Optimisation de circuits logiques

Alexandre JANNIAUX

Circuits logiques et fonctions combinatoires

- **Circuit = Fonction logique**

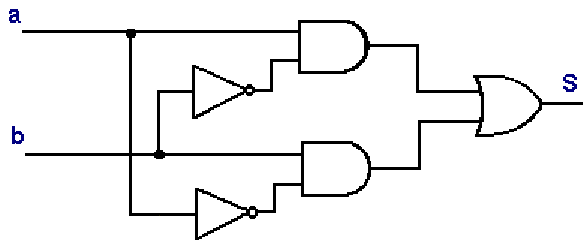
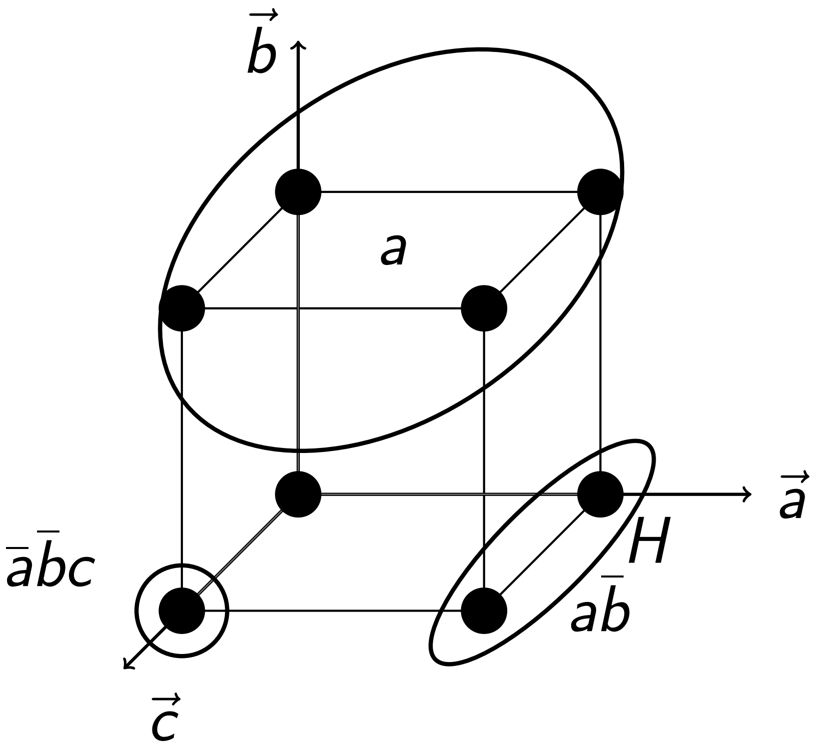


FIGURE : Exemple de circuit logique associé à $f = a\bar{b} + \bar{a}b$

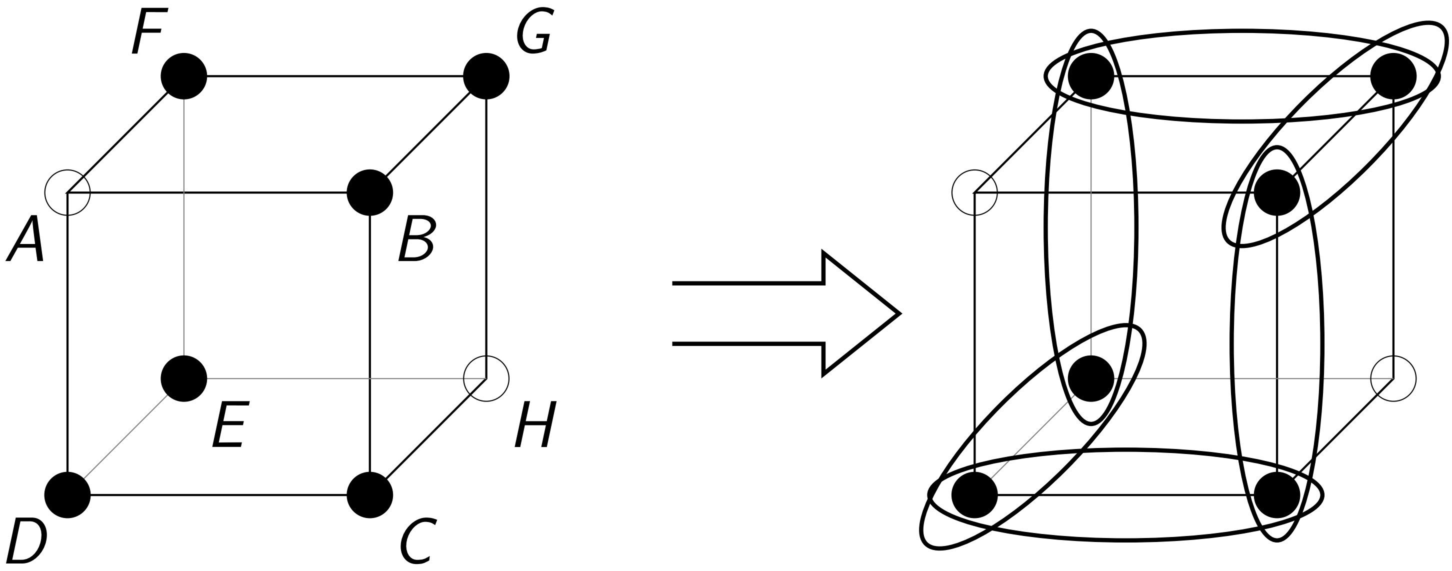
- **Problème de minimisation :** trouver la forme de la fonction avec le moins de terme possible.

- **Représentation :** Le cube

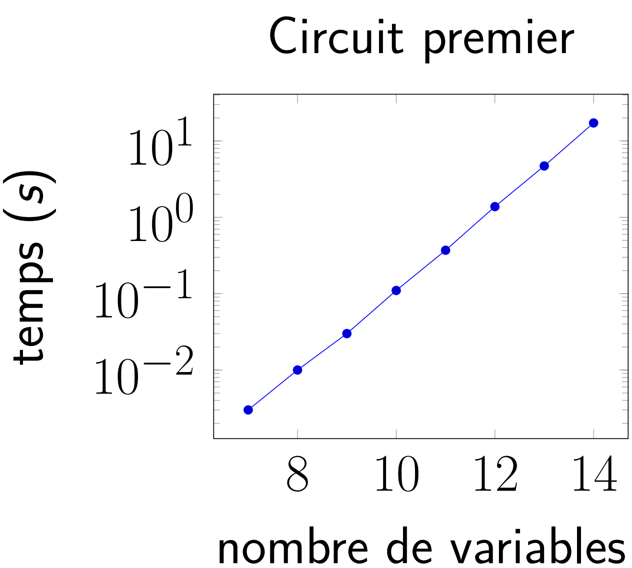


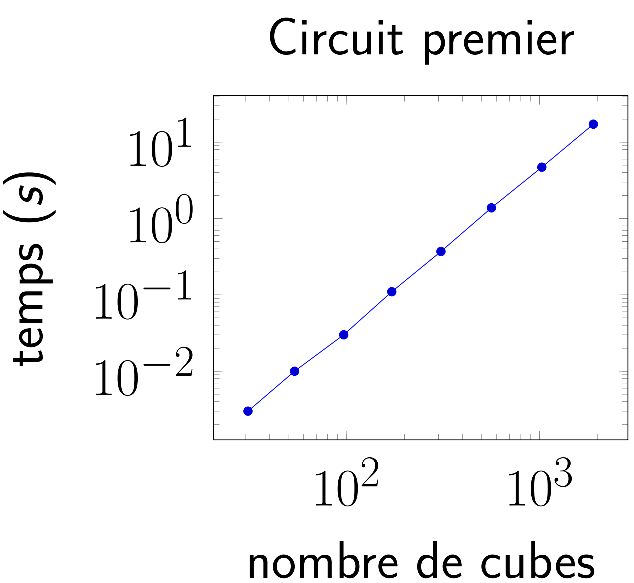
Méthode de Quine-McCluskey

Expansion des cubes :



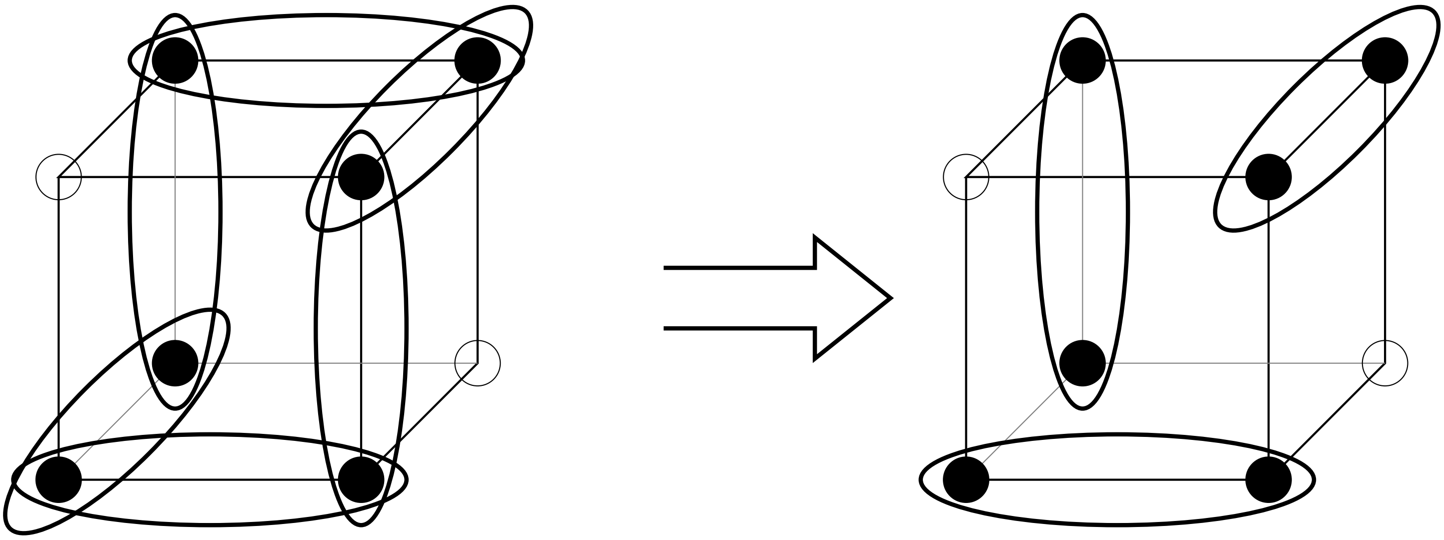
Performances :





Conclusion : Coissance exponentielle en le nombre de variable.

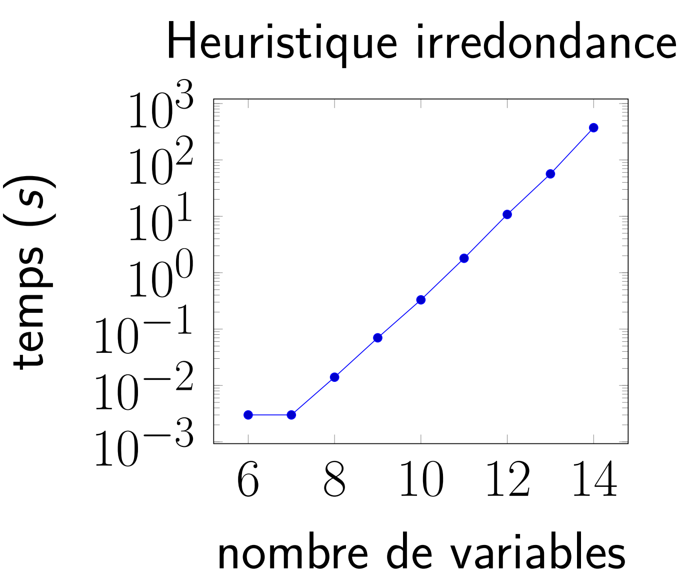
Suppression des redondances inutiles :

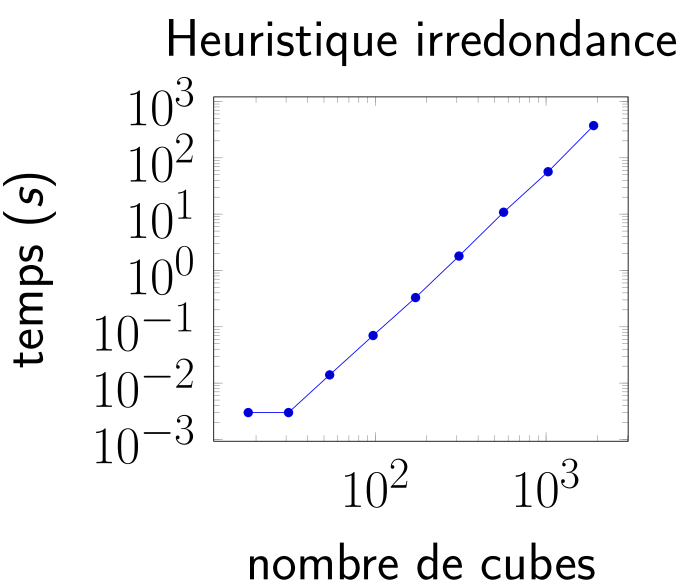


Méthodes :

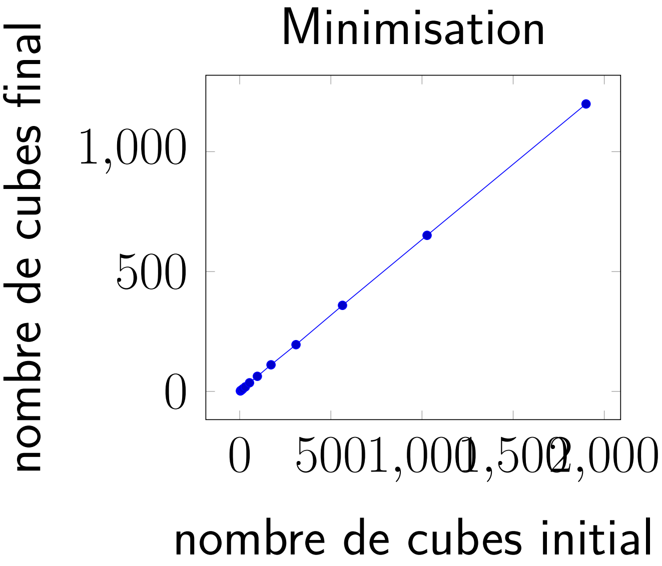
- Développement naïf : débordement pour $n \geq 5$
- Backtracking : $t > 20600s$ pour $n \geq 7$
- Algorithme de couverture minimale glouton

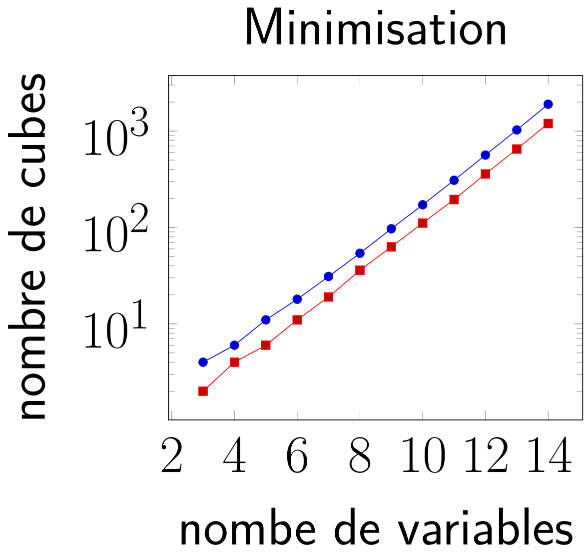
Performances

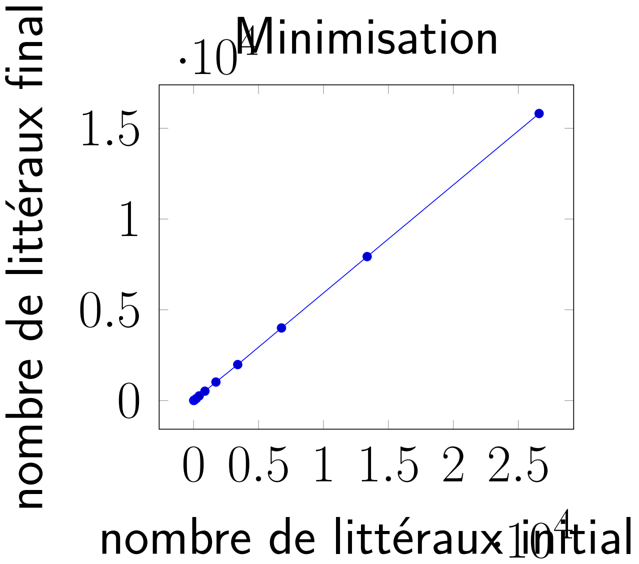


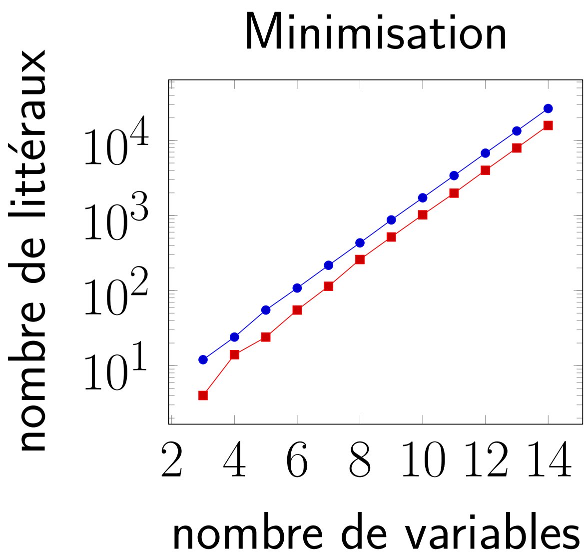


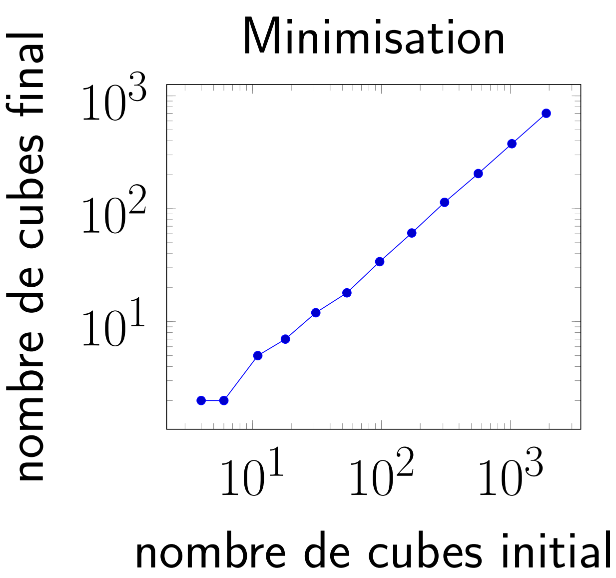
Résultats


Minimisation
nombre de cubes final
1,000
500
0
0
500
1,000
1,500
2,000
nombre de cubes initial


Minimisation
nombre de cubes
10^3
10^2
10^1
2 4 6 8 10 12 14
nombe de variables


Minimisation
$\cdot 10^4$
nombre de littéraux final
1.5
1
0.5
0
0 0.5 1 1.5 2 2.5
$\cdot 10^4$
nombre de littéraux initial


Minimisation
nombre de littéraux
10^4
10^3
10^2
10^1
2 4 6 8 10 12 14
nombre de variables


Minimisation
nombre de cubes final
10^3
10^2
10^1
10^1 10^2 10^3
nombre de cubes initial

Fonctions multivaluées

Définition :

$$f : \mathcal{P}_1 \times \cdots \times \mathcal{P}_n \longrightarrow \mathbb{B}^m$$

Littéraux :

$$X_i^{S_i} = \begin{cases} 1 \text{ si } X_i \in S_i \\ 0 \text{ sinon} \end{cases} \qquad \text{où } X_i \in \mathcal{P}_i \text{ et } \mathcal{S}_i \subset \mathcal{P}_i$$

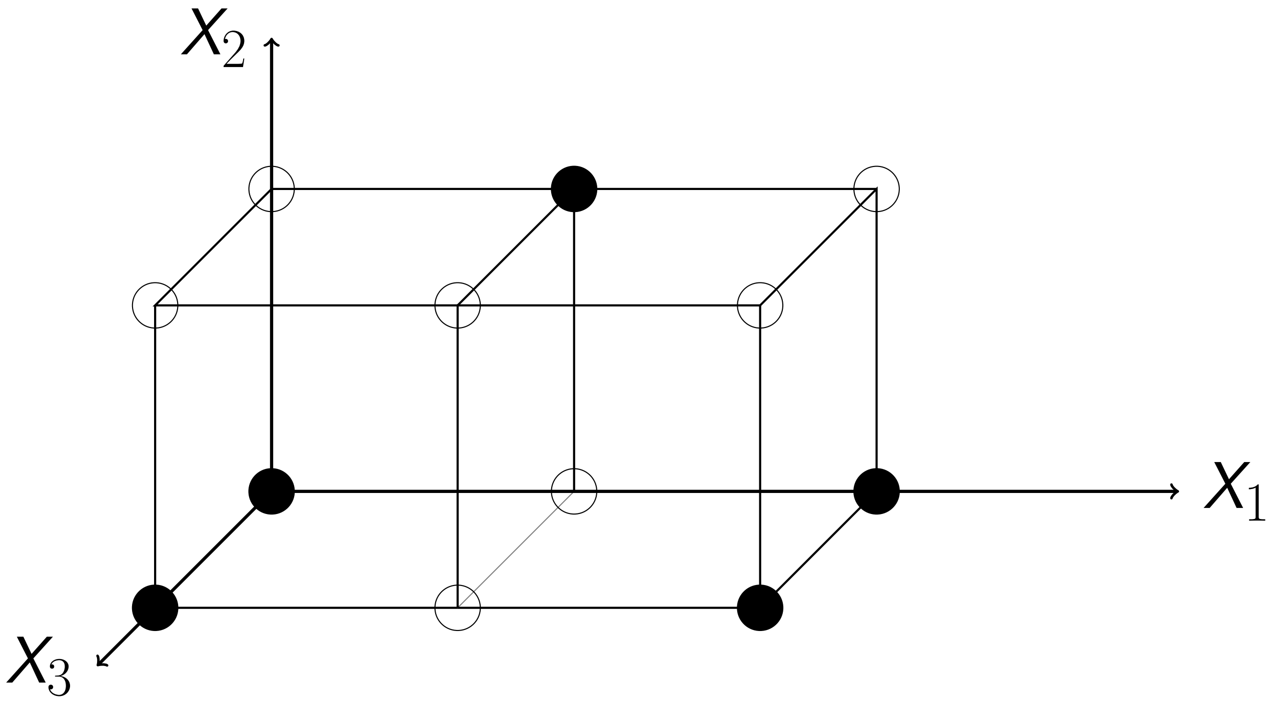


FIGURE : Représentation de $f = X_1^{\{0,2\}} X_2^{\{0\}} + X_1^{\{1\}} X_2^{\{1\}} X_3^{\{0\}}$

Décomposition de Shannon :

$$\bigcup_{i=1}^{n} c_i = 1 \implies f \equiv \bigcup_{i=1}^{n} c_i \cap f|_{c_i}$$

Espresso

$$\bigcup_{i=1}^{n} c_i = 1 \implies f \equiv \bigcup_{i=1}^{n} c_i \cap f|_{c_i}$$

Atout : Optimisation récursive

Initialisation :

- Simplification
- Élimination des redondances
- Recherche des impliquants essentiels

Algorithme :

- Réduction d'un impliquant.
- Simplification
- Élimination des redondances

Conclusion

- Problème difficile...
- ...mais bien maîtrisé.
- Approximable facilement.
- Généralisable (XOR-NAND)