IA POUR LES JEUX

Génération de réseaux de tri

Guillaume FREYERMUTH

INSA Rennes

2 Avril 2024

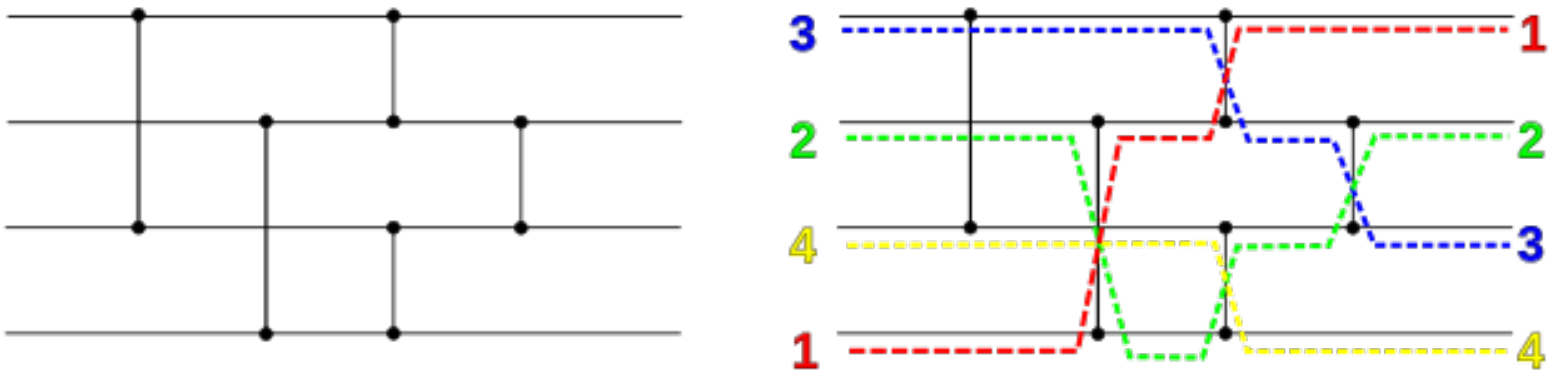
Table of contents

1. Qu'est-ce qu'un réseau de tri

Résau de tri

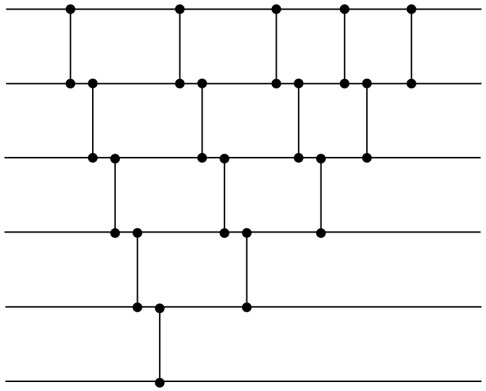
- N nombres désordonnés en entrée
- N nombres ordonnés en sortie
- Collection de comparateurs

Représentation graphique

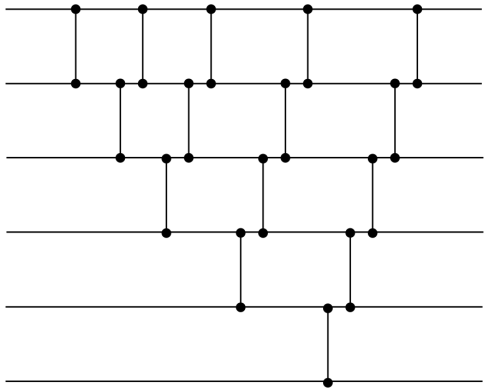


Exemple d'un réseau de tri pour N=4

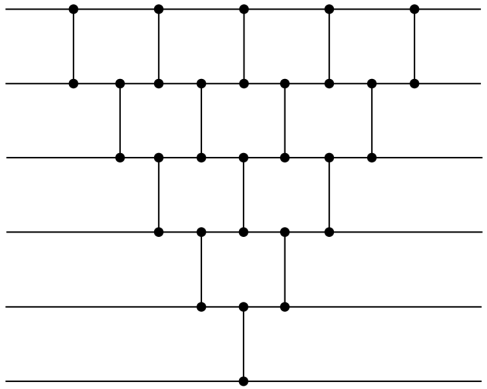
Tri à bulle & Tri par insertion



Bubble sort



Tri par insertion



Version parallèle

Fonctions à optimiser

Nombre de comparateurs (Taille)

La recherche de réseaux de tri utilisant le plus faible nombre de comparateurs (coût minimum en silicium)

Etages de comparaison (Profondeur)

La recherche de résaux de tri tirant au mieux parti de la parallélisation. (réseau le plus rapide)

NB : Il n'existe pas toujours un réseau optimal pour les deux critères à la fois

2. Construction de réseaux optimaux

Cadre du projet

- Optimisation de la taille uniquement
- Caractérisation du problème
- Evaluation de la performance d'un algorithme génétique pour la construction de réseaux de tri
- Evaluation de l'apport de filtres à la recherche exhaustive

3. Validité d'un réseau de tri

Complexité

Le problème consistant à vérifier la validité d'un réseau donné comme réseau de tri appartient à la classe des problèmes co-NP-complet. Pour en vérifier la validité, il n'y a pas d'autre choix que de tester toutes les combinaisons possibles en entrée. Pour un réseau de taille N, avec des nombres sur B bits, cela revient à tester $A_{2^B}^N$ entrées.

Ordre de grandeur

En considérant B figé, la complexité est de l'ordre de O(N!) avec des constantes cachées importantes.

Zero-One principle

Il est possible de démontrer que si un réseau de tri est valide pour des entrées dans l'ensemble $\{0, 1\}$, alors il est valide pour n'importe quel ensemble de nombres

Complexité avec le Zero-One principle

On a désormais une complexité de $O(2^N)$ pour la vérification

4. Recherche avec arbre

Utilisation de BFS

Permet la génération d'une solution optimale avec consommation mémoire et temporelle toutes les deux exponentielles

Utilisation de Iterative Deepening

Permet la génération d'une solution optimale avec consommation mémoire en $log(n)$ et une complexité temporelle exponentielle.

Simplification par préfixe

Possibilité de commencer la recherche avec un début de réseau déjà construit. Plus la taille du préfixe est grande, plus le temps de calcul de la recherche est réduit.

Optimalité des préfixes

L'utilisation des préfixes ne garantie plus de trouver une solution optimale au problème. Cependant, cela peut permettre de trouver des bornes supérieures

Lemme de Van Voorhis

On peut déterminer une borne inférieure pour un n donné à partir d'une solution pour $n - 1$.

$$S_n \geq S_{n-1} + \lceil log_2 n \rceil$$

Prouver l'optimalité avec un préfixe

On est capable de déterminer l'optimalité d'un solution calculée à partir d'un préfixe lorsque la borne supérieure obtenue et la borne inférieure calculée avec Van Voorhis sont égales.

Résultats avec IDD (sans préfixe)

n	taille	temps
2	1	15ms
3	3	15ms
4	5	31ms
5	9	50s

Résultats obtenus avec IDD sans préfixe

Résultats avec IDD (+ préfixe)

n	taille	temps
2	1	15ms
3	3	15ms
4	5	31ms
5	9	50s
6	**12**	**30s**

Résultats étendus avec préfixe pour n=6 (optimal)

5. Construction par algorithme génétique

Mutations

Mélange

Une section de taille aléatoire sur un réseau de tri est échangée avec une autre section de taille aléatoire sur un autre réseau.

Modification

Un comparateur aléatoire est déplacé à un nouvel emplacement aléatoire.

Mutations

Insertion

Un comparateur est inséré aléatoirement dans le réseau.

Suppression

Un comparateur est supprimé aléatoirement dans le réseau.

Echange

Deux comparateurs sont échangés de place dans le réseau.

Générer la population initiale

Aléatoire

Des comparateurs sont ajoutés de manière aléatoire à un réseau jusqu'à ce qu'il soit valide. (très peu optimal mais très variés)

Méthodes de construction

Générer des candidats de base en utilisant des méthodes efficaces pour construire des réseaux valides (moins de variété)

Résultats de l'agorithme génétique

n	Résultat	Sol. Opt.
8	19	19
9	25	25
10	30	29
11	38	35
12	45	39

Meilleurs résultats obtenus lors des tests

Guillaume FREYERMUTH

2 Avril 2024