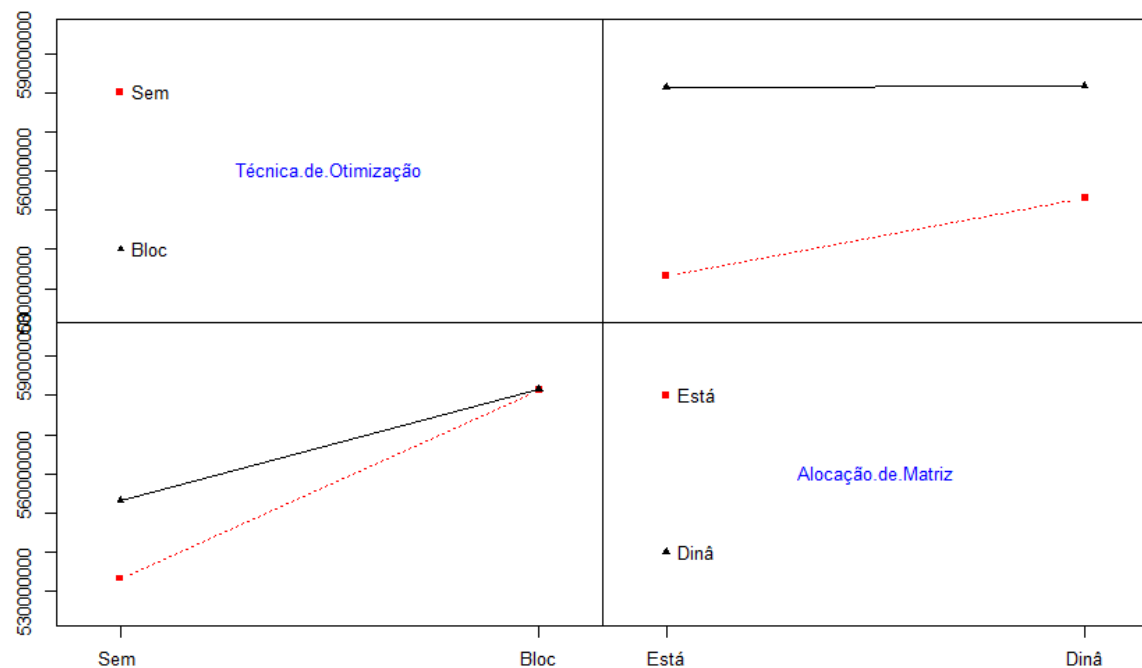
**Conclusão:** A escolha da técnica possui uma influência um pouco maior no número de erros de cache do que o método de alocação de matrizes.

Gráficos para métrica **branch instructions**:

### Main effects plot for results



### Interaction plot matrix for results



Influência do fator técnica : 0.8819482

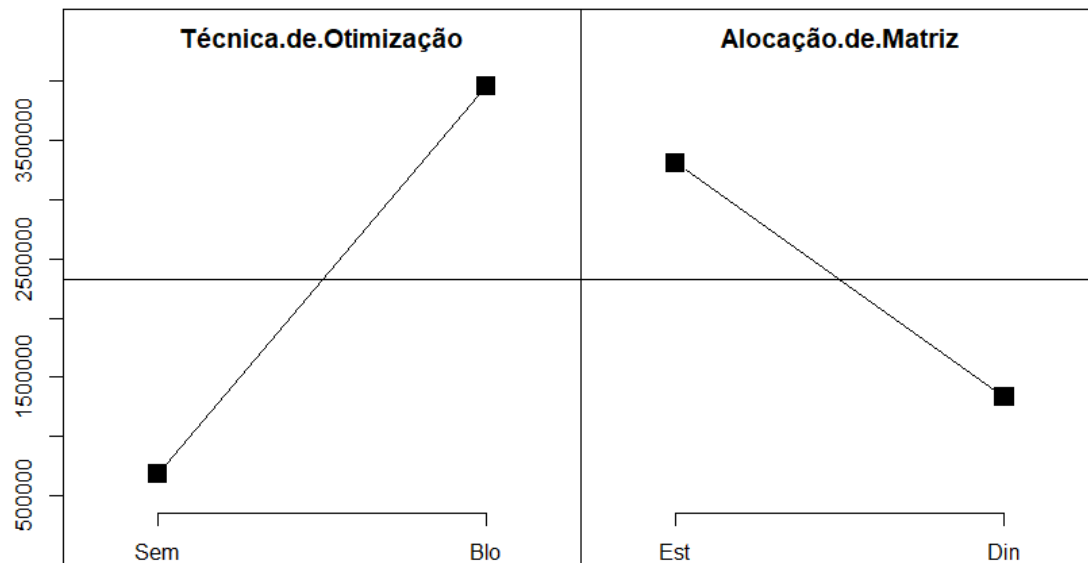
Influência do fator alocação : 0.06036698

Influência dos fatores técnica + alocação : 0.05768483

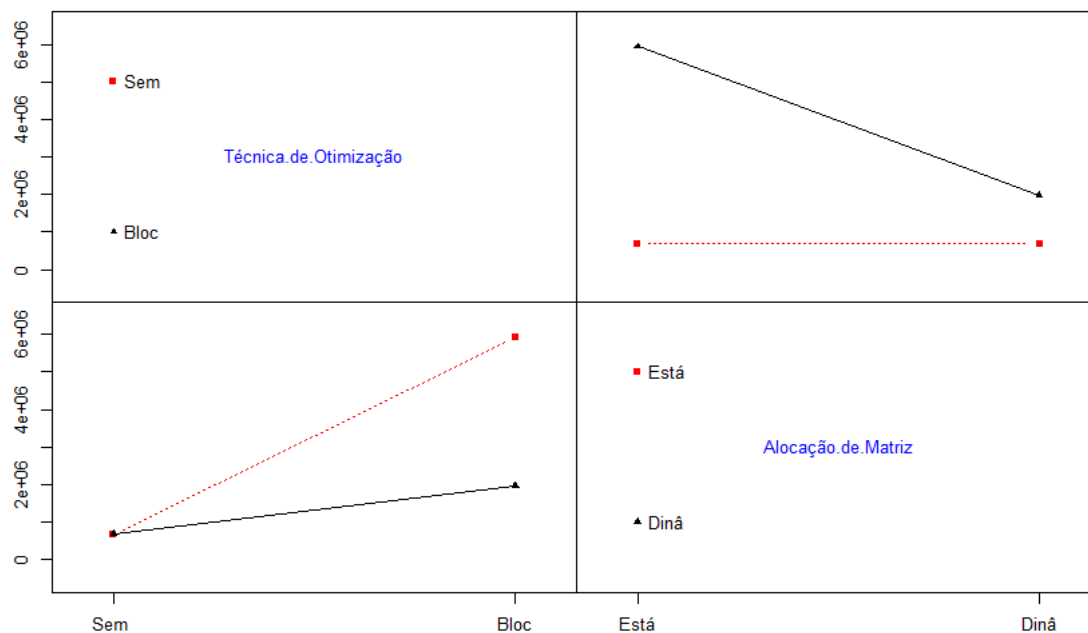
**Conclusão:** A escolha da técnica possui uma influência consideravelmente maior no número de instruções de branch do que o método de alocação das matrizes

Gráficos para métrica **branch misses**:

### Main effects plot for results



### Interaction plot matrix for results



Influência do fator técnica : 0.5783871

Influência do fator alocação : 0.2108575

Influência dos fatores técnica + alocação : 0.2107553

**Conclusão:** A escolha da técnica possui uma influência um pouco maior no número de erros de branch do que o método de alocação de matrizes.

## Conclusão

Após obtermos todos os dados, podemos concluir que na multiplicação de matrizes em memória estática, o método de block tiling tem o menor tempo de resposta, já em memória dinâmica o método loop interchange tem o menor tempo (não sabemos o porque, mas o teste foi consistente em várias iterações).

Nas instruções de desvio, o método loop unrolling obteve o menor número de branch instructions, aproximadamente metade em relação aos demais. Isso se deve ao fato de que, ao expandir manualmente os loops, a quantidade de iterações é reduzida, o que implica menor número de verificações de condição e saltos de controle.

Nos métodos com memória estática, a quantidade de operações de carregamento (cache loads) foi aproximadamente 20% menor em comparação aos métodos com memória dinâmica. Esse resultado confirma que a disposição fixa dos dados em memória facilita a previsão e o reaproveitamento de blocos em cache. Entretanto, o método block tiling apresentou cerca de 10 vezes mais cache misses que os demais.