

# Décodage SC par listes optimisées de codes polaires

Malek ELLOUZE<sup>1</sup>, Camille LEROUX<sup>1</sup>, Romain TAJAN<sup>1</sup>, Charly POUILLIAT<sup>2</sup> et Christophe JÉGO<sup>1</sup>

<sup>1</sup>Université de Bordeaux, Bordeaux INP, Laboratoire IMS, UMR CNRS 5218, France

<sup>2</sup>Université de Toulouse, INPT-ENSEEIH, Laboratoire IRIT, UMR CNRS 5505, France

malek.ellouze@ims-bordeaux.fr camille.leroux@ims-bordeaux.fr

romain.tajan@ims-bordeaux.fr, charly.poulliat@enseeiht.fr

christophe.jego@ims-bordeaux.fr

**Résumé** – Une étude sur la simplification des étapes de mise à jour et de tri des métriques de l’algorithme de décodage à annulation successive par liste d’un code polaire est décrite dans cet article. Elle repose sur le constat qu’une partie significative des bits fiables sont traités à la fin du processus de décodage. L’algorithme de décodage résultant appelé Tailored CRC-Aided Successive Cancellation List (T-CA-SCL) est une version simplifiée d’un décodage à annulation successive par liste. La pertinence de ce nouvel algorithme de décodage est validée par des expérimentaux qui montrent une réduction de 45% de la mise à jour et du tri des métriques sans impact significatif sur les performances de décodage.

**Abstract** – The simplification of the metric update and sorting steps for the successive cancellation list decoding algorithm of a polar code is described in this paper. It is based on the observation that a non-negligible portion of the reliable bits are located at the tail of the decoding process. The resulting decoding algorithm called Tailored CRC-Aided Successive Cancellation List (T-CA-SCL) is a simplified version of a successive cancellation list decoding. The relevance of this novel decoding algorithm is validated by experimentations that show a 45% reduction in metric updating and sorting without altering the performance.

## 1 Introduction

Les codes polaires [1] atteignent asymptotiquement la capacité de Shannon sur canal sans mémoire. Cependant, les performances d’un code polaire de taille modérée sont limitées lors d’un décodage par annulation successive (SC). Ainsi, le décodage à annulation successive par liste (SCL) a été proposé dans [2]. Cette approche consiste à considérer  $L$  chemins durant le processus de décodage en choisissant les chemins les plus probables. Il est précisé dans [2] que lorsque l’algorithme SCL ne converge pas vers le bon chemin, ce dernier a une très forte probabilité d’être présent dans la liste. C’est pourquoi, il a été proposé dans [2] et [3] une concaténation d’un code de redondance cyclique (CRC) avec un code polaire afin de faciliter la discrimination parmi les différents chemins candidats. L’algorithme de décodage CRC Aided-SCL (CA-SCL) résultant permet de s’approcher des performances optimales pour une taille de code finie. Ces performances sont obtenues au prix d’une complexité calculatoire au moins  $L$  fois supérieure à celle de l’algorithme de décodage SC en raison de la prise en compte et du tri dynamique des  $L$  métriques de la liste.

Il existe principalement deux approches pour réduire la complexité calculatoire de l’algorithme de décodage CA-SCL. La première consiste à réduire la complexité calculatoire de la composante décodage SC [4][5]. L’idée est d’identifier des sous-codes particuliers dans le code

polaire afin d’appliquer localement des traitements parallèles. Cette approche est statique dans le sens où la réduction de complexité ne dépend ni du mot de code, ni du rapport signal à bruit. Elle peut donc bénéficier tout autant à des implémentations matérielles [6] et logicielles [7] du décodeur. Une seconde approche repose sur la réduction de la taille de la liste durant le processus de décodage [8][9]. Cette approche peut permettre de réduire significativement la taille moyenne de la liste. En revanche, elle est plus adaptée pour une implémentation logicielle pour laquelle le pire cas est moins critique.

Dans cet article, nous présentons une méthode qui permet de limiter le nombre de mises à jour des métriques et de leur tri. Contrairement à des méthodes similaires de réduction de taille de la liste [8][9], l’algorithme de décodage proposé a des performances similaires à celles de l’algorithme CA-SCL. De plus, l’adaptation de la liste est définie de manière statique. Cela signifie que la réduction de complexité est similaire quelque soit le mot de code traité. Lors des expérimentations, il a été observé que le décodage T-CA-SCL [10] proposé réduit le nombre de duplications de chemins par  $\sim 35\%$ - $45\%$  selon les codes polaires.

La suite de l’article est organisée de la manière suivante : Dans la section 2, le principe du décodage par liste d’un code polaire est rappelé. Puis, notre contribution, à savoir l’algorithme de décodage tailored-CA-SCL est détaillée

dans la section 3. Enfin des résultats expérimentaux sont présentés et des perspectives sont discutées dans la section 4.

## 2 Le décodage SCL

Un code polaire se définit par une longueur de code  $N = 2^n$ , une dimension du code  $K$  et un rendement  $R = K/N$ . La matrice génératrice du code polaire correspond à une sous-matrice ayant  $K$  lignes dans la transformation suivante  $G_N = G_2^{\otimes n}$  où  $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$  et  $(\cdot)^{\otimes n}$  désigne la  $n^{\text{ième}}$  puissance de Kronecker. Dans ce qui suit, on note par  $a_i^j$  la section  $(a_i, a_{i+1}, \dots, a_j)$  et par  $\hat{a}_i^j$  l'estimation de  $a_i^j$ .  $\mathbf{u} = u_1^N$  est le vecteur d'information binaire. Quant au vecteur  $\mathbf{x} = x_1^N$ , il désigne le mot de code transmis tel que  $\mathbf{x} = \mathbf{u}G_N$ . Les bits de  $\mathbf{x}$  sont transmis sur un canal binaire sans mémoire. Un décodage SC permet d'estimer  $u_i$  à partir du mot reçu  $y_1^N$  et des décisions antérieures  $\hat{u}_1^{i-1}$  telles que :

$$\hat{u}_i = \begin{cases} 0 & \text{si } L(u_i) \geq 0 \\ 1 & \text{sinon} \end{cases} \quad (1)$$

où  $L(u_i)$  est défini comme :

$$L(u_i) = \log \left( \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | u_i = 0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1} | u_i = 1)} \right) \quad (2)$$

et où  $W_N^{(i)}$  ont été introduits dans [1] tel que :

$$W_N^{(i)}(\mathbf{y}, u_1^{i-1} | u_i) = \sum_{u_{i+1}^N \in \{0,1\}^{N-i}} \frac{1}{2^{N-i}} W_N(\mathbf{y} | G_N \mathbf{u}). \quad (3)$$

Concevoir un code polaire revient donc à désigner les  $K$  positions des bits d'information, tandis que les positions restantes sont *gelées*. L'ensemble des positions des bits gelés est noté  $\mathcal{F}$ .

L'algorithme de décodage par liste d'un code polaire est une évolution d'un décodage SC. Au lieu de prendre des décisions uniquement basées sur la valeur du LLR comme dans (1), les deux solutions de décodage sont sauvegardées dans une liste de taille  $L$ . Ensuite, chaque chemin gardé par le SCL est mis à jour selon un processus de décodage SC. Une métrique est introduite pour pénaliser un chemin vis à vis de l'autre. A l'étage de décodage  $i$ , le  $j^{\text{ème}}$  décodeur SC évalue le LLR  $L_j^{(i)} = L_j(u_i)$  et la métrique est mise à jour comme suit :

$$m_j^{(i)} = \begin{cases} m_j^{(i-1)} & \text{si } L_j^{(i)} \geq 0 \text{ et } \hat{u}_j^i = 0 \\ m_j^{(i-1)} + |L_j^{(i)}| & \text{si } L_j^{(i)} < 0 \text{ et } \hat{u}_j^i = 0 \\ m_j^{(i-1)} + |L_j^{(i)}| & \text{si } L_j^{(i)} \geq 0 \text{ et } \hat{u}_j^i = 1 \\ m_j^{(i-1)} & \text{si } L_j^{(i)} < 0 \text{ et } \hat{u}_j^i = 1 \end{cases} \quad (4)$$

où  $|\cdot|$  désigne l'opération valeur absolue. Les  $L$  meilleurs candidats, c'est à dire ceux ayant les métriques les plus

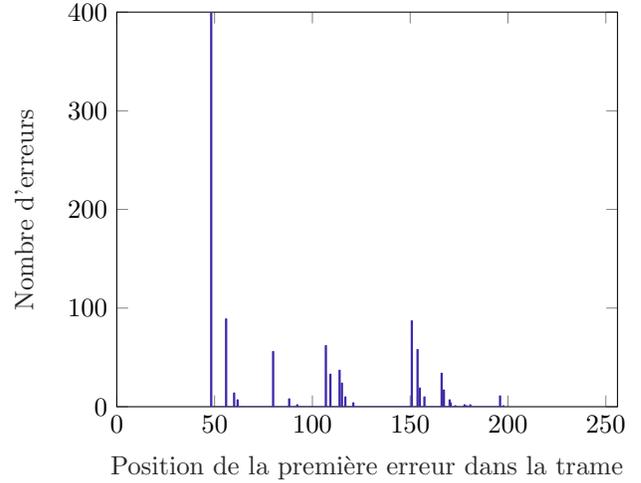


FIG. 1: Histogramme de la première erreur lors d'un décodage SC d'un code polaire (256,128) pour  $\frac{E_b}{N_0} = 3$  dB

faibles, sont ensuite sélectionnés. A la fin du processus de décodage, le chemin ayant la plus faible métrique globale est retenu. L'algorithme de décodage CA-SCL est une amélioration du décodage SCL. Il permet de choisir le chemin candidat ayant un CRC valide. Puis, si aucun ou plusieurs chemins ont un CRC valide alors le chemin candidat qui a la métrique la plus faible est retenu.

## 3 Décodage SC par listes optimisées

Il a été observé dans [2] qu'un algorithme SC assisté par un "génie" améliore les performances de décodage. Au cours de l'expérimentation menée, un "génie" est autorisé à corriger jusqu'à deux erreurs durant un processus de décodage SC. Cela signifie que les premières erreurs ont un très grand impact sur les performances du décodage. Dans la Figure 1, l'histogramme de la position de la première erreur du décodage SC pour 1000 décodages SC ayant échoué pour un code polaire (256, 128) est présenté. Cette simulation de Monte-Carlo montre que la fin de la séquence de décodage que nous appelons "tail" ne comporte quasiment pas d'erreurs. L'application d'un décodage SCL à la fin de la séquence est donc superflue. Un simple décodage SC permet d'atteindre des performances similaires. A partir de cette observation, une simplification de l'algorithme CA-SCL dans laquelle le "tail" de la séquence est décodé avec un décodeur SC peut être formulée. Ainsi, à partir d'un certain index  $\omega = N - T$ , les métriques des chemins ne sont plus mises à jour et le processus de tri est supprimé. Seuls  $L$  chemins sont gardés et un décodage SC est appliqué à chaque chemin. En considérant le code polaire (256,128) de la Figure 1, si  $T = 256$  alors le processus de décodage est équivalent à un décodage SC sur  $L$  chemins identiques. Lorsque  $T = 0$ , le processus de décodage est

---

**Algorithme 1:** T-CA-SCL decoding

---

```
1 Entrée Code polaire  $\mathcal{P}(N, K, \mathcal{F}), T, S, L$ 
2 pour  $i \in \llbracket 1; N - T \rrbracket$  faire
3 | Effectuer la  $i^{\text{ème}}$  étape de SCL
4 fin
5  $\mathcal{L} \leftarrow$  liste des chemins gardés par le SCL
6 pour  $i \in \llbracket N - T + 1; N \rrbracket$  faire
7 | si  $i \notin \mathcal{F}$  and  $i \notin S$  alors
8 | | pour chaque  $l \in \mathcal{L}$  faire
9 | | | Décodage SC de  $u_i$  pour le chemin  $l$ 
10 | | fin
11 | sinon
12 | | Effectuer la  $i^{\text{ème}}$  étape de SC
13 | fin
14 fin
15  $valide \leftarrow$  Le chemin ayant la plus faible métrique
   et qui vérifie le CRC
16 si  $valide = faux$  alors
17 | Signaler un échec de décodage
18 sinon
19 | retourner  $u[valide]$ 
20 fin
```

---

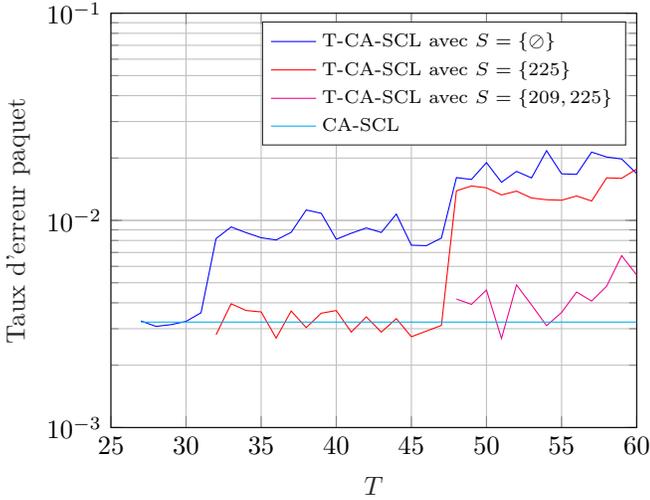


FIG. 2: Taux d'erreur trame selon  $T$ . Code polaire (256,128),  $L = 8$ ,  $L_{CRC} = 24$  et  $\frac{E_b}{N_0} = 3$  dB.

équivalent à un décodage CA-SCL. Pour des valeurs intermédiaires de  $T$ , quelques pics d'erreurs sont observées dans la zone du "tail". En effet, le SC est sensible aux canaux les moins fiables. De plus lorsque  $T$  est suffisamment petit ( $T < 32$ ), aucune dégradation significative du taux d'erreur trame n'a été observée. Les performances sont alors similaires à celles d'un décodage CA-SCL.

L'évolution du taux d'erreur trame (FER) selon la taille du "tail"  $T$  est présentée pour le code polaire (256,128) dans la Figure 2. Pour  $T < 32$ , le FER de l'algorithme T-CA-SCL est équivalent à celui de l'algorithme CA-SCL.

Pour  $T > 31$ , une perte de performance est observée. En revanche, si un traitement de type SCL est appliqué à la position  $i = 225$ , il n'y a plus de dégradation des performances jusqu'à  $T = 47$ . L'algorithme 1 détaille l'algorithme T-CA-SCL proposé. Le processus de décodage se déroule suivant deux étapes successives. La première étape consiste au décodage de  $u_1^{N-T}$ . Durant cette étape, l'algorithme T-CA-SCL est identique à un décodage SCL classique. A la fin de cette étape,  $\mathcal{L}$  désigne la liste des chemins candidats. Puis, la seconde étape débute par le décodage du  $N - T + 1^{\text{ème}}$  bit. Il y a deux alternatives au niveau de chaque étage de décodage. Si  $i \notin S$  alors les bits à la fin de la séquence reçue favorisent un chemin probable. C'est pourquoi, un décodeur SC est appliqué sur chacun des chemins gardés. Cela signifie qu'aucun tri et aucune mise à jour des métriques sont nécessaires. Si  $i \in S$  alors le décodage SCL est appliqué à ces bits spécifiques.

Il faut noter que les bits gelés sont mis à zéro et sont omis dans l'algorithme 1. Un contrôle CRC est ensuite appliqué pour conclure sur la séquence décodée. L'algorithme T-CA-SCL a une complexité calculatoire moins élevée grâce à la suppression des mises à jours et des tris de métriques pour un nombre significatif d'étages de décodage.

## 4 Résultats expérimentaux

Nous commençons par introduire  $S$  comme l'ensemble des indices pour lesquels un décodage SC est appliqué. Avant d'évaluer les performances de l'algorithme T-CA-SCL proposé, les paramètres  $T$  et  $S$  doivent être estimés. Pour ce faire, l'heuristique décrite dans l'algorithme (2) permet d'effectuer des simulations de Monte Carlo sur 100000 séquences avec un décodage T-CA-SCL en augmentant progressivement la taille du "tail"  $T$ . Comme expliqué préalablement, certains bits peu fiables peuvent se retrouver à la fin de la séquence. Dans le but d'augmenter la valeur de  $T$  sans dégrader les performances, un décodage de type liste est appliqué sur les bits peu fiables.

L'algorithme 2 a été appliqué à six codes polaires ( $R = \{1/2, 3/4\}$  et  $N = \{128, 256, 512\}$ ) sélectionnés à partir du standard 5G [11]. Les valeurs de  $T$  et de  $S$  obtenues pour ces codes sont récapitulées dans la Table 1. Le ratio  $\rho = (T - |S|)/K$  désigne la proportion des bits décodés sans envisager de duplication. Selon le rendement considéré, ce ratio peut atteindre  $\sim 45\%$ . En d'autres termes, près de la moitié des bits d'information sont décodés sans mise à jour ni tri des métriques. Des simulations du taux d'erreur trame sous canal AWGN ont été effectuées pour les six codes polaires. Les courbes des figures 3 et 4 confirment que la sélection des paramètres  $T$  et  $S$  permettent à l'algorithme T-CA-SCL d'avoir des performances similaires à celles de l'algorithme CA-SCL. Des résultats identiques ont été obtenus lors de décodages avec de plus grandes tailles de liste.

---

**Algorithme 2:** Génération des paramètres de l'algorithme de décodage T-CA-SCL

---

```

1 Entrée Polar code  $\mathcal{P}(N, K, \mathcal{F})$ 
2  $S = \{\emptyset\}, T = 0, j = 0;$ 
3  $\mathbf{Y}$  : set de 10000 mots de codes bruités;
4  $FER \leftarrow$  T-CA-SCL( $\mathcal{P}, T, S, \mathbf{Y}$ );
5 tant que  $FER \sim FER_{CA-SCL}$  faire
6   |  $FER \leftarrow$  T-CA-SCL( $\mathcal{P}, T, S, \mathbf{Y}$ );
7   | si  $FER \sim FER_{CA-SCL}$  alors
8     |  $T \leftarrow T + 1;$ 
9   | sinon
10    |  $s_j = T;$ 
11    |  $j \leftarrow j + 1;$ 
12  | fin
13 fin
14 retourner ( $T, S$ )

```

---

## 5 Conclusion

Dans cet article, nous proposons une méthode de réduction de la mise à jour des métriques et de leur tri. Ainsi, à partir d'une certaine position,  $L$  décodages SC exécutés en parallèle remplacent le décodage SCL classique. L'algorithme de décodage T-CA-SCL résultant permet d'éviter la complexité du tri. Les résultats expérimentaux montrent que la complexité calculatoire est significativement réduite pour des performances de décodage similaires.

TAB. 1: Paramètres des codes polaires et de T-CA-SCL

$N$	128	256	512	128	256	512
$K$	64	128	256	96	192	384
$R$	1/2	1/2	1/2	3/4	3/4	3/4
$S$	113	209,225	289	97	$\emptyset$	$\emptyset$
$ S $	1	2	1	1	0	0
$T$	23	54	93	45	87	126
$\rho(\%)$	34	36	36	46	45	32

## References

- [1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. on Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.
- [2] I. Tal and A. Vardy, "List decoding of polar codes," in *2011 IEEE Int. Symposium on Inf. Theory Proceedings*, pp. 1–5.
- [3] K. Niu and K. Chen, "Crc-aided decoding of polar codes," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1668–1671.
- [4] A. Alamdar-Yazdi and F. R. Kschischang, "A simplified successive-cancellation decoder for polar codes," *IEEE Communications Letters*, vol. 15, no. 12, pp. 1378–1380, 2011.
- [5] G. Sarkis, P. Giard, A. Vardy, C. Thibault, and W. J. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 5, pp. 946–957, 2014.

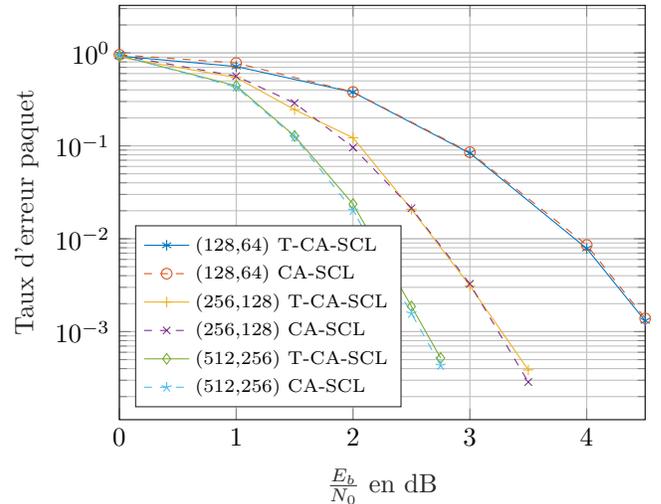


FIG. 3: CA-SCL versus T-CA-SCL.  $R = 1/2$  et  $L = 8$

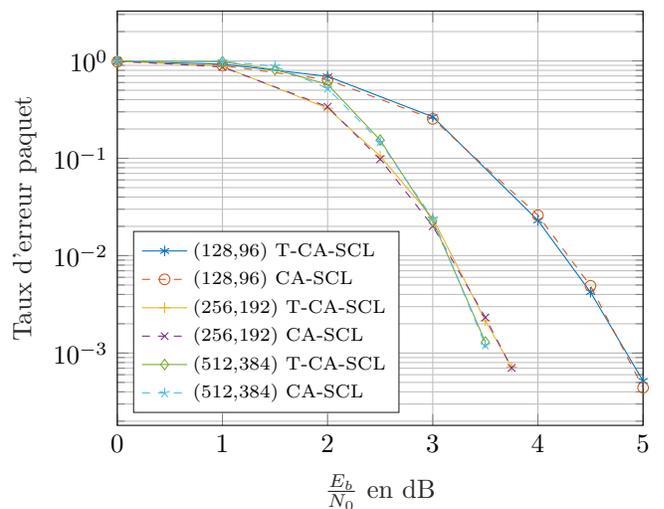


FIG. 4: CA-SCL versus T-CA-SCL.  $R = 3/4$  et  $L = 8$

- [6] P. Giard, G. Sarkis, A. Balatsoukas-Stimming, Y. Fan, C.-y. Tsui, A. Burg, C. Thibault, and W. J. Gross, "Hardware decoders for polar codes: An overview," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016, pp. 149–152.
- [7] M. Léonardon, A. Cassagne, C. Leroux, C. Jégo, L.-P. Hamelin, and Y. Savaria, "Fast and flexible software polar list decoders," *submitted to Journal of Signal Processing Systems*, vol. 91, 08 2019.
- [8] Z. Zhang, L. Zhang, X. Wang, C. Zhong, and H. V. Poor, "A split-reduced successive cancellation list decoder for polar codes," *IEEE J. Sel. Areas in Commun.*, vol. 34, no. 2, pp. 292–302, 2016.
- [9] L. Zhang, A. Cao, J. Qiao, and Y. He, "A CRC-Aided SR-SCL decoding algorithm for polar codes," in *2019 IEEE MTT-S International Wireless Symposium (IWS)*, 2019, pp. 1–3.
- [10] M. Ellouze, C. Leroux, R. Tajan, C. Poulliat, and C. Jégo, "Tailored list decoding of polar codes," in *2021 11th International Symposium on Topics in Coding (ISTC)*, 2021, pp. 1–5.
- [11] "Multiplexing and channel coding," 3GPP TS 38.212 version 15.2.0 Release 15.