

Partitionnement de circuits

Safia Kedad-Sidhoum
LIP6 - Université Paris 6
safia.kedad-sidhoum@lip6.fr

DEA-ASIME, Mars 2004

Introduction

Objet

Définition

Multiway partition

Définition

Instance

Configuration

Solution - Optimale

Complexité

Applications

Bi-partitionnement

Définition

Complexité

Hypergraphes

Heuristiques d'amélioration itérative

Objectifs

Heuristique de Kernighan-Lin
Heuristique de Fiduccia-Mattheyses
Conclusion

Algorithme de recuit simulé

Principes généraux
Problème de bi-partitionnement

Notion de partitionnement

- Diviser un circuit en petites parties.

Notion de partitionnement

- ▶ Diviser un circuit en petites parties.
- ▶ Aspect important des problèmes de Layout.

Notion de partitionnement

- ▶ Diviser un circuit en petites parties.
- ▶ Aspect important des problèmes de Layout.
- ▶ Partitionnement : opération sur les graphes et hypergraphes.

Notion de partitionnement

- ▶ Diviser un circuit en petites parties.
- ▶ Aspect important des problèmes de Layout.
- ▶ Partitionnement : opération sur les graphes et hypergraphes.
- ▶ Peut apparaître pour des problèmes de routage.

Définition du partitionnement

- Diviser un graphe ou un hypergraphe en plusieurs composants déconnectés.

Définition du partitionnement

- ▶ Diviser un graphe ou un hypergraphe en plusieurs composants déconnectés.
- ▶ Graphe ou Hypergraphe $G = (V, E) \equiv \text{circuit}$.

Définition du partitionnement

- ▶ Diviser un graphe ou un hypergraphe en plusieurs composants déconnectés.
- ▶ Graphe ou Hypergraphe $G = (V, E) \equiv$ circuit.
- ▶ **V- Ensemble de sommets** $\equiv$ éléments du circuit.

Définition du partitionnement

- ▶ Diviser un graphe ou un hypergraphe en plusieurs composants déconnectés.
- ▶ Graphe ou Hypergraphe $G = (V, E) \equiv$ circuit.
- ▶ V - Ensemble de sommets $\equiv$ éléments du circuit.
- ▶ E - Ensemble des arêtes ou hyper-arêtes $\equiv$ fils du circuit.

Définition du partitionnement

- ▶ Diviser un graphe ou un hypergraphe en plusieurs composants déconnectés.
- ▶ Graphe ou Hypergraphe $G = (V, E) \equiv$ circuit.
- ▶ V - Ensemble de sommets $\equiv$ éléments du circuit.
- ▶ E - Ensemble des arêtes ou hyper-arêtes $\equiv$ fils du circuit.
- ▶ Graphe de routage : E - canaux et V Switch boxes.

Définition du partitionnement

- ▶ Diviser un graphe ou un hypergraphe en plusieurs composants déconnectés.
- ▶ Graphe ou Hypergraphe $G = (V, E) \equiv$ circuit.
- ▶ V - Ensemble de sommets $\equiv$ éléments du circuit.
- ▶ E - Ensemble des arêtes ou hyper-arêtes $\equiv$ fils du circuit.
- ▶ Graphe de routage : E - canaux et V Switch boxes.
- ▶ Un partitionnement peut être obtenu en supprimant les arêtes (ou hyper-arêtes) ou les sommets (et les arêtes incidentes) (problème de séparation).

Définition du partitionnement

- ▶ Diviser un graphe ou un hypergraphe en plusieurs composants déconnectés.
- ▶ Graphe ou Hypergraphe $G = (V, E) \equiv$ circuit.
- ▶ **V- Ensemble de sommets** $\equiv$ éléments du circuit.
- ▶ **E- Ensemble des arêtes** ou hyper-arêtes $\equiv$ fils du circuit.
- ▶ Graphe de routage : E - canaux et V Switch boxes.
- ▶ Un partitionnement peut être obtenu en supprimant les arêtes (ou hyper-arêtes) ou les sommets (et les arêtes incidentes) (problème de séparation).
- ▶ Le coût d'une partition est la somme pondérée des éléments supprimés du graphe.

Définition du partitionnement

- ▶ Diviser un graphe ou un hypergraphe en plusieurs composants déconnectés.
- ▶ Graphe ou Hypergraphe $G = (V, E) \equiv$ circuit.
- ▶ **V- Ensemble de sommets** $\equiv$ éléments du circuit.
- ▶ **E- Ensemble des arêtes** ou hyper-arêtes $\equiv$ fils du circuit.
- ▶ Graphe de routage : E - canaux et V Switch boxes.
- ▶ Un partitionnement peut être obtenu en supprimant les arêtes (ou hyper-arêtes) ou les sommets (et les arêtes incidentes) (problème de séparation).
- ▶ Le coût d'une partition est la somme pondérée des éléments supprimés du graphe.
- ▶ Contraintes additionnelles : taille des composants...

Problème de la "multiway partition"

Partitionnement d'un hypergraphe en un nombre de parties déconnectées (composants) par suppression d'hyper-arêtes.

Instance du problème

- ▶ hypergraphe $G = (V, E)$ avec $n = |V|$,

Instance du problème

- ▶ hypergraphe $G = (V, E)$ avec $n = |V|$,
- ▶ pondéré, $\omega : V \rightarrow \mathbb{N}$,

Instance du problème

- ▶ hypergraphe $G = (V, E)$ avec $n = |V|$,
- ▶ pondéré, $\omega : V \rightarrow \mathbb{N}$,
- ▶ valué, $c : E \rightarrow \mathbb{N}$.

Instance du problème

- ▶ hypergraphe $G = (V, E)$ avec $n = |V|$,
- ▶ pondéré, $\omega : V \rightarrow \mathbb{N}$,
- ▶ valué, $c : E \rightarrow \mathbb{N}$.
- ▶ $r \in \mathbb{N}$ parties.

Instance du problème

- ▶ hypergraphe $G = (V, E)$ avec $n = |V|$,
- ▶ pondéré, $\omega : V \rightarrow \mathbb{N}$,
- ▶ valué, $c : E \rightarrow \mathbb{N}$.
- ▶ $r \in \mathbb{N}$ parties.
- ▶ Taille min $b(i)$ avec $i = 1, \dots, r$.

Instance du problème

- ▶ hypergraphe $G = (V, E)$ avec $n = |V|$,
- ▶ pondéré, $\omega : V \rightarrow \mathbb{N}$,
- ▶ valué, $c : E \rightarrow \mathbb{N}$.
- ▶ $r \in \mathbb{N}$ parties.
- ▶ Taille min $b(i)$ avec $i = 1, \dots, r$.
- ▶ Taille max $B(i)$ avec $i = 1, \dots, r$.

Configuration du partitionnement

- r -way partition $\Pi = (V_1, \dots, V_r)$ de V ,

Configuration du partitionnement

- ▶ r -way partition $\Pi = (V_1, \dots, V_r)$ de V ,
- ▶ $V_i \subset V$,

Configuration du partitionnement

- ▶ r -way partition $\Pi = (V_1, \dots, V_r)$ de V ,
- ▶ $V_i \subset V$,
- ▶ $V_i \neq \emptyset$,

Configuration du partitionnement

- ▶ r -way partition $\Pi = (V_1, \dots, V_r)$ de V ,
- ▶ $V_i \subset V$,
- ▶ $V_i \neq \emptyset$,
- ▶ $V_i \cap V_j = \emptyset$ pour tout $i, j \in \{1, \dots, r\}$,

Configuration du partitionnement

- ▶ r -way partition $\Pi = (V_1, \dots, V_r)$ de V ,
- ▶ $V_i \subset V$,
- ▶ $V_i \neq \emptyset$,
- ▶ $V_i \cap V_j = \emptyset$ pour tout $i, j \in \{1, \dots, r\}$,
- ▶ $\bigcup_{i \in \{1, \dots, r\}} V_i = V$.

Solution et critère d'optimisation

- **Solution:** r -way partition $\Pi = (V_1, \dots, V_r)$ de V tel que $b(i) \leq \omega(V_i) \leq B(i)$ pour tout $i \in \{1, \dots, r\}$.

Solution et critère d'optimisation

- **Solution**: r -way partition $\Pi = (V_1, \dots, V_r)$ de V tel que $b(i) \leq \omega(V_i) \leq B(i)$ pour tout $i \in \{1, \dots, r\}$.
- **Optimisation**: Il s'agit de minimiser

$$c(\Pi) = \frac{1}{2} \sum_{i \in \{1, \dots, r\}} \sum_{e \in E_{\text{ext}, i}} c(e)$$

avec $E_{\text{ext}, i}$ ensemble des arêtes incidentes extérieures à V_i .

Solution et critère d'optimisation

- ▶ **Solution**: r -way partition $\Pi = (V_1, \dots, V_r)$ de V tel que $b(i) \leq \omega(V_i) \leq B(i)$ pour tout $i \in \{1, \dots, r\}$.
- ▶ **Optimisation**: Il s'agit de minimiser

$$c(\Pi) = \frac{1}{2} \sum_{i \in \{1, \dots, r\}} \sum_{e \in E_{\text{ext}, i}} c(e)$$

avec $E_{\text{ext}, i}$ ensemble des arêtes incidentes extérieures à V_i .

- ▶ Les hyper-arêtes connectant différentes parties dans Π contribuent à la coupe.

Solution et critère d'optimisation

- ▶ **Solution**: r -way partition $\Pi = (V_1, \dots, V_r)$ de V tel que $b(i) \leq \omega(V_i) \leq B(i)$ pour tout $i \in \{1, \dots, r\}$.
- ▶ **Optimisation**: Il s'agit de minimiser

$$c(\Pi) = \frac{1}{2} \sum_{i \in \{1, \dots, r\}} \sum_{e \in E_{ext,i}} c(e)$$

avec $E_{ext,i}$ ensemble des arêtes incidentes extérieures à V_i .

- ▶ Les hyper-arêtes connectant différentes parties dans Π contribuent à la coupe.
- ▶ Le coût $c(\Pi)$ est appelé **cutsizes**. Il représente la somme pondérée des connections externes des composants.

Complexité du problème

- Problème défini est NP-difficile.

Complexité du problème

- ▶ Problème défini est NP-difficile.
- ▶ Le problème de décision associé avec $\omega(i) = 1$, $c(e) = 1$, $b(i) = 1$ et $B(i) = n$ est NP-complet au sens fort.

Complexité du problème

- ▶ Problème défini est NP-difficile.
- ▶ Le problème de décision associé avec $\omega(i) = 1$, $c(e) = 1$, $b(i) = 1$ et $B(i) = n$ est NP-complet au sens fort.
- ▶ Cas planaire avec restrictions peut être résolu en temps polynomial.

Applications types

- Méthodes de clustering.

Applications types

- ▶ Méthodes de clustering.
- ▶ Top-down partitionnement.

Problème du Bi-partitionnement

- Multiway partition avec $r = 2$.

Problème du Bi-partitionnement

- ▶ Multiway partition avec $r = 2$.
- ▶ L'ensemble des arêtes (hyper-arêtes) participant au partitionnement est appelée une **coupe**.

Problème du Bi-partitionnement

- ▶ Multiway partition avec $r = 2$.
- ▶ L'ensemble des arêtes (hyper-arêtes) participant au partitionnement est appelée une **coupe**.
- ▶ Cutsizes : $\sum_{e \in \text{Coupe}} c(e)$.

Problème du Bi-partitionnement

- ▶ Multiway partition avec $r = 2$.
- ▶ L'ensemble des arêtes (hyper-arêtes) participant au partitionnement est appelée une **coupe**.
- ▶ Cutsizes : $\sum_{e \in \text{Coupe}} c(e)$.
- ▶ Bornes : $b(1) = b(2) = 1$ et $B(1) = B(2) = \lfloor \alpha \omega(V) \rfloor$ avec $\frac{1}{2} \leq \alpha < 1$. α est appelé **facteur d'équilibre**.

Problème du Bi-partitionnement

- ▶ Multiway partition avec $r = 2$.
- ▶ L'ensemble des arêtes (hyper-arêtes) participant au partitionnement est appelée une **coupe**.
- ▶ Cutsizes : $\sum_{e \in \text{Coupe}} c(e)$.
- ▶ Bornes : $b(1) = b(2) = 1$ et $B(1) = B(2) = \lfloor \alpha \omega(V) \rfloor$ avec $\frac{1}{2} \leq \alpha < 1$. α est appelé **facteur d'équilibre**.
- ▶ Partitions α -équilibrées.

Problème du Bi-partitionnement

- ▶ Multiway partition avec $r = 2$.
- ▶ L'ensemble des arêtes (hyper-arêtes) participant au partitionnement est appelée une **coupe**.
- ▶ Cutsizes : $\sum_{e \in \text{Coupe}} c(e)$.
- ▶ Bornes : $b(1) = b(2) = 1$ et $B(1) = B(2) = \lfloor \alpha \omega(V) \rfloor$ avec $\frac{1}{2} \leq \alpha < 1$. α est appelé **facteur d'équilibre**.
- ▶ Partitions α -équilibrées.
- ▶ Cas particulier : n est pair, $\omega(i) = 1$ et $\alpha = \frac{1}{2}$: problème de **bi-section**.

Complexité

- Bi-partitionnement est un problème NP-difficile.

Complexité

- ▶ Bi-partitionnement est un problème NP-difficile.
- ▶ k -Arbres partiels : algorithme quadratique.

Complexité

- ▶ Bi-partitionnement est un problème NP-difficile.
- ▶ k -Arbres partiels : algorithme quadratique.
- ▶ Graphes planaires : problème est ouvert.

Complexité

- ▶ Bi-partitionnement est un problème NP-difficile.
- ▶ k -Arbres partiels : algorithme quadratique.
- ▶ Graphes planaires : problème est ouvert.
- ▶ Si $B = \omega(V)$, on peut appliquer $n - 1$ itérations d'un algorithme de recherche d'un flot maximum.

Complexité

- ▶ Bi-partitionnement est un problème NP-difficile.
- ▶ k -Arbres partiels : algorithme quadratique.
- ▶ Graphes planaires : problème est ouvert.
- ▶ Si $B = \omega(V)$, on peut appliquer $n - 1$ itérations d'un algorithme de recherche d'un flot maximum.
- ▶ Difficulté : partitions équilibrées.

Modélisation des hypergraphes par des graphes

- Problèmes difficiles.

Modélisation des hypergraphes par des graphes

- ▶ Problèmes difficiles.
- ▶ Transformer une instance sur hypergraphe en une instance sur graphe d'un problème de partition (hyper-arête $\equiv$ clique).

Modélisation des hypergraphes par des graphes

- ▶ Problèmes difficiles.
- ▶ Transformer une instance sur hypergraphe en une instance sur graphe d'un problème de partition (hyper-arête $\equiv$ clique).
- ▶ Heuristiques robustes.

Amélioration itérative pour le bi-partitionnement

- Algorithmes qui se sont avérés les plus efficaces pour résoudre ce type de problèmes (d'un point de vue pratique).

Amélioration itérative pour le bi-partitionnement

- ▶ Algorithmes qui se sont avérés les plus efficaces pour résoudre ce type de problèmes (d'un point de vue pratique).
- ▶ Robustes.

Amélioration itérative pour le bi-partitionnement

- ▶ Algorithmes qui se sont avérés les plus efficaces pour résoudre ce type de problèmes (d'un point de vue pratique).
- ▶ Robustes.
- ▶ Peuvent être utilisés dans le cas des hypergraphes.

Amélioration itérative pour le bi-partitionnement

- ▶ Algorithmes qui se sont avérés les plus efficaces pour résoudre ce type de problèmes (d'un point de vue pratique).
- ▶ Robustes.
- ▶ Peuvent être utilisés dans le cas des hypergraphes.
- ▶ Poids arbitraires sur les sommets.

Amélioration itérative pour le bi-partitionnement

- ▶ Algorithmes qui se sont avérés les plus efficaces pour résoudre ce type de problèmes (d'un point de vue pratique).
- ▶ Robustes.
- ▶ Peuvent être utilisés dans le cas des hypergraphes.
- ▶ Poids arbitraires sur les sommets.
- ▶ Coûts des arêtes quelconques.

Amélioration itérative pour le bi-partitionnement

- ▶ Algorithmes qui se sont avérés les plus efficaces pour résoudre ce type de problèmes (d'un point de vue pratique).
- ▶ Robustes.
- ▶ Peuvent être utilisés dans le cas des hypergraphes.
- ▶ Poids arbitraires sur les sommets.
- ▶ Coûts des arêtes quelconques.
- ▶ Critères d'équilibre.

Amélioration itérative pour le bi-partitionnement

- ▶ Algorithmes qui se sont avérés les plus efficaces pour résoudre ce type de problèmes (d'un point de vue pratique).
- ▶ Robustes.
- ▶ Peuvent être utilisés dans le cas des hypergraphes.
- ▶ Poids arbitraires sur les sommets.
- ▶ Coûts des arêtes quelconques.
- ▶ Critères d'équilibre.
- ▶ Heuristiques utilisées dans des algorithmes de type "diviser pour régner" pour les problème de placement et de "floorplanning".

Idée de base

Problème de bi-section : Méthode de recherche locale.

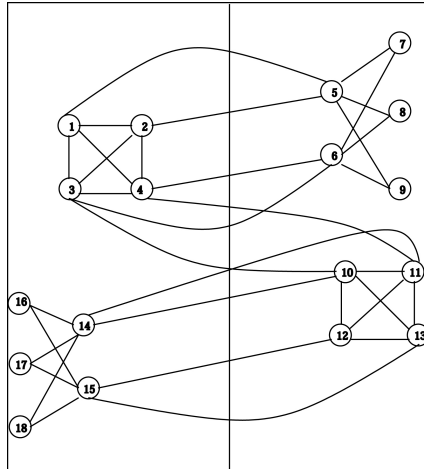
- ▶ Commencer par une bi-section initiale.

Idée de base

Problème de bi-section : Méthode de recherche locale.

- ▶ Commencer par une bi-section initiale.
- ▶ Echanger des paires de sommets de la coupe pour améliorer sa taille (cutsizes).

Exemple



Principe

Choix de la paire de sommets permettant

- ▶ la plus grande diminution de la taille de la coupe **ou**

Principe

Choix de la paire de sommets permettant

- ▶ la plus grande diminution de la taille de la coupe **ou**
- ▶ la plus petite augmentation de la taille de la coupe s'il n'est pas possible de la diminuer.

Principe

Principe

- Problème : Eviter de reconsidérer les sommets.

Principe

- ▶ Problème : Eviter de reconsidérer les sommets.
- ▶ On verrouille les sommets de l'itération courante.

Principe

- ▶ Problème : Eviter de reconsidérer les sommets.
- ▶ On verrouille les sommets de l'itération courante.
- ▶ Critère d'arrêt : Tous les sommets sont verrouillés.

Principe

- ▶ Problème : Eviter de reconsidérer les sommets.
- ▶ On verrouille les sommets de l'itération courante.
- ▶ Critère d'arrêt : Tous les sommets sont verrouillés.

Dérouler l'exemple.

Algorithme

Calcul des gains sur la taille de la coupe notés $D(i)$

- ▶ Coût des arêtes externes : $E(i) = \sum_{e=\{v_i, v_j\}, v_j \in B} c(e)$

Algorithme

Calcul des gains sur la taille de la coupe notés $D(i)$

- ▶ Coût des arêtes externes : $E(i) = \sum_{e=\{v_i, v_j\}, v_j \in B} c(e)$
- ▶ Coût des arêtes internes : $I(i) = \sum_{e=\{v_i, v_j\}, v_j \in A} c(e)$

Algorithme

Calcul des gains sur la taille de la coupe notés $D(i)$

- ▶ Coût des arêtes externes : $E(i) = \sum_{e=\{v_i, v_j\}, v_j \in B} c(e)$
- ▶ Coût des arêtes internes : $I(i) = \sum_{e=\{v_i, v_j\}, v_j \in A} c(e)$
- ▶ Gain : $D(i) = E(i) - I(i)$

Algorithme

Calcul des gains sur la taille de la coupe notés $D(i)$

- ▶ Coût des arêtes externes : $E(i) = \sum_{e=\{v_i, v_j\}, v_j \in B} c(e)$
- ▶ Coût des arêtes internes : $I(i) = \sum_{e=\{v_i, v_j\}, v_j \in A} c(e)$
- ▶ Gain : $D(i) = E(i) - I(i)$
- ▶ Calcul de $D(i) + D(j) - 2c(i, j)$

Complexité

- Si l'on examine tous les échanges possibles : $O(n^3)$

Complexité

- ▶ Si l'on examine tous les échanges possibles : $O(n^3)$
- ▶ En utilisant des structures de données adaptées : $O(n^2 \log(n))$

Avantages

- ▶ Cet algorithme peut être utilisé plusieurs fois.

Avantages

- ▶ Cet algorithme peut être utilisé plusieurs fois.
- ▶ Robustesse.

Avantages

- ▶ Cet algorithme peut être utilisé plusieurs fois.
- ▶ Robustesse.
- ▶ Contraintes additionnelles.

Inconvénients

- Poids unitaires sur les sommets.

Inconvénients

- ▶ Poids unitaires sur les sommets.
- ▶ Bi-sections exactes.

Inconvénients

- ▶ Poids unitaires sur les sommets.
- ▶ Bi-sections exactes.
- ▶ Ne s'applique pas aux hypergraphes.

Inconvénients

- ▶ Poids unitaires sur les sommets.
- ▶ Bi-sections exactes.
- ▶ Ne s'applique pas aux hypergraphes.
- ▶ Complexité relativement élevée.

Inconvénients

- ▶ Poids unitaires sur les sommets.
- ▶ Bi-sections exactes.
- ▶ Ne s'applique pas aux hypergraphes.
- ▶ Complexité relativement élevée.
- ▶ Tous les choix ne sont pas déterministes.

Idée de base

Pallier quatre premiers inconvénients cités pour l'heuristique de Kernighan-Lin :

- ▶ Un seul sommet est déplacé à travers la coupe dans un mouvement.

Idée de base

Pallier quatre premiers inconvénients cités pour l'heuristique de Kernighan-Lin :

- ▶ Un seul sommet est déplacé à travers la coupe dans un mouvement.
- ▶ Extensions des valeurs D aux hypergraphes.

Idée de base

Pallier quatre premiers inconvénients cités pour l'heuristique de Kernighan-Lin :

- ▶ Un seul sommet est déplacé à travers la coupe dans un mouvement.
- ▶ Extensions des valeurs D aux hypergraphes.
- ▶ Algorithme de sélection des sommets à déplacer à travers la coupe.

Complexité

► $O(pb_{max})$,

Complexité

- ▶ $O(pb_{max})$,
- ▶ p - nombre d'hyper-arêtes et $b_{max} = \max_{e \in E} b(e)$

Complexité

- ▶ $O(pb_{max})$,
- ▶ p - nombre d'hyper-arêtes et $b_{max} = \max_{e \in E} b(e)$
- ▶ Si $b_{max} = 1$, l'algorithme est linéaire.

Problème de la multiway partition

Les heuristiques précédentes peuvent être utilisées pour résoudre le problème de multiway partition:

- ▶ Partir d'une partition arbitraire du graphe en r ensembles de taille égale.

Problème de la multiway partition

Les heuristiques précédentes peuvent être utilisées pour résoudre le problème de multiway partition:

- ▶ Partir d'une partition arbitraire du graphe en r ensembles de taille égale.
- ▶ Appliquer l'heuristique à chaque paire de sous-ensembles.

Principes

- ▶ *Kirkpatrick* (1983), famille des méthodes stochastiques,

Principes

- ▶ *Kirkpatrick* (1983), famille des méthodes stochastiques,
- ▶ évolution d'un système désordonné vers un système ordonné,

Principes

- ▶ *Kirkpatrick* (1983), famille des méthodes stochastiques,
- ▶ évolution d'un système désordonné vers un système ordonné,
- ▶ fonction coût $\equiv$ énergie,

Principes

- ▶ *Kirkpatrick* (1983), famille des méthodes stochastiques,
- ▶ évolution d'un système désordonné vers un système ordonné,
- ▶ fonction coût $\equiv$ énergie,
- ▶ structures gelées $\equiv$ minima locaux,

Principes

- ▶ *Kirkpatrick* (1983), famille des méthodes stochastiques,
- ▶ évolution d'un système désordonné vers un système ordonné,
- ▶ fonction coût $\equiv$ énergie,
- ▶ structures gelées $\equiv$ minima locaux,
- ▶ généralisation de l'algorithme de *Métropolis*: acceptation d'une configuration avec une probabilité $p = e^{-(E_{k+1}-E_k)/T}$ (distribution de Boltzmann),

Principes

- ▶ *Kirkpatrick* (1983), famille des méthodes stochastiques,
- ▶ évolution d'un système désordonné vers un système ordonné,
- ▶ fonction coût $\equiv$ énergie,
- ▶ structures gelées $\equiv$ minima locaux,
- ▶ généralisation de l'algorithme de *Métropolis*: acceptation d'une configuration avec une probabilité $p = e^{-(E_{k+1}-E_k)/T}$ (distribution de Boltzmann),
- ▶ convergence théorique de l'algorithme (chaînes de Markov).

Algorithme du recuit simulé

recuit(c_a, η, c_p, c_T)

Générer une solution initiale s^0 dans S $s^* \leftarrow s \leftarrow s^0$

Choisir une température initiale $T^0 > 0$

Tant Que le système n'est pas figé (c_a) **Faire**

Tant que l'équilibre thermodynamique n'est pas atteint

 Choisir un voisin $v \in V(s)$ par l'application de g_a

$\Delta F = F(v) - F(s)$

Si $\Delta F < 0$ alors $s \leftarrow v$

Si $F(v) \leq F(s^*)$ alors $s^* \leftarrow v$

Sinon $s \leftarrow v$ avec la probabilité $\exp\left(-\frac{\Delta F}{T}\right)$

Fin Tant Que

 Abaisser la température $T = \eta T$

Fin Tant Que

Paramètres

- Température initiale.

Paramètres

- ▶ Température initiale.
- ▶ Loi de décroissance de la température.

Paramètres

- ▶ Température initiale.
- ▶ Loi de décroissance de la température.
- ▶ Nombre d'itérations.

Paramètres

- ▶ Température initiale.
- ▶ Loi de décroissance de la température.
- ▶ Nombre d'itérations.
- ▶ Critère d'arrêt: refroidissement total du système.

Adaptation

- Configuration ou état: bi-partition de G .

Adaptation

- Configuration ou état: bi-partition de G .
- Coût d'une bi-partition (pénalisation):
$$c(V_1, V_2) = \sum_{e=\{v_1, v_2\}, v_1 \in V_1, v_2 \in V_2} c(e) + \beta(\omega(V_1) - \omega(V_2))^2.$$