COLLÈGE DE FRANCE
1530

Algorithmes quantiques

Optimisation quantique

19-05-2021

Frédéric Magniez

Professeur invité sur la chaire Informatique et sciences numériques

En partenariat avec Inria

Année académique 2020-2021

frederic.magniez@college-de-france.fr

Présentation du cours 2/3

Partie 2 - Les bases algorithmiques

- Concepts du calcul et principales méthodes algorithmiques
- Mise en évidence de propriétés algébriques (déchiffrement)
- Optimisation et applications algorithmiques

05 mai 2021

Cours : Circuits quantiques, premiers algorithmes : portes universelles, algorithmes de Deutsch-Jozsa et Bernstein-Vazirani, supériorité des algorithmes quantiques

Séminaire : Langages graphiques pour programmer et raisonner en informatique quantique
Simon PERDRIX, *CNRS, Nancy*

12 mai 2021

Cours : Transformée de Fourier quantique : réalisation, estimation de phase, algorithmes de Simon et de Shor (recherche de période et factorisation) et généralisations récentes

Séminaire : Le problème du sous-groupe caché, Miklos SANTHA, *CNRS, Paris et CQT, Singapour*

19 mai 2021

Cours : Optimisation quantique : algorithme de Grover, estimateurs quantiques, chaînes de Markov quantiques, heuristiques quantiques

Séminaire : A Unified Framework for Quantum Walk Search, Stacey JEFFERY, *CWI, Amsterdam*

Algorithme de Grover [1995] et extensions

- Recherche/optimisation par essais successifs
- Recherche/optimisation par exploration (par ex marche aléatoire)

  T essais/étapes probabilistes → $\sqrt{T}$ essais/étapes quantiques

Heuristiques

- Parcours arborescents

  Type Branch and bound, Backtracking éventuellement stochastique

  T étapes probabilistes → $\sqrt{T}$ étapes quantiques
- Applications : SAT solver

Monte Carlo

- Utilisation d'estimateurs statistiques

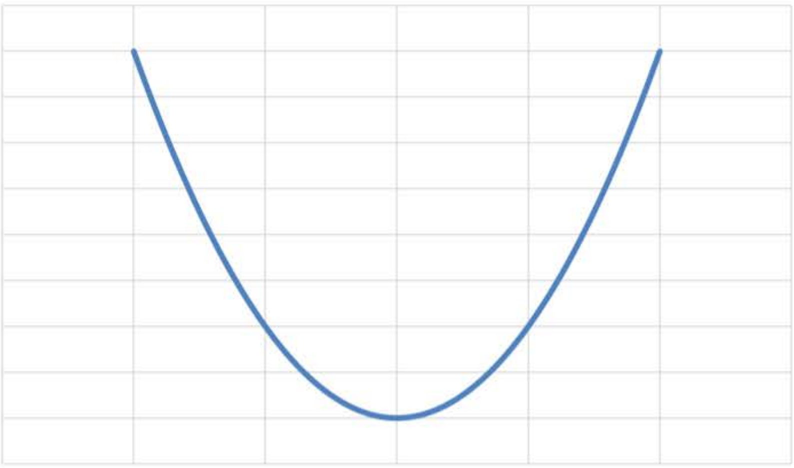
  T échantillons probabilistes → $\sqrt{T}$ échantillons quantiques

Optimisation continue (non abordée)

Calcul du gradient

- Le calcul quantique du gradient n'utilise qu'une étape au lieu d'un nombre linéaire en la dimension en classique [Jordan 2005]

Applications algorithmiques



- Optimisation convexe
- Descente de gradient
- Programmation linéaire, semi-définie

Challenge

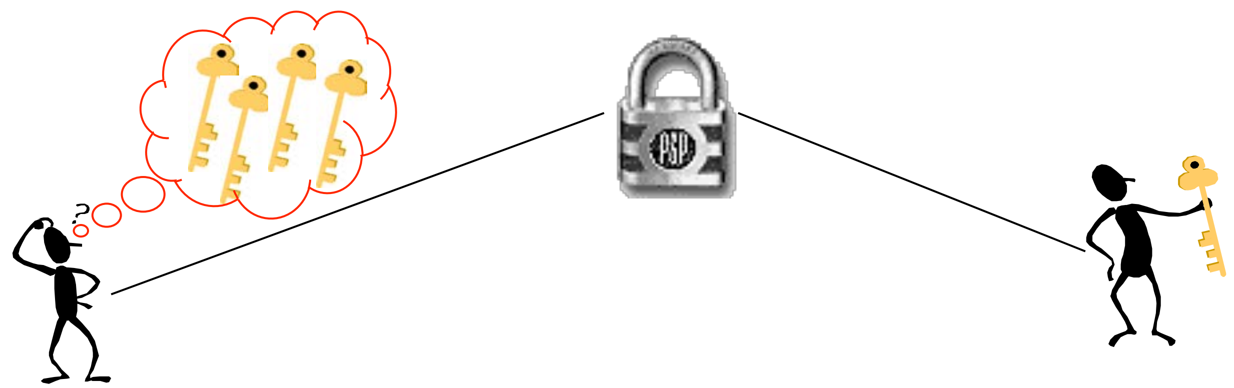
- Trouver une application industrielle compétitive avec les heuristiques actuelles

Algorithme de Grover

Problème de Grover

- Oracle : $f : [N]=\{0,1,\ldots,N-1\} \rightarrow \{0,1\}$
- Output : Trouver x tel que $f(x)=1$, s'il en existe un

  Elements solutions/marqués $M=\{x : f(x)=1\}$



- Complexité : nombre de requêtes à f (et temps/espace)

Solution

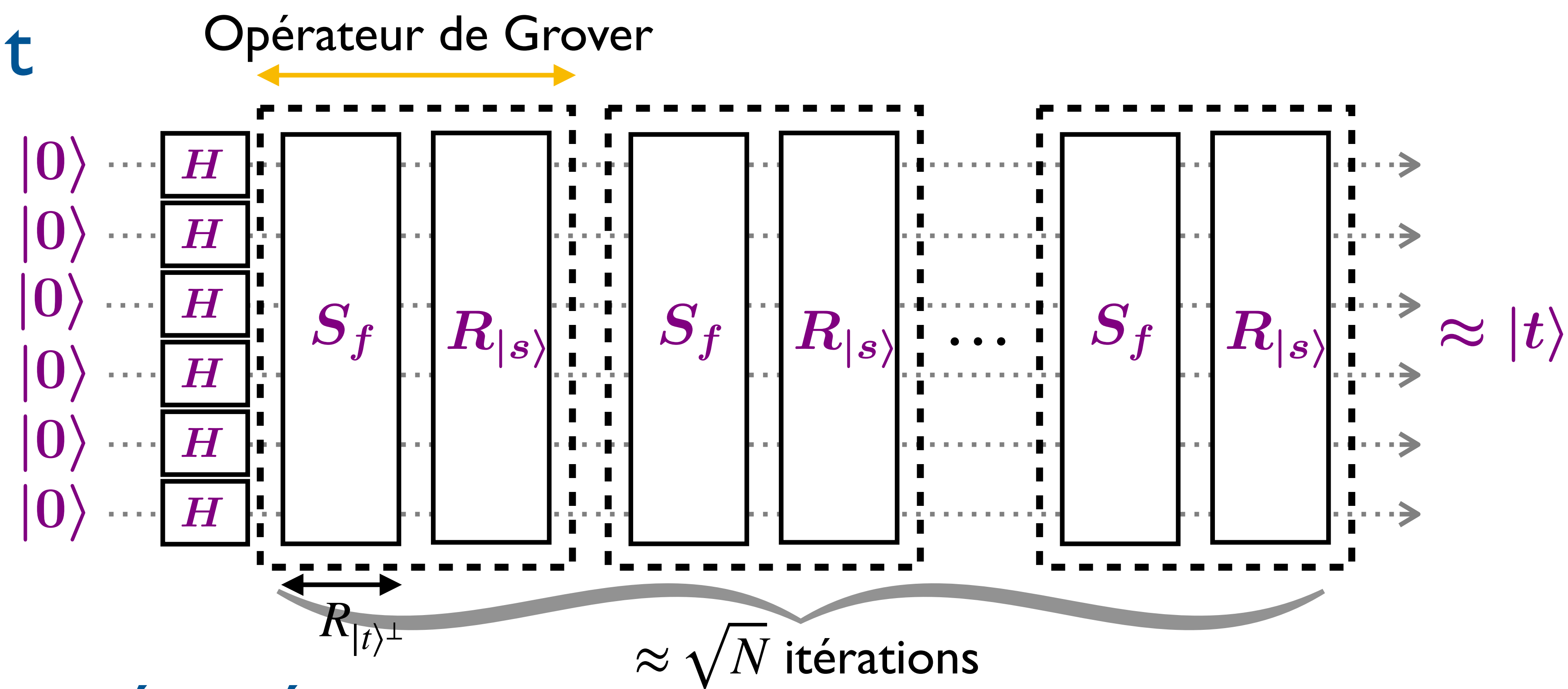
- Classique (probabiliste) : $\Omega(N)$ requêtes
- Quantique [Grover 1995] : $O(\sqrt{N})$ requêtes →CE RESULTAT EST OPTIMAL (cours 7)

  temps $O(\sqrt{N}\log N) = \tilde{O}(\sqrt{N})$ (sans compter l'oracle)

  espace $O(\log N)$ qubits (sans compter l'oracle)

Intuition - Cas $N=2^n$, unique solution t

Circuit



Analyse géométrique

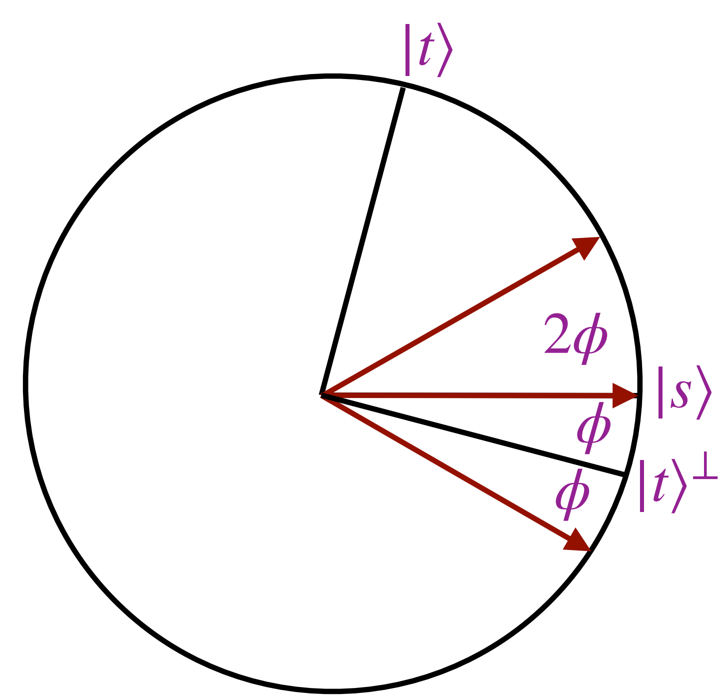
- Initialisation : $|0\dots0\rangle$
- Parallélisation : $|s\rangle$
- Appel de f : Symétrie orthogonale

$$|t\rangle \mapsto -|t\rangle$$

$$|t\rangle^\perp = \sqrt{\frac{N-1}{N}} \sum_{x:f(x)=0} |x\rangle \mapsto |t\rangle^\perp$$

- Symétrie orthogonale par rapport à $|s\rangle$
- $\pi/(4\phi) \approx \sqrt{N}$ itérations transforment $|s\rangle$ en $|t\rangle$

Opérateur de Grover : Rotation d'angle 2ϕ



$\sin\phi = \langle t | s \rangle = \sqrt{1/N}$

Symétrie par rapport à $|s\rangle$

- En utilisant les portes de Hadamard et une fonction de référence

  Deconstruction avec $H^{\otimes^n} : |s\rangle \mapsto |0\ldots0\rangle$

  Phase flip S défini par : $|0\ldots0\rangle \mapsto |0\ldots0\rangle$ et $|x \neq 0\ldots0\rangle \mapsto -|x\rangle$

  Reconstruction avec $H^{\otimes^n}$: $|0\ldots0\rangle \mapsto |s\rangle$

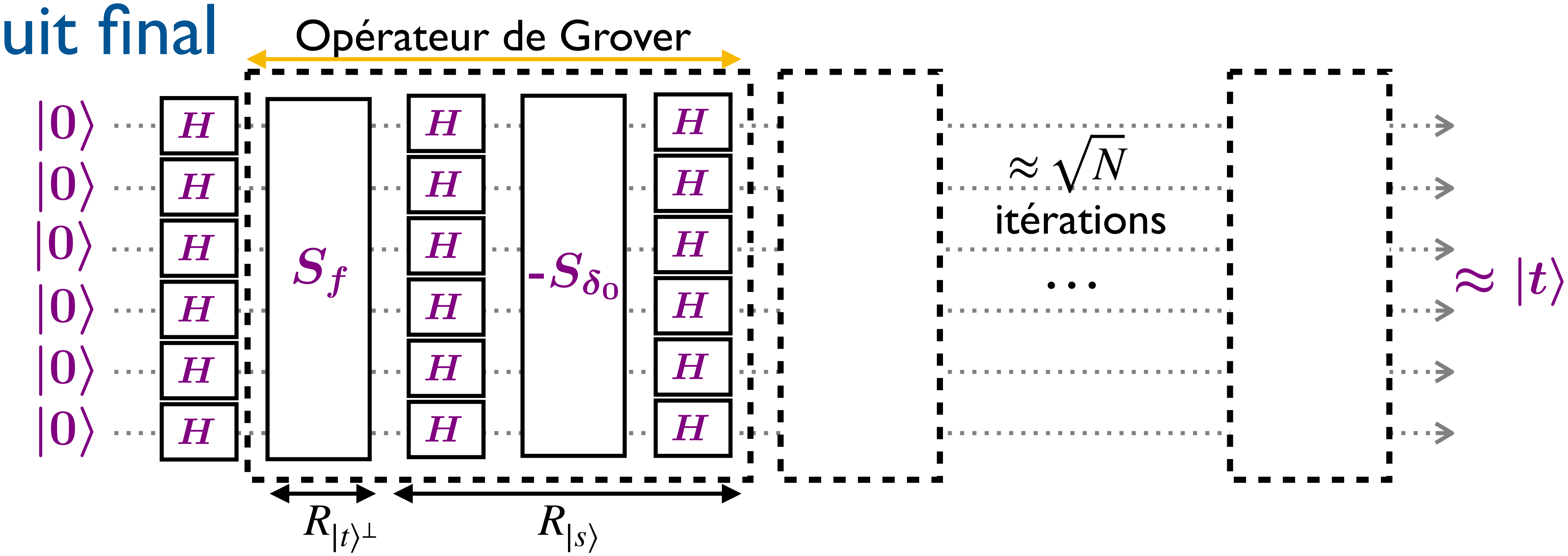
- Remarque : $-S$ correspond à l'appel de fonction

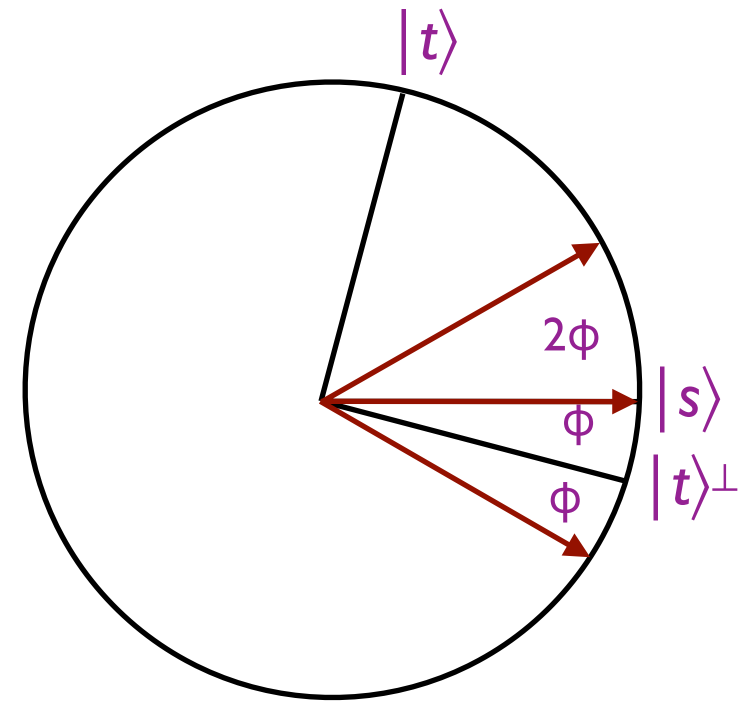
  $\delta_0 : 0\ldots0 \mapsto 1, \quad x \neq 0\ldots0 \mapsto 1$

  donc $S = -S_{\delta_0}$

- Complexité : $O(n)$ portes

Circuit final





Angle de rotation

- $|t\rangle = \frac{1}{\sqrt{k}} \sum_{x:f(x)=1} |x\rangle$
- $\sin\phi = \langle t\,|\,s\rangle = \sqrt{k/N}$
- Donc $\approx \sqrt{N/k}$ rotations suffisent
- Mais trop tourner n'est pas bon !

k inconnu, mais $k \geq k_0$ ou $k=0$

- Prendre *T* au hasard dans $[100\sqrt{N/k_0}]$
- Exécuter Grover avec *T* itérations → Probabilité de succès constante !

k inconnu

- Executer l'algorithme de Grover avec *T* itérations (initialement *T*=1)
- Si l'élément observé n'est pas solution,

  Multiplier *T* par 8/7

  Recommencer
- S'arrêter quand le nombre d'itérations dépasse $100\sqrt{N}$

  → Temps moyen = $\sqrt{N/k}$

(mieux : un nb d'itérations pris au hasard entre 1 et *T*)

Cas booléen

- Entrée : Suite de m contraintes définies par k variables 0/1 parmis n

  Exemple : $x_1=x_2$, IF $x_3=1$ THEN $x_4=x_5$, OR(x_1,NOT(x_3),x_5)=1
- Sortie : Une solution $x=(x_1,x_2,\ldots,x_n)$ qui satisfait toutes les contraintes

Réduction à Grover

- Définir $f(x)=1$ si x satisfait toutes les contraintes, et $f(x)=0$ sinon
- Nombre de candidats : $N = 2^n$
- Complexité du calcul de f :

  Circuit (quantique) de taille $O(m2^k)$ sur $O(m+2^k)$ qubits

Solution quantique

- Temps $O(\sqrt{N}((m2^k)+\log N) = O((m2^k+n)2^{n/2}) = \tilde{O}(2^{n/2})$
- Espace $O(m+2^k+\log N) = O(m+n)$ (quand k constant)

Solutions classiques

- Complexités théoriques meilleures : mais Grover peut aussi être utilisé
- Complexités pratiques encore plus rapides : des accélérations quadratiques basées sur l'études des marches quantiques

Application : Programmation dynamique

Principe

1. Décomposer un problème algorithmique en sous-problèmes,
2. Puis résoudre les sous-problèmes, des plus petits aux plus grands en stockant les résultats intermédiaires.

Version quantique [Ambainis et al 2019]

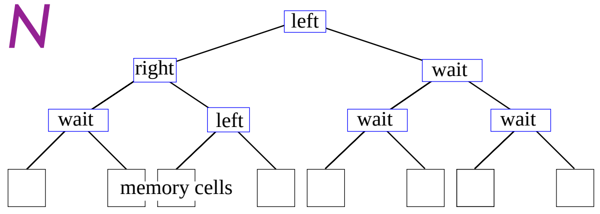
- Principe
  1. Précalculer des solutions pour une partie des sous-ensembles à l'aide de la programmation dynamique
  2. Puis utiliser la recherche de Grover sur le reste des sous-ensembles pour trouver la réponse au problème.
- Voyageur de commerce : calculer un plus court circuit qui passe une et une seule fois par n villes

  Temps quantique $\tilde{O}(1.728^n)$ vs $\tilde{O}(2^n)$ en classique

Quantum RA(Q)M [Giovannetti, Lloyd, Maccone 2008]

- Accès en superposition à une mémoire (quantique) de taille N

  en temps $O(\log N)$
- Modèle accepté en classique, mais débattu en quantique…



Amplification d’amplitudes

Cadre plus général : Amplification quantique

Problème

- Etant donné un algorithme (probabiliste/quantique) *A* qui trouve une solution (x tq $f(x)$=1) avec probabilité ≥ ε (s'il en existe une)
- Construire un algorithme qui trouve une solution (x tq $f(x)$=1)

  avec grande probabilité (en pratique ≥ 2/3)

Modélisation

- Probabiliste : *A* retourne x avec probabilité p_x telle que $\sum_{x:f(x)=1} p_x \geq \varepsilon$
- Quantique : *A* retourne $\sum_x \alpha_x |x\rangle|\psi_x\rangle$ telle que $\sum_{x:f(x)=1} |\alpha_x|^2 \geq \varepsilon$

Amplification classique

- Effectuer 1/ε × [exécutions de *A* et vérifications (requêtes à *f*)]

Amplification quantique [Brassard, Høyer, Mosca, Tapp 2000]

- Effectuer $1/\sqrt{\varepsilon}$ ×[exécutions quantiques de *A* et vérifications quantiques]

Amplification quantique

Etat de départ

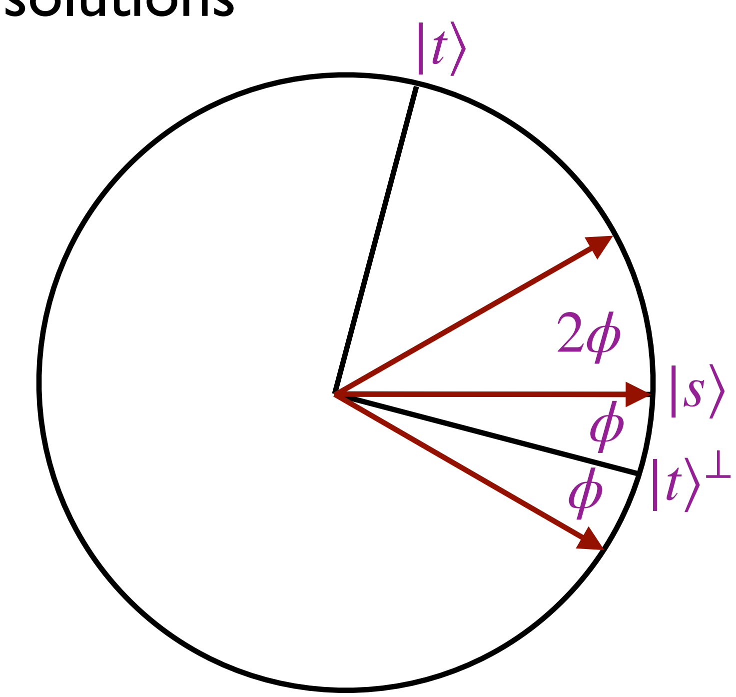
Simplification : pas d'état $|\psi_x\rangle$

- $|s\rangle = \sum_{x \in X} \alpha_x |x\rangle = A|0\ldots0\rangle$: construit par la version quantique de A

Etat cible

- $|t\rangle = \frac{1}{\sqrt{\varepsilon}} \sum_{x:f(x)=1} \alpha_x |x\rangle$: projection de $|s\rangle$ sur les solutions

Simplification :
valeur exacte



$$|t\rangle^\perp = \frac{1}{\sqrt{1-\varepsilon}} \sum_{x:f(x)=0} \alpha_x |x\rangle$$

Alternance de symétries

- $\sin\phi = \langle t | s \rangle = \sqrt{\varepsilon}$
- Opérateur de Grover : Rotation d'angle 2ϕ

  S_f : Symétrie par rapport à $|t\rangle^\perp$ puis

  Symétrie par rapport à $|s\rangle$
- Après $\Theta(1/\sqrt{\varepsilon})$ iterations

  Etat final proche de la projection de l'état initial sur les solutions

Symétrie par rapport à $|s\rangle$

- En utilisant l'algorithme *A* et une fonction de référence

  Deconstruction avec A^{-1} : $|s\rangle \mapsto |0\ldots0\rangle$

  Phase flip *S* défini par : $|0\ldots0\rangle \mapsto |0\ldots0\rangle$ et $|x \neq 0\ldots0\rangle \mapsto -|x\rangle$

  Reconstruction avec A : $|0\ldots0\rangle \mapsto |s\rangle$

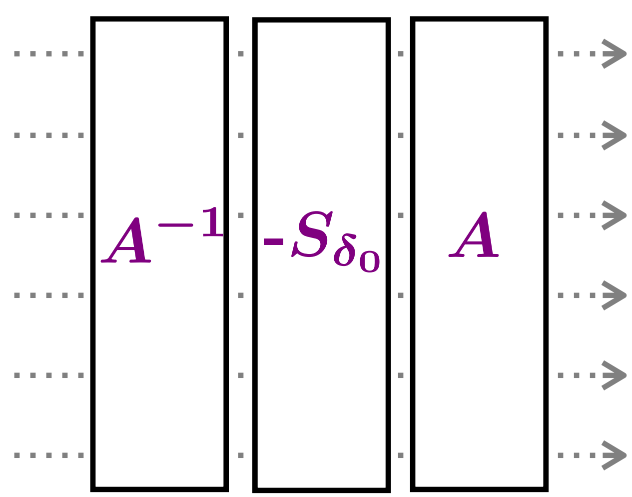
- Remarque (bis) : -*S* correspond à l'appel de fonction

  $\delta_0 : 0\ldots0 \mapsto 1, \quad x \neq 0\ldots0 \mapsto 1$

  donc $S = -S_{\delta_0}$

- Complexité : *O*(log *N*) portes en plus de A

$R_{|s\rangle} =$

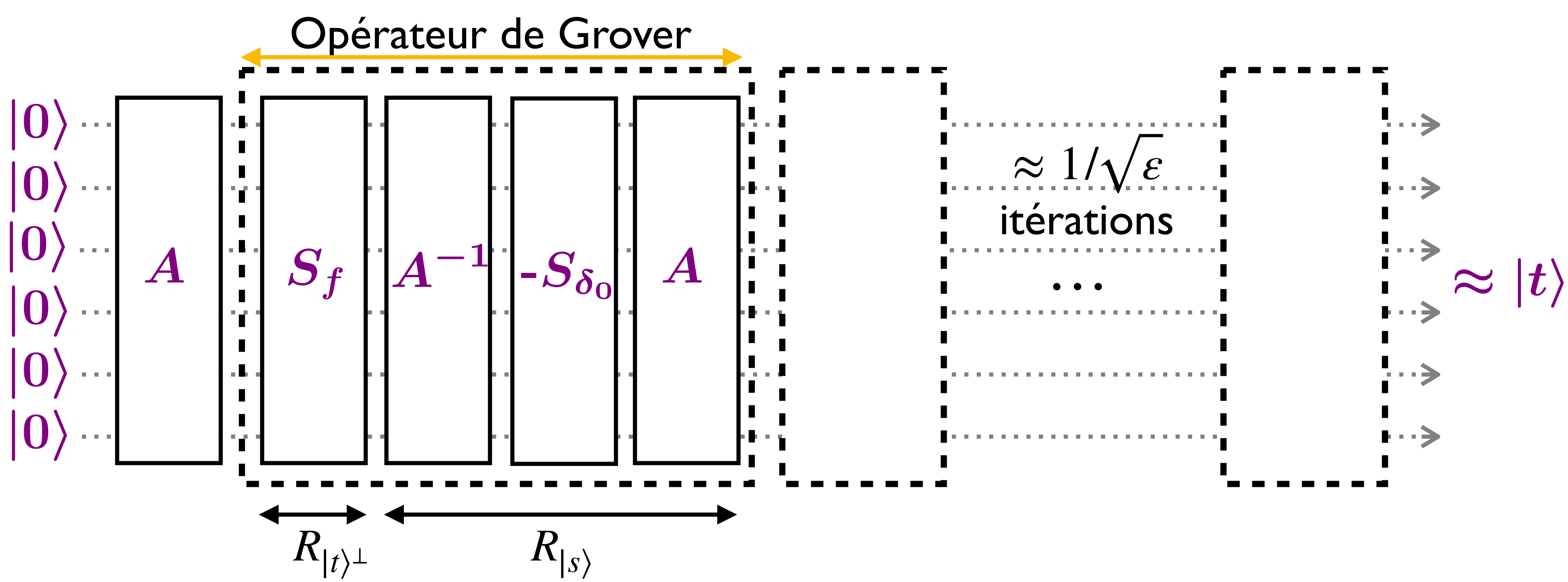


Amplification quantique [Brassard, Høyer, Mosca, Tapp 2000]

- Effectuer $1/\sqrt{\varepsilon}$ ×[exécutions quantiques de A et vérifications quantiques]
- Remarque : la solution fournie (si elle est correcte) suit la distribution initiale projetée sur les solutions

Circuit complet

- A : création d'une superposition uniforme sur $N=2^n$ éléments
- δ_0 : fonction qui faut 1 en 0…0 et 0 sinon



pris de *Hash functions: Theory, attacks, and applications* [Mironov'05]

Contexte

- Fonction $H : [N] \to [R]$ se comportant comme une fonction aléatoire
- Complexité : Nb d'évaluations de *H* (mais aussi temps, espace, nb de processeurs...)

Niveaux de résistance

- Préimage

  Pour une valeur *z*

  Trouver *x* tel que *H*(*x*)=z

  Algorithme de Grover
  $\sqrt{N}$ requêtes quantiques

- Seconde préimage

  Pour une entrée *x*

  Trouver *y*≠*x* tel que *H*(*x*)=*H*(*y*)

  Algorithme de Grover
  $\sqrt{N}$ requêtes quantiques

- Collision

  Trouver *x* et *y*≠*x* tel que *H*(*x*)=*H*(*y*)

  Algorithme de Grover
  $\sqrt{N}$ à N requêtes quantiques

Attaques standard

- Recherche exhaustive
- Paradoxe des anniversaires (et ses variantes itératives)

Recherche de collision

- Entrée : une fonction $H : [N] \to [N]$ aléatoire
- Sortie : une paire (x,y) telle que $H(x)=H(y)$ et $x \neq y$
- Hypothèse H aléatoire garantit que

  pour chaque la plupart des x il existe $y \neq x$ tq $H(x)=H(y)$

Solution quantique [Brassard, Høyer, Tapp 1998]

- Précalcul

  Evaluer et trier $H(0), H(1), \ldots, H(k\text{-}1)$

  $O(k)$ requêtes, temps $\tilde{O}(k)$
- Algorithme

  Renvoyer un entier i au hasard dans $\{k, k+1, \ldots, N\text{-}1\}$
- Vérification : 1 requête
- Probabilité de succès : $\varepsilon \geq k/N$
- Amplification : $O(k + \sqrt{N/k})$ requêtes, temps $\tilde{O}(k + \sqrt{N/k})$ (si QRAM)
- Optimisation ($k = \sqrt[3]{N}$) : $O(\sqrt[3]{N})$ requêtes, temps $\tilde{O}(\sqrt[3]{N})$ (si QRAM)

CE RESULTAT EST OPTIMAL (cours 7)

Recherche de collision

- Entrée : une fonction $H : [N] \to [N^2]$ aléatoire
- Sortie : une paire (x,y) telle que $H(x)=H(y)$ et $x \neq y$
- Hypothèse H aléatoire garantit qu'il existe sans doute une collision…

Solution quantique [Buhrman et al 2001]

- Algorithme

  Choisir au hasard $a_1, a_2, \ldots, a_k$, évaluer et trier $H(a_1), H(a_2), \ldots, H(a_k)$

  $O(k)$ requêtes

  Chercher une collision avec l'un des $a_1, a_2, \ldots, a_k$

  $O(\sqrt{N})$ requêtes
- Vérification : 1 requête
- Probabilité de succès : $\varepsilon \geq k/N$
- Amplification : $O(\sqrt{N/k})$ itérations

  Au total $O(\sqrt{N/k} \times (k + \sqrt{N}))$ requêtes
- Optimisation ($k = \sqrt{N}$) : $O(N^{3/4})$ requêtes, temps $\tilde{O}(N^{3/4})$ (si QRAQM)

CE RESULTAT N'EST OPTIMAL…

Extensions

Recherche de minimum

- Entrée : une fonction $f : [N] \to [R]$
- Sortie : x tel que $f(x)$ est minimale

Solution itérative [Dürr, Høyer 1996]

- Choisir au hasard x dans $[N]$
- Chercher avec l'algorithme de Grover y tel que $f(y)<f(x)$
- Si y est trouvé : Remplacer x par y et recommencer l'étape précédente
- Sinon renvoyer x

Analyse

- Chaque nouveau x est uniformément au hasard parmi les éléments ayant une image plus petite par f
- En moyenne, au début $N/2$ nouveaux candidats, puis $N/4$, $N/8$…
- Complexité en moyenne

  $\sqrt{N/2} + \sqrt{N/4} + \sqrt{N/8} + \ldots + 1 = O(\sqrt{N})$ évaluations de f

  temps $\tilde{O}(\sqrt{N})$

Estimation d'amplitude et comptage

- La rotation possède un angle 2ϕ directement relié à la probabilité que l'Algorithme initial renvoie une solution

  $\sin \phi = \langle t | s \rangle = (\Pr[\text{Algorithme initial } A \text{ renvoie une solution}])^{1/2}$

- Cas particulier, Algorithme A renvoie un élément au hasard parmi N

  $$\sin \phi = \sqrt{\frac{K}{N}}$$

Solution quantique

- Approximation de $\langle t | s \rangle$ avec erreur relative ε en $O\left(\frac{1}{\varepsilon \langle t | s \rangle}\right)$ échantillons et vérifications
- De même K avec $O\left(\frac{1}{\varepsilon}\sqrt{\frac{N}{K}}\right)$ échantillons et vérifications

Deux approches

- Analyse spectrale et Estimation de phase (cours 4) [Brassard, Høyer, Mosca, Tapp 2000]
- De proche en proche [Aaronson, Rall 2020]

Un exemple

- Oracle qui produit un échantillon d'une variable aléatoire $X = (x, p_x)$
  disponible aussi en superposition quantique : $\sum_x \sqrt{p_x} |x\rangle |\psi_x\rangle$
- Output : la moyenne $\mu = \sum_x p_x x$ de X avec erreur relative ε
- Question : Combien d'échantillons sont-ils nécessaires ?

Cas X=0/1

- Classique : $1/(\varepsilon^2 \mu)$
- Quantique : $1/(\varepsilon\sqrt{\mu})$ (généralisation du comptage)

Cas général

- Inégalité de Chebyshev classique : $(\sigma/(\varepsilon\mu))^2$ où σ^2 est la variance de X
- Inégalité de Chebyshev quantique : $\sigma/(\varepsilon\mu)$
  [Montanaro 2015][Hamoudi M 2020]

Nombreuses applications

- Accélération d'algorithmes classiques
- Calcul numérique : dont le calcul de volume d'enveloppe convexes

Marches quantiques

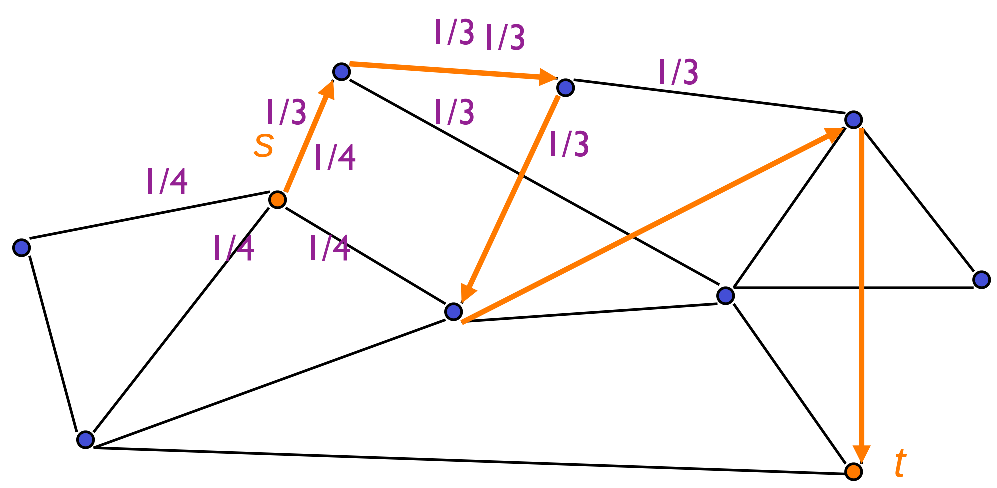
Définition

- $G = (V,E)$ un graphe (non orienté) de sommets V et d'arêtes E
- Une marche aléatoire est un déplacement aléatoire

  sur les sommets V de G

  en suivant les arêtes E de G

  tel que $\Pr[\, u \rightarrow v \,] = 1/\deg(u)$



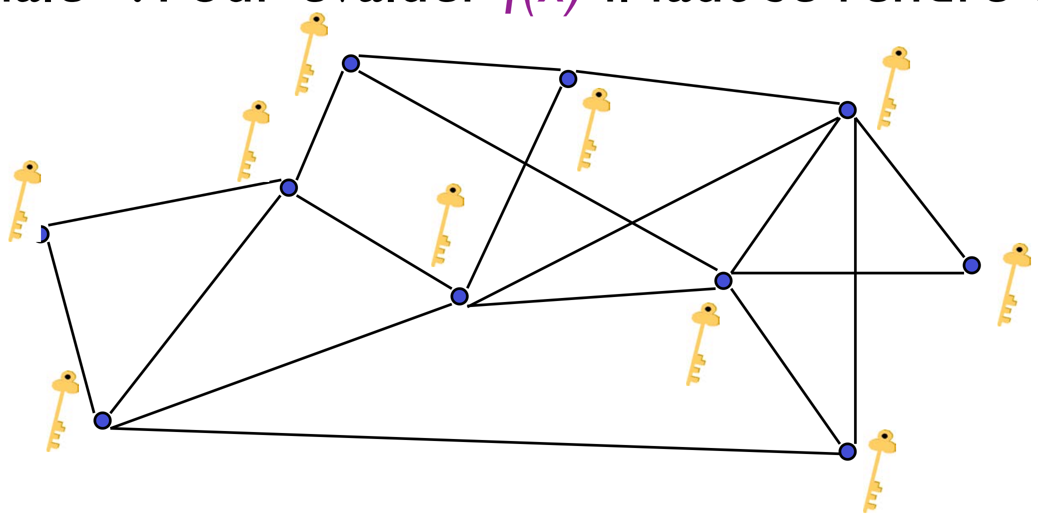
Application typique

- Rechercher un chemin de s à t, mélanger
- Modéliser, analyser
- Concevoir des algorithmes

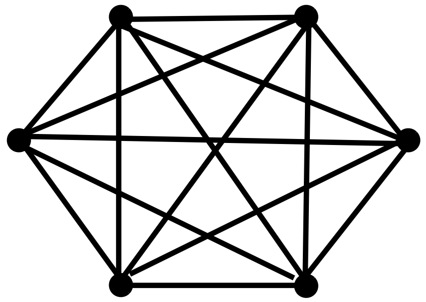
Problème de Grover

- Oracle : $f : V \rightarrow \{0,1\}$
- Output : Trouver x tel que $f(x)=1$, s'il en existe un

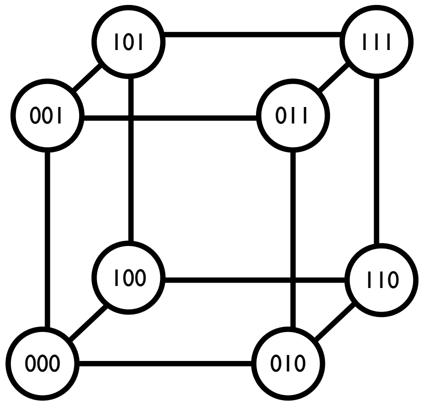
  Elements marqués/solutions $M=\{x : f(x)=1\}$
- Contrainte "spatiale" : Pour évaluer $f(x)$ il faut se rendre en x en suivant les arêtes de G

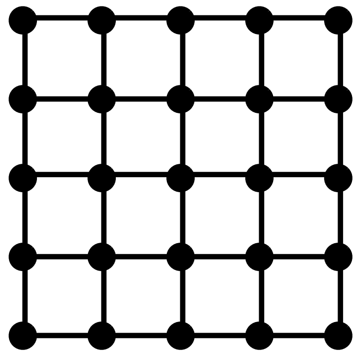


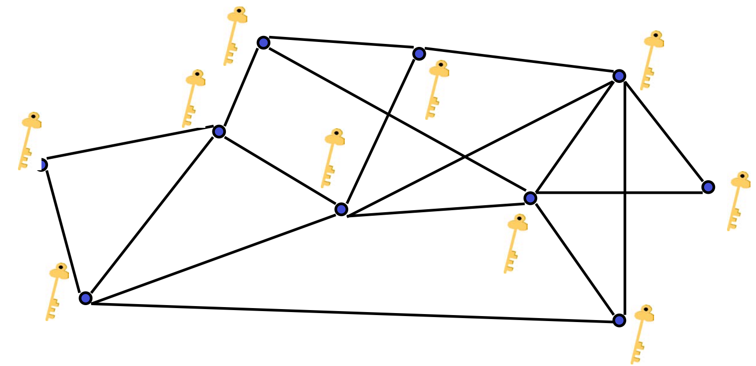
Exemples avec accélération quadratique



- Graphe complet [Grover'95]
- Hypercube [Shenvi, Kempe, Whaley'03]
- Grille 2D [Ambainis, Kempe, Rivosh'05] [Tulsi'08]







Problème de Grover

- Oracle : $f : V \rightarrow \{0,1\}$
- Output : Trouver x tel que $f(x)=1$, s'il en existe un

  Elements marqués/solutions $M=\{x : f(x)=1\}$
- Contrainte "spatiale" : Pour évaluer $f(x)$ il faut se rendre en x en suivant les arêtes de G

Complexités des opérations élémentaires

- Setup : Préparation de la distribution/superposition initiale
- Checking : Vérification (requête à f)
- Update : Déplacement sur G

Paramètres importants

- Propriétés spectrale de la matrice P : $P_{uv} = \Pr[\, u \longrightarrow v \,]$ $(= 1/\deg(u))$

  Distribution stationnaire π (valeur propre 1) : uniforme si degré constant

  L'écart entre 1 et la deuxième valeur propre est au moins δ
- ε = probabilité qu'un élément pris selon la distribution π soit solution

ε = probabilité qu'un élt au hasard soit marqué

δ = écart spectral de la marche aléatoire sur G

Recherche par marche aléatoire

Mélange 1. Partir d'un sommet au hasard

Amplification 2. Répéter $1/\varepsilon$ fois

a. Vérifier si le sommet courant est solution

Mélange b. Effectuer $1/\delta$ déplacements aléatoires sur G

3. Si aucune solution n'a été trouvée, renvoyer "aucune solution"

Théorème

- L'algorithme "Recherche par marche aléatoire" trouve une solution, si elle existe, avec complexité

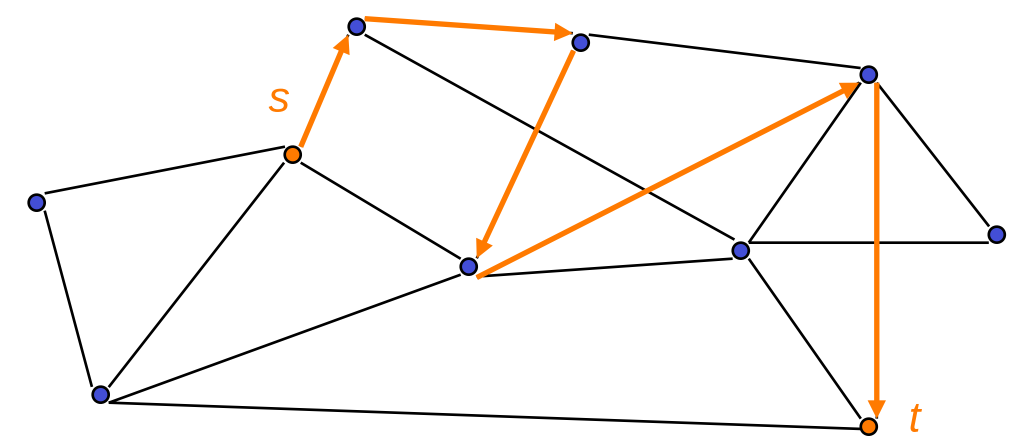
$$\text{Setup} + 1/\varepsilon \times (\text{Checking} + 1/\delta \times \text{Update})$$

Théorème

- L'analogue quantique dû à [Ambainis'04][MNayakRolandSantha'07] **trouve** une solution, si elle existe, avec complexité

$$\text{Setup} + 1/\sqrt{\varepsilon} \times (\text{Checking} + 1/\sqrt{\delta} \times \text{Update})$$

- Preuve : Amplification d'amplitude avec $R_{|s\rangle}$ implémentée à l'aide de $1/\sqrt{\delta}$ déplacements quantiques selon P

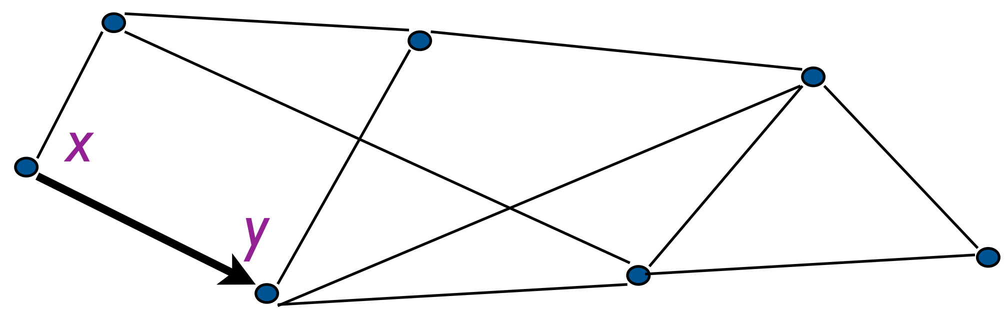


Etudes

- Quantum random walks
  [Aharonov, Davidovich, Zagury 1993]
- Quantum walks on graphs
  [Aharonov, Ambainis, Kempe, Vazirani 2001]
- Exponential algorithmic speedup by quantum walk
  [Childs, Cleve, Deotto, Farhi 2003]
- Quantum speed-up of Markov chain based algorithms, [Szegedy 2004]
- Universal computation by quantum walk, [Childs 2008]

Applications

- Technique algorithmique majeure
  Utilisée pour simuler les systèmes quantiques
- Au cœur de la complexité quantique
  Toute formule logique *read-once* à N variables s'évalue quantiquement en $\sqrt{N}$ requêtes (optimal)



Marche quantique *W* sur *G*

- Depuis l'état $|x\rangle|y\rangle$

  1. Symétrie R : Effectuer sur $|y\rangle$ la symétrie $R_{|s_x\rangle}$ par rapport à la superposition uniforme $|s_x\rangle$ des voisins de $|x\rangle$

  $$R = \sum_{x\in V} |x\rangle\langle x| \otimes R_{|s_x\rangle} \qquad |s_x\rangle = \frac{1}{\sqrt{d(x)}} \sum_{y:(xy)\in G} |y\rangle$$

  2. *SWAP* : Echanger les deux registres (se déplacer)

Propriétés de *W*

- Superposition stationnaire (valeur propre 1) déduite de la distribution stationnaire π de *P*

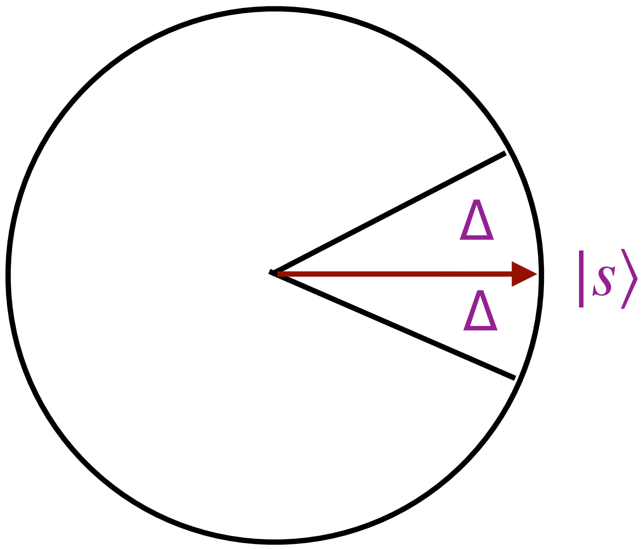
  $$|s\rangle = \sum_x \sqrt{\pi_x}|x\rangle|s_x\rangle$$

- L'écart de phase entre $1 = e^{i0}$ et celle de la plus proche valeur propre est

  $\Delta(W) \approx \sqrt{\delta(P)}$ [Szegedy 2004]

Symétrie par rapport à $|s\rangle$ l'aide de W

- Propriétés de la marche quantique W



W opérateur unitaire

$|s\rangle$ vecteur stationnaire isolé

$\Delta \approx \sqrt{\delta}$ écart de phase [Szegedy 2004]

- Algorithme

  Utiliser l'estimation de phase (cours 3)

  sur W sur l'état courant et precision $\Delta/2$

  Effectuer un Phase flip si la phase estimée est à distance $\geq \Delta$ de 0

  Inverser l'estimation de phase

- Rappel : Estimation de phase avec précision $1/T$ nécessite T itérations
- Coût total : $1/\Delta$ × Update = $1/\sqrt{\delta}$ × Update

Recherche de collision

- Entrée : une fonction $H : [N] \to [N^2]$ aléatoire
- Sortie : une paire (x,y) telle que $H(x)=H(y)$ et $x \neq y$
- Hypothèse H aléatoire garantit qu'il existe sans doute une collision…

Rappel de la solution de [Buhrman et al 2001]

- Algorithme

  Choisir au hasard $a_1, a_2, \ldots, a_k$, évaluer et trier $H(a_1), H(a_2), \ldots, H(a_k)$

  $O(k)$ requêtes

  Chercher une collision avec l'un des $a_1, a_2, \ldots, a_k$

  $O(\sqrt{N})$ requêtes

- Probabilité de succès : $\varepsilon \geq k/N$ → Amplification : $O(\sqrt{N/k})$ itérations

  Au total $O(\sqrt{N/k} \times (k + \sqrt{N}))$ requêtes

Idée

- Il y a déjà une proba $\approx (k/n)^2$ d'avoir des collisions parmi $a_1, a_2, \ldots, a_k$
- Supprimons la recherche de Grover, et amplifions la 1e étape en modifiant 1 à 1 les éléments choisis…

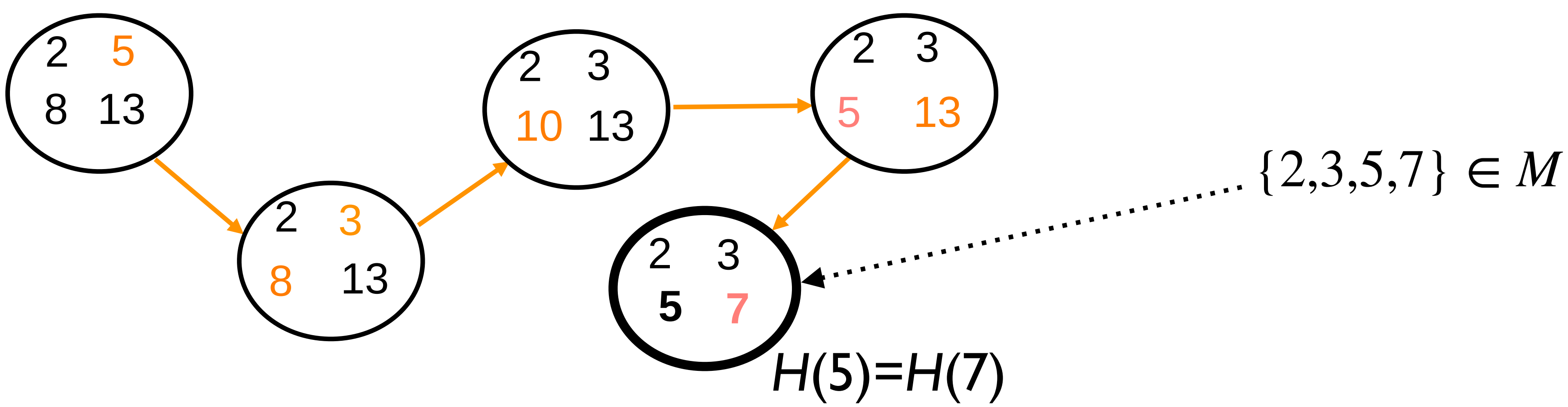
Solution avec marches quantiques [Ambainis'04]

Recherche d'un sous-ensemble avec collision

- Graphe de Johnson $J(N,k)$

  Nœuds : $\{S\}$ sous-ensemble S de taille k de $[N]$

  Arêtes : $\{S, T\}$ est une arête ssi S, T diffèrent exactement de 2 éléments
- Marcher sur $J(N,k)$ en maintenant les valeurs de H sur S
- Solutions $M = \{S$ avec une collision$\}$



- Ecart spectral: $\delta \approx 1/k$
- Probabilité de succès : $\varepsilon = \Pr[\, S$ a une collision $] \geq (k/N)^2$
- Complexité: $\underset{\text{setup}}{k} + (N^2/k^2)^{1/2}\,(\underset{\text{checking}}{0} + \underset{\text{update}}{k^{1/2}} \times 1)$ $\rightarrow$ $n^{2/3}$ évaluations de f

  Temps/Epace: $n^{2/3}$ polylog n

  (si QRAQM)

CE RESULTAT EST OPTIMAL (cours 7)

Références en lien avec le cours

Articles

- Quantum query complexity of some graph problems [Dürr, Heiligman, Høyer, Mhalla 2004]
- Quantum algorithm for tree size estimation, with applications to backtracking and 2-player games [Ambainis, Kokainis 2017]
- Quantum speedup of branch-and-bound algorithms [Montanaro 2020]
- Quantum Speedups for Exponential-Time Dynamic Programming Algorithms [Ambainis et al 2019]
- Quantum Speedup for Graph Sparsification, Cut Approximation and Laplacian Solving [Apers, de Wolf 2020]

Thèses

- Frameworks for Quantum Algorithms [Jeffery 2014]
- Classical and Quantum Cryptanalysis for Euclidean Lattices and Subset Sums [Shen 2021]
- Quantum Algorithms for the Monte Carlo Method [Hamoudi 2021]

Séminaire du cours !

- A Unified Framework for Quantum Walk Search

  avec Stacey Jeffery, CWI, Amsterdam