Algorithmes d'Approximation

1. Définition et Enjeux des Algorithmes d'Approximation

Un **algorithme d'approximation** est utilisé lorsqu'il est techniquement difficile, beaucoup trop long, ou mathématiquement impossible de trouver une solution exacte (analytique) à un problème donné. Son but principal est de fournir un résultat **suffisamment proche de la valeur réelle (l'optimum)**, tout en garantissant un temps de calcul et une consommation de ressources mémoire raisonnables.

En informatique, on rencontre souvent des problèmes dits **NP-difficiles ou NP-complets**. Pour ces problèmes, le temps nécessaire pour calculer la solution parfaite augmente de manière exponentielle avec la taille des données. Par exemple, calculer le trajet *absolument parfait* entre 100 villes pourrait prendre des milliards d'années à un supercalculateur. L'algorithme d'approximation permet d'obtenir un trajet "très bon" (à seulement quelques pourcents du trajet parfait) en quelques secondes seulement.

2. Heuristique vs Approximation

Il est crucial pour un développeur de faire la distinction entre une heuristique et un algorithme d'approximation, car les deux cherchent à contourner la difficulté d'un problème complexe, mais pas de la même manière :

- **L'heuristique :** C'est une méthode de résolution empirique, basée sur l'intuition ou l'expérience (le "bon sens"). Elle donne souvent de très bons résultats en pratique et très rapidement, mais *elle ne fournit aucune garantie mathématique*. Il se peut que le résultat soit extrêmement mauvais dans certains cas particuliers.
- **L'algorithme d'approximation :** Contrairement à l'heuristique, il est rigoureux et s'accompagne d'une **preuve mathématique appelée "ratio d'approximation"**. Ce ratio garantit que la solution trouvée ne sera jamais pire qu'un certain pourcentage de la solution optimale (par exemple, on garantit que le chemin trouvé ne sera jamais plus de 1,5 fois plus long que le chemin parfait, même dans le pire des scénarios).

3. Exemple classique : Le Problème du Voyageur de Commerce

Le problème du Voyageur de Commerce (TSP - Traveling Salesperson Problem) est l'exemple le plus célèbre d'algorithmique d'optimisation. Un livreur doit visiter un ensemble de villes une seule fois et revenir à son point de départ, en minimisant la distance totale parcourue.

apcpedagogie.com

Puisque trouver le chemin exact le plus court parmi des dizaines de villes est impossible dans un temps raisonnable (le nombre de combinaisons explose), on utilise des algorithmes d'approximation. Ces algorithmes vont construire rapidement un itinéraire extrêmement performant, garantissant de réduire les coûts logistiques sans s'épuiser à chercher une perfection inatteignable.

4. Exemple d'application mathématique : Le calcul de la Racine Carrée

L'approximation n'est pas utilisée que pour les graphes logistiques, elle est au cœur du calcul numérique. Par exemple, comment votre ordinateur calcule-t-il la racine carrée d'un nombre s'il ne possède dans son processeur que des circuits pour l'addition, la soustraction, la multiplication et la division ?

Il utilise un algorithme d'approximation, comme la **méthode de Héron (ou méthode de Newton)**. Cet algorithme va affiner un résultat, étape par étape, dans une boucle, jusqu'à atteindre la précision souhaitée (par exemple, s'arrêter quand on est sûr d'avoir 5 chiffres exacts après la virgule).

Voici cet algorithme d'approximation en pseudo-code :

```
Fonction RacineCarreeApprox(N : Réel, Precision : Réel) : Réel
Variables
    Estimation : Réel
Début
    // 1. On commence par une estimation de départ (ex: N divisé par 2)
    Estimation <- N / 2.0

    // 2. Tant que l'erreur entre le carré de notre estimation et N est trop grande
    Tant Que ValeurAbsolue((Estimation * Estimation) - N) > Precision Faire

        // 3. On affine l'estimation avec la formule mathématique d'approximation
        Estimation <- (Estimation + (N / Estimation)) / 2.0

    FinTantQue

    // 4. Dès que la condition de précision est atteinte, on renvoie la valeur
    Retourner Estimation
Fin
```

5. Bilan et Domaines d'application

Accepter de perdre une fraction infime de précision pour gagner des heures ou des années de temps de calcul est l'un des compromis les plus puissants en ingénierie logicielle. Les algorithmes d'approximation sont invisibles mais utilisés quotidiennement dans :

- **L'Intelligence Artificielle (Machine Learning) :** L'entraînement des modèles (comme la descente de gradient) cherche une approximation de l'erreur minimale, pas l'erreur absolue zéro.
- **Le GPS et le routage réseau :** Calcul d'itinéraires sur des cartes contenant des millions de routes.
- **La 3D et les moteurs de Jeux Vidéo :** Le calcul de la lumière et des ombres en temps réel utilise des algorithmes qui "approximent" les lois complexes de la physique pour maintenir 60 images par seconde.

Rédigé et structuré par : **Med Khalil Kribi**

Source : https://apcpedagogie.com/algorithmes-approximation · téléchargé le 07/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com