

Travaux Pratiques Archi SEOC-3A

Prédiction de branchements

Table des matières

1 Identification

Travail réalisé par BENJELLOUN EL KBIBI Youssef

2 Informations

Les expérimentations ont été réalisées avec des tailles de tableaux variant de 5 à 12 bits, c'est-à-dire des tailles de 32 jusqu'à 4096, et des compteurs de 1 à 6 bits.

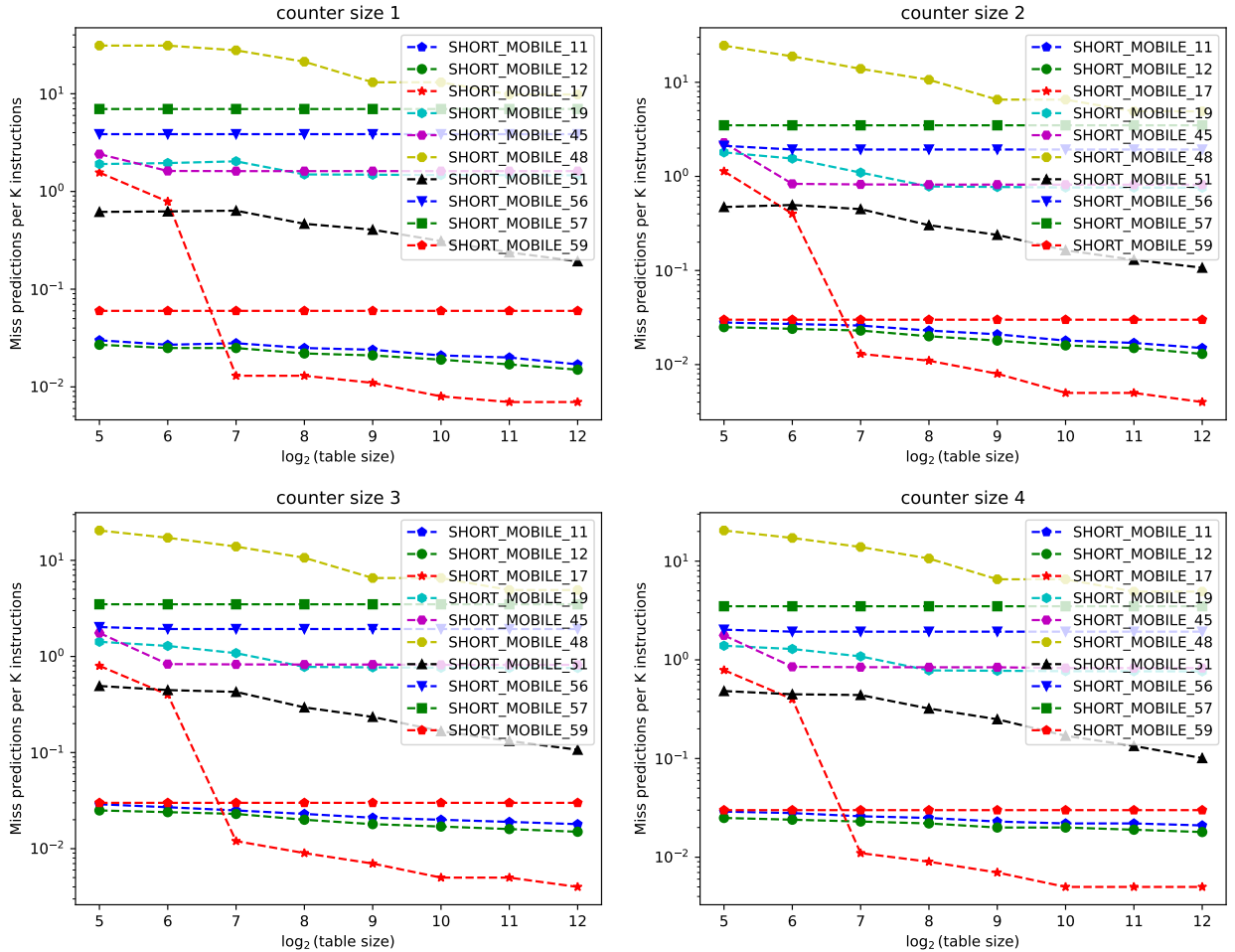
3 Prédicteur n modal : conception et résultats

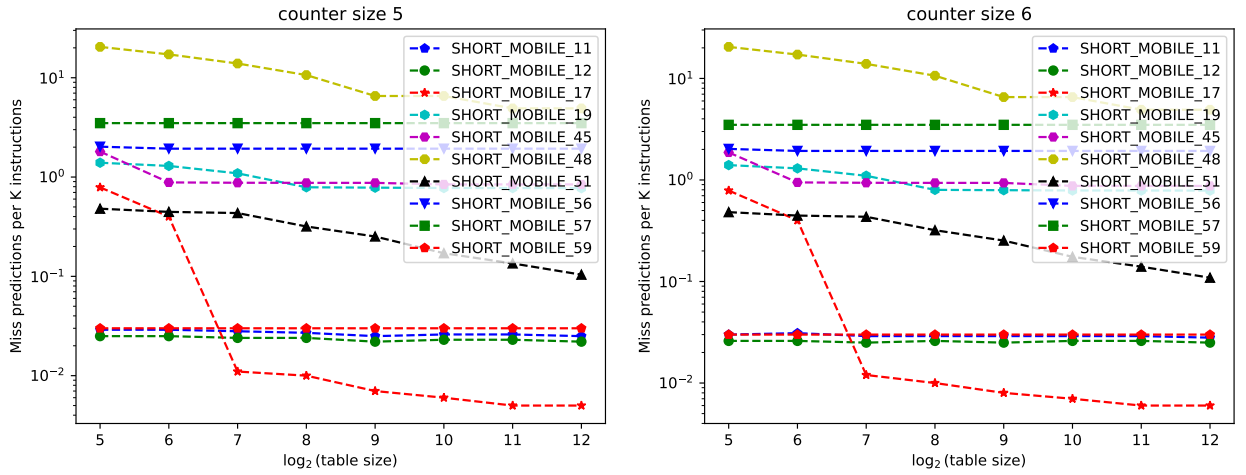
3.1 Conception

Le prédicteur n modal bit est constitué d'un unique tableau contenant de 2^1 à 2^6 bits.

3.2 Résultats

Les résultats issus de la simulation sont les suivants.





3.3 Analyse

On voit une asymptote due à la disparition des collisions lorsque la taille du prédicteur augmente. Le coût du prédicteur est linéaire avec la taille du tableau, et il n'est pas raisonnable de dépasser 2^{16} éléments, d'autant que le gain à partir de 2^{12} devient très faible. Par ailleurs, il y a toujours moins de 7% de mauvaises prédictions, ce qui est remarquable pour une approche aussi simpliste.

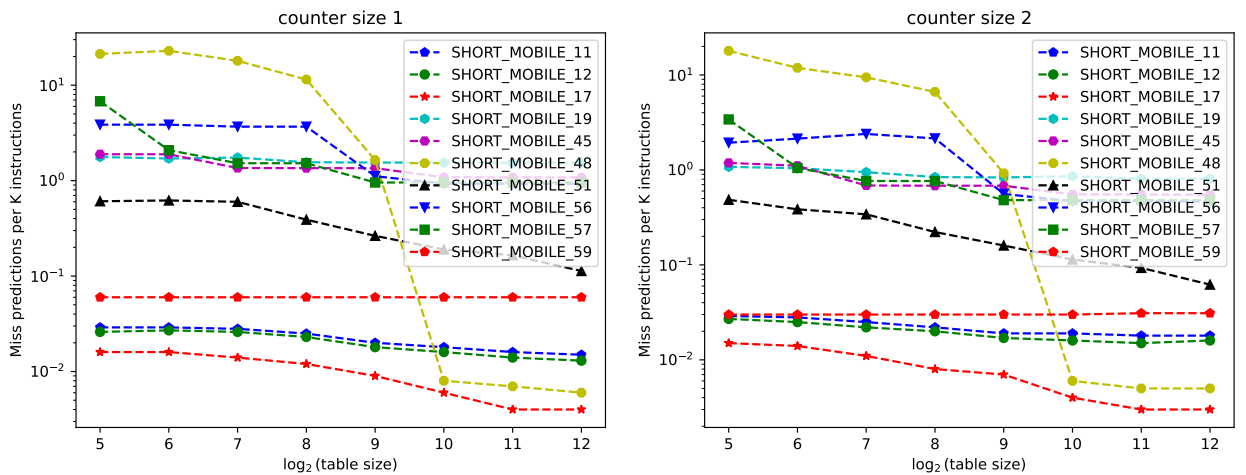
Nous remarquons aussi que les gains pour un *counter size* supérieure à 2 sont négligeables. En ce qui concerne la taille de la table, nous remarquons qu'à partir de 10, on commence à voir une asymptote pour les test 17 mais pour des autres test cette asymptote apparaît beaucoup plus tôt (11,12,57...)

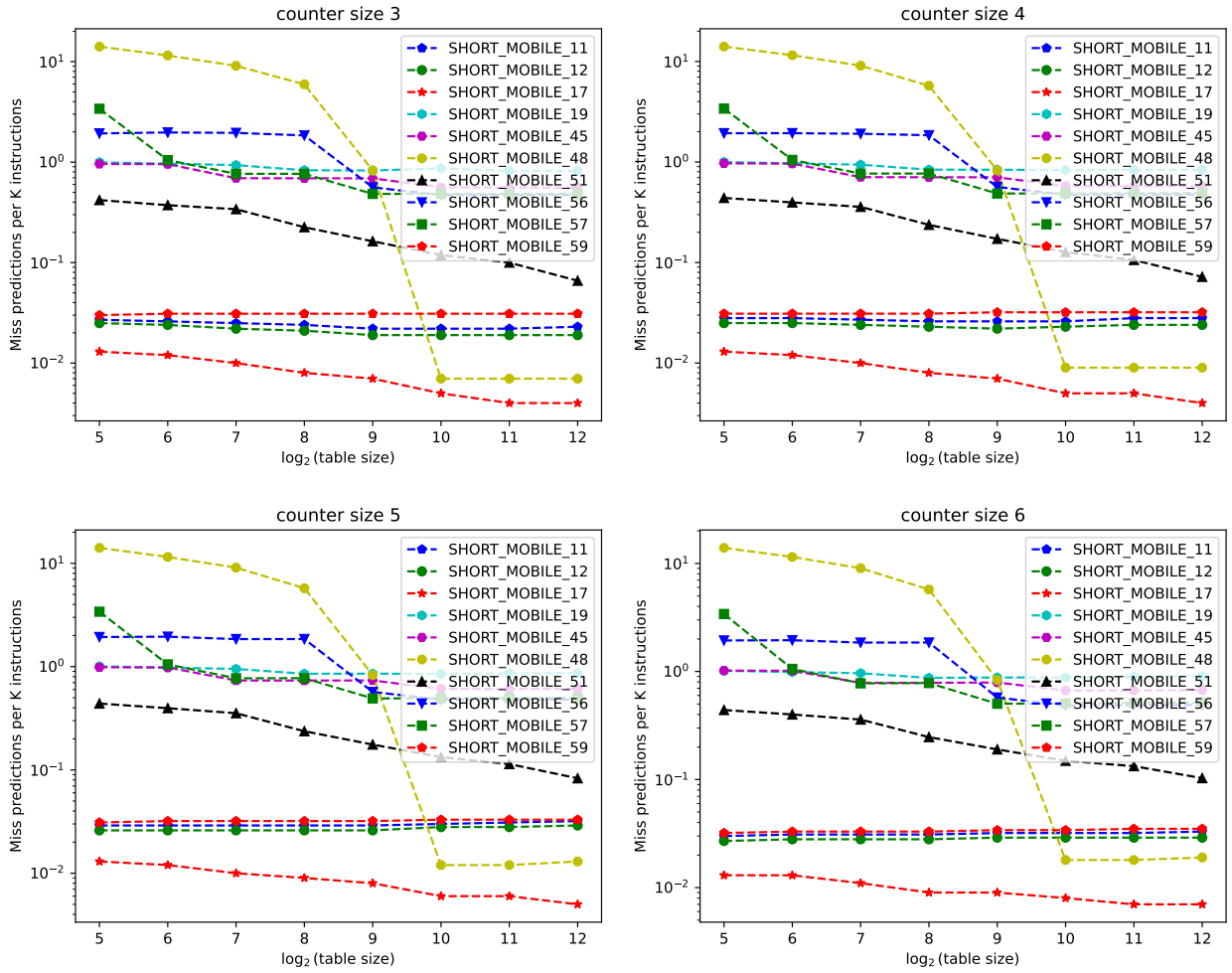
4 G Share

4.1 Conception

On applique le même principe que celui utilisé pour le prédicteur n modal, en incorporant un historique global. Qu'on utilise ensuite pour faire un *xor* avec les bits du PC et indexer la table des compteurs à états.

4.2 Résultats





4.3 Analyse

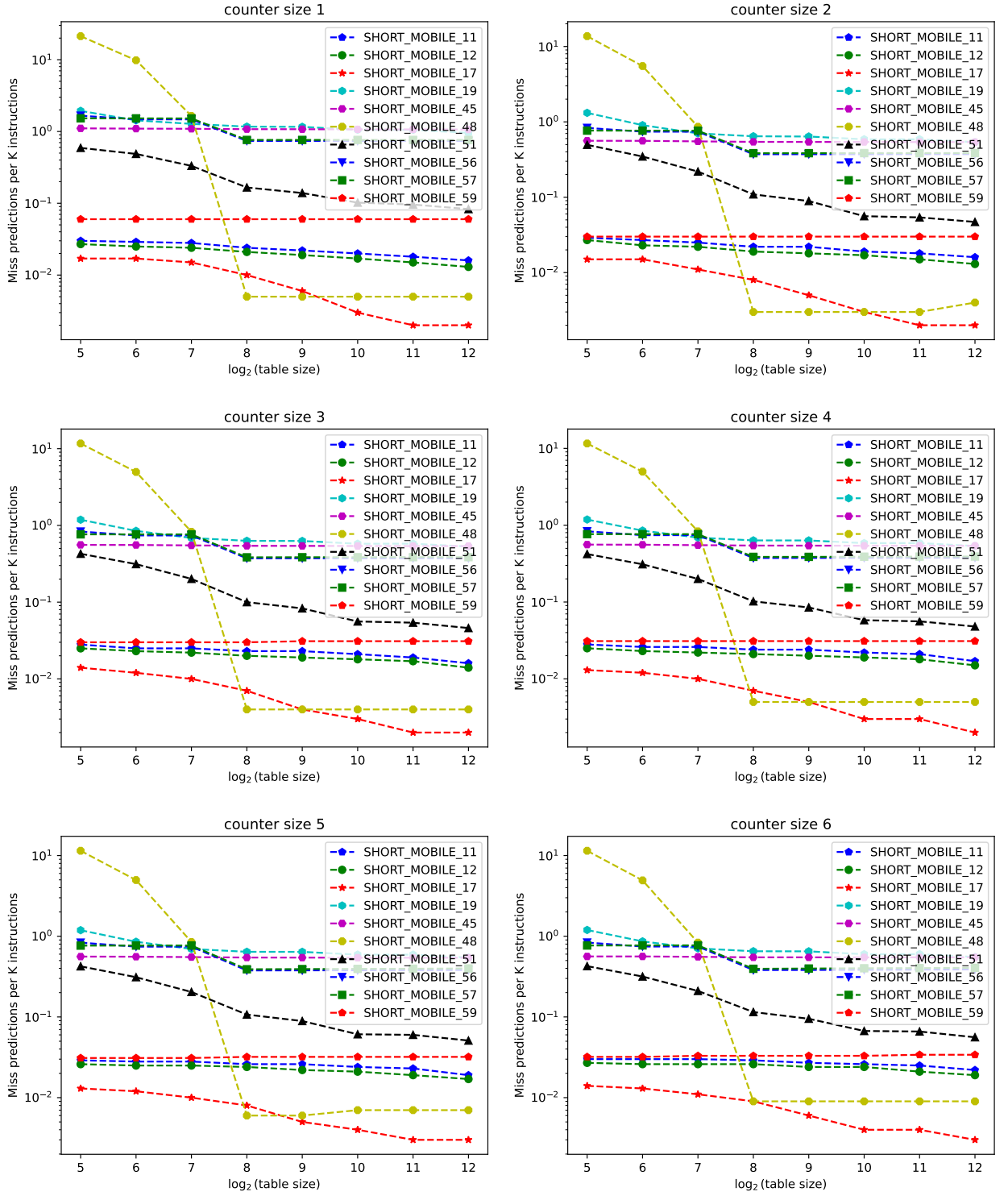
Comme pour le prédicteur précédent, pour une taille de compteur supérieure à 2 bits, les gains sont très faibles et commencent même à être pires, donc la taille idéale semble être 2. Nous observons également une asymptote pour le test 17 à partir d'une *table size* de 2^{10} .

En comparant avec le prédicteur précédent, nous notons que ce prédicteur, pour certains tests comme le 48, parvient à obtenir des performances bien meilleures tout en maintenant les mêmes performances sur les autres tests.

5 Predictor PHT

5.1 Conception

On utilise une table d'historiques (*PHT*) au lieu d'un historique globale, et on fait varier la taille de celle-ci en même temps que pour la taille de la *BHT*.



5.2 Analyse

Nous remarquons que pour des tailles plus petites, notre prédicteur atteint une asymptote beaucoup plus tôt que le *gshare*, surtout pour le test 48. Pour les autres tests, les gains sont peu significatifs, et à partir de 1024 entrées, le *gshare* présente à peu près le même comportement que le prédicteur *PHT*.

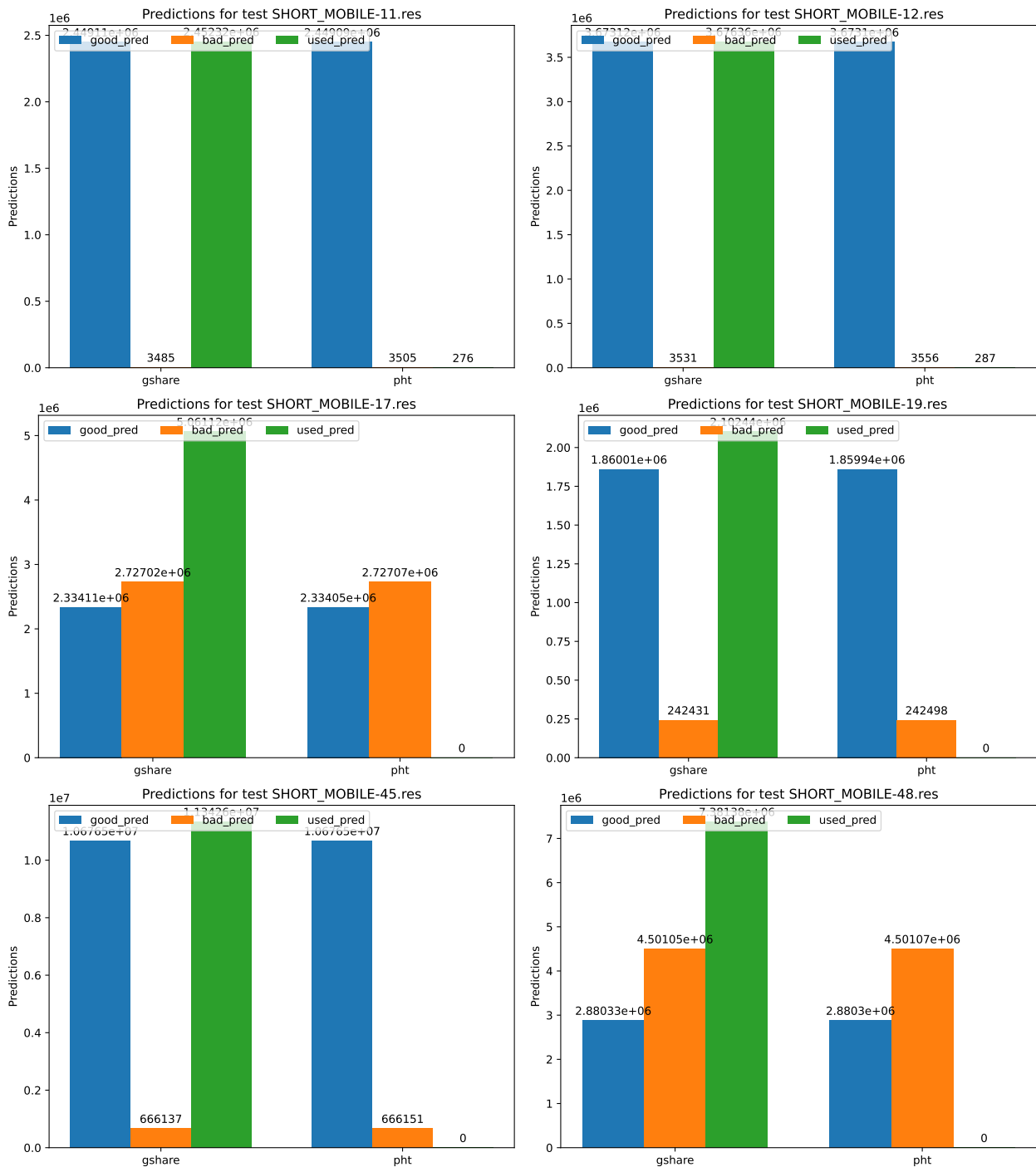
6 Dual predictor

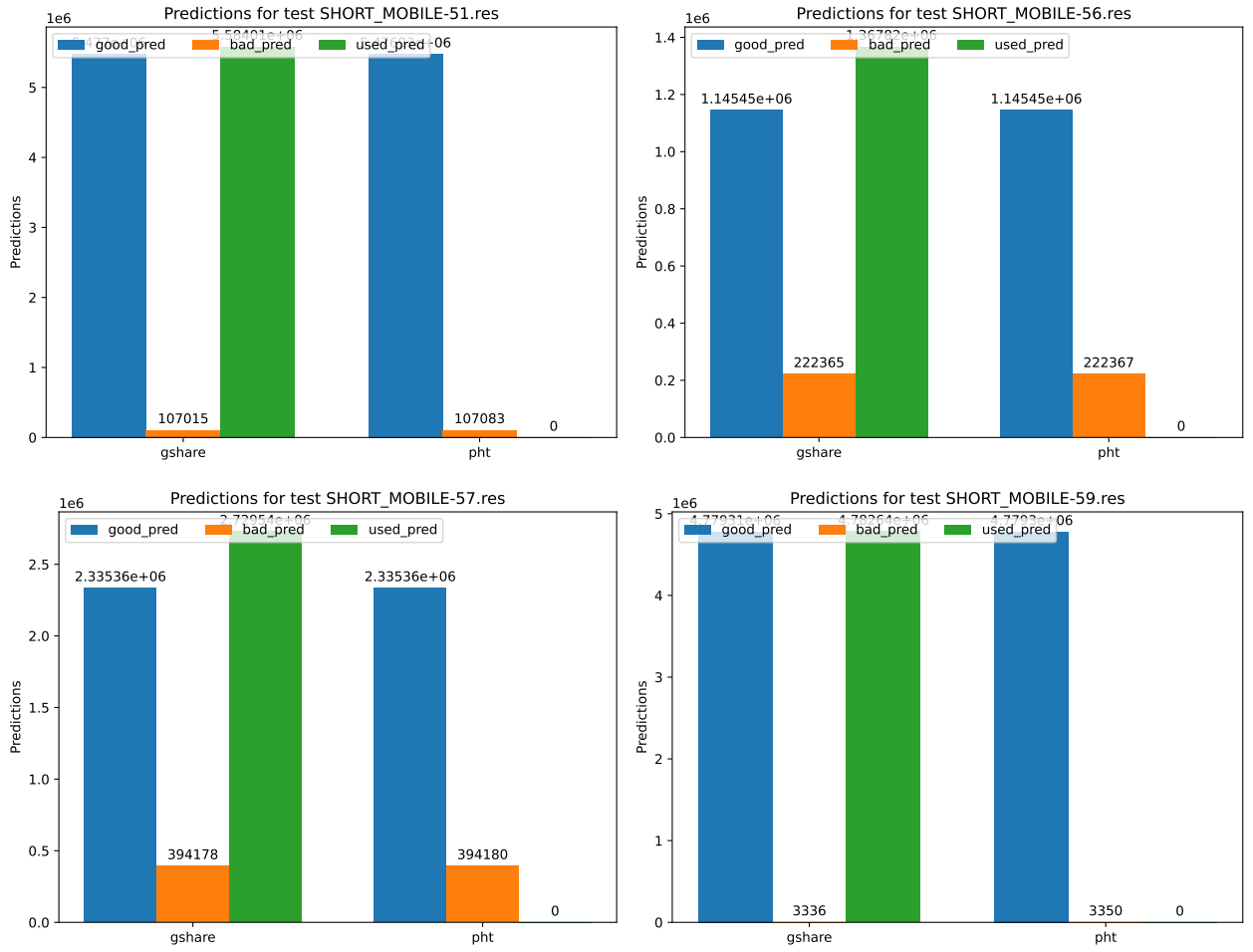
6.1 Conception

Nous construisons les 2 prédicteurs avec les paramètres suivants : *gshare* avec un historique de 1024 entrées et un tableau de compteurs à 2 bits de taille 1024 également. Pour la *PHT*, nous utilisons 1024 entrées, et pour la *BHT*, nous utilisons également 1024 entrées, indexées par des compteurs à 2 bits.

Nous décidons quel prédicteur utiliser grâce au méta-prédicteur qui est composé de 1024 entrées de compteurs bimodaux.

6.2 Résultats





6.3 Analyse

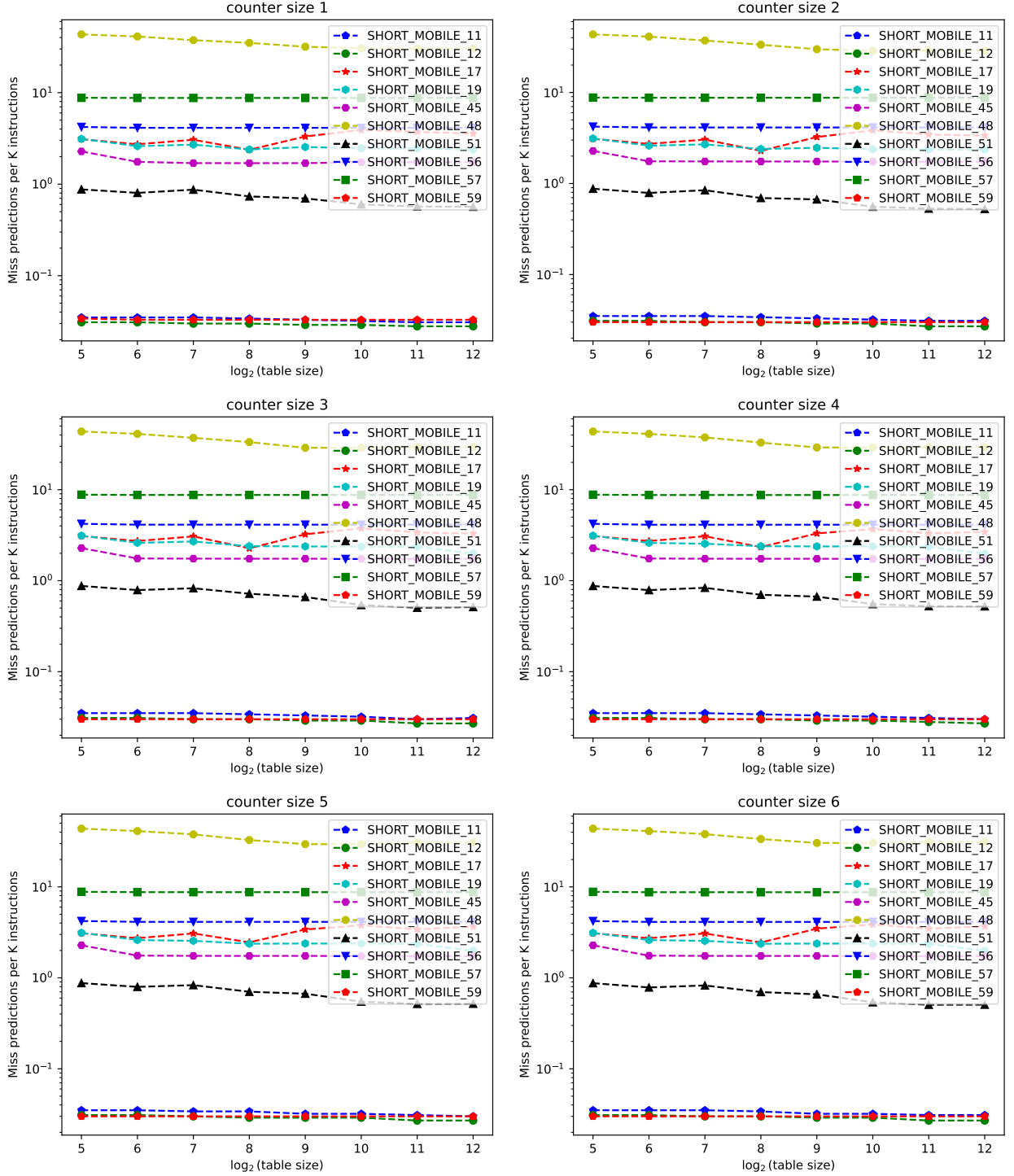
Nous observons un comportement très similaire des deux prédicteurs, ce qui est cohérent avec les résultats précédents. Si nous examinons les graphiques précédents pour une *counter size* de deux et une *table size* de 10, les résultats sont presque identiques. Il aurait été intéressant d'analyser les différences pour des tableaux de taille 2^8 , car nous aurions pu constater une amélioration des performances du prédicteur *PHT* par rapport au *gshare*.

7 Perceptron

7.1 Conception

Les *counter size* dans la section suivante représentent des valeurs de $\theta * 5$. Par exemple, une *counter size* de 5 représente un θ de 25.

7.2 Résultats



7.3 Analyse

Les performances du prédicteur perceptron sont assez décevantes, ne surpassant pas le *gshare* et ajoutant une complexité supplémentaire avec le calcul de y .