Optimisation sur les graphes et applications aux problèmes de routage et synthèse dans les réseaux

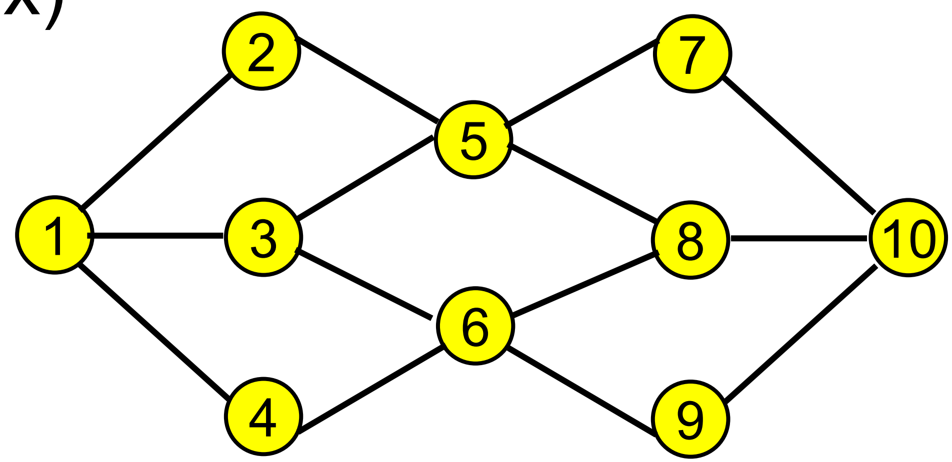
Celso C. Ribeiro

Universidade Federal Fluminense

Brésil

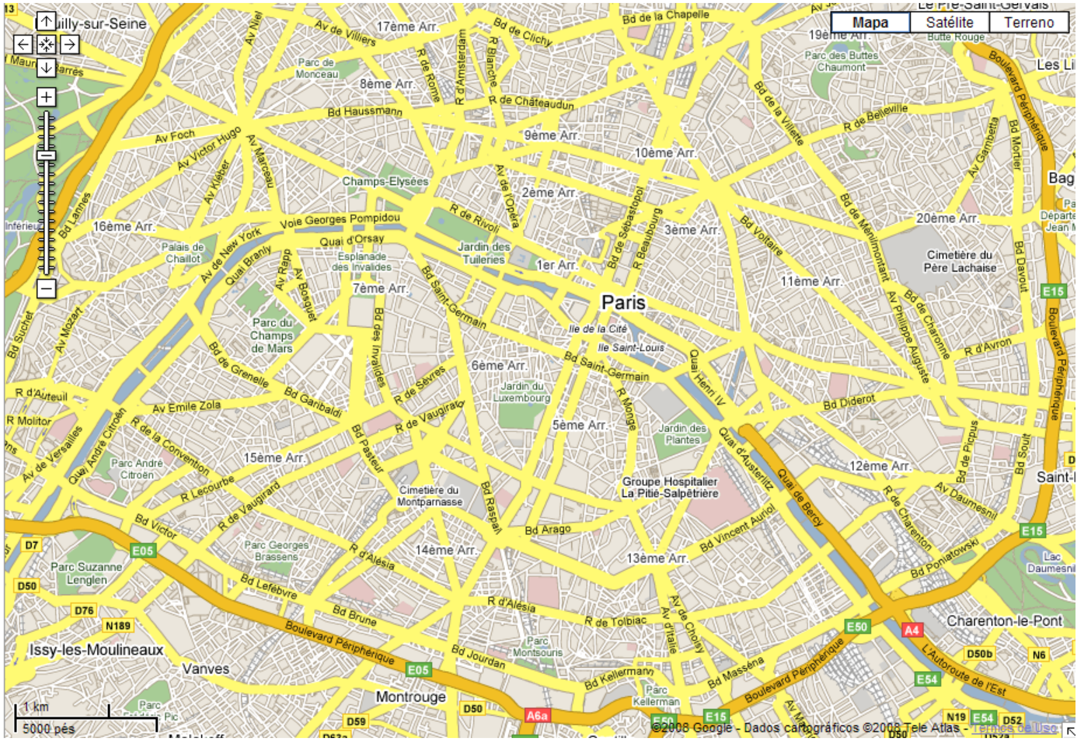
Graphes

- Graphe G=(V,E)
  V: ensemble de sommets
  E: ensemble d'arêtes (arcs)
- Données:
  - Capacités (min, max)
  - Distances
  - Coûts:
    - Transport
    - Construction
    - Communication



Graphes

- Applications aux réseaux de:
  - Transport
    - Urbain
    - Routier
  - Communications
  - Energie
  - Internet

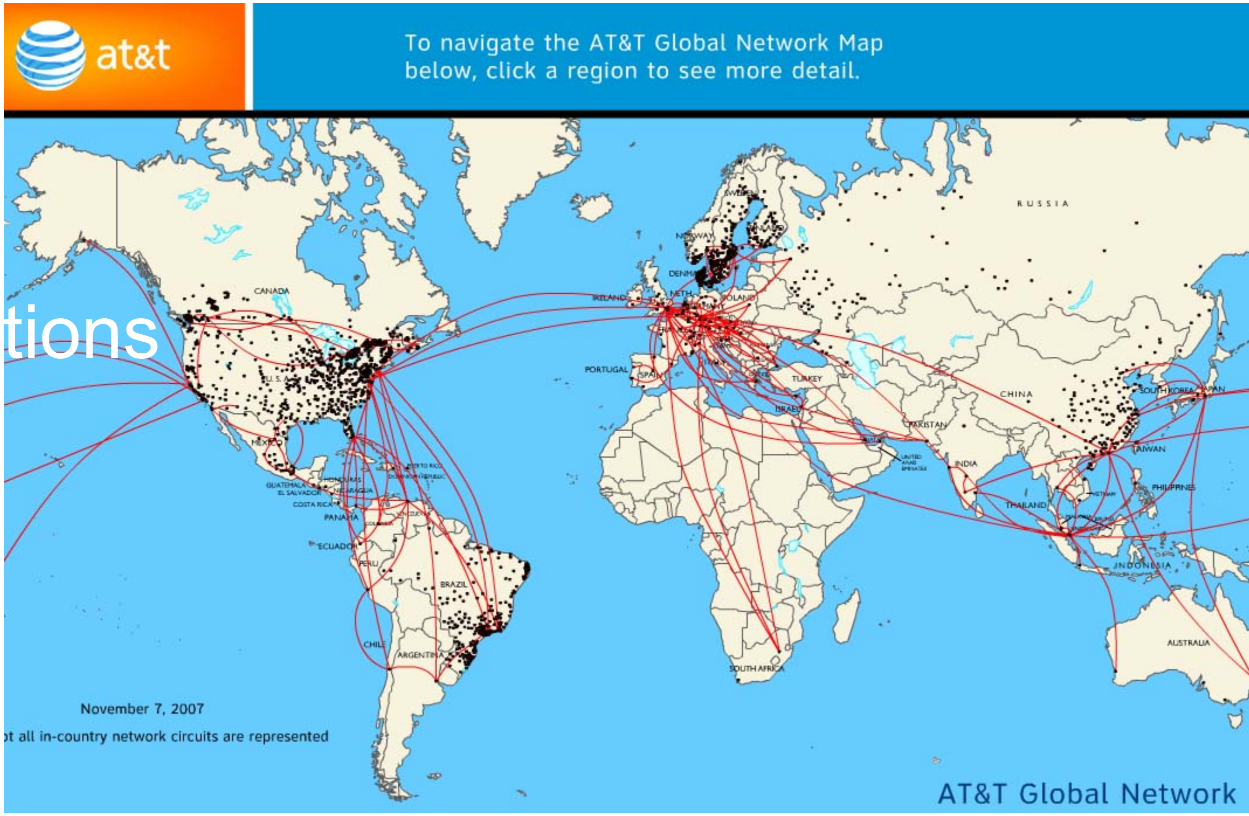


Graphes

- Applications aux réseaux de:
  - Transport:
    - Urbain
    - Routier
  - Communications
  - Energie
  - Internet

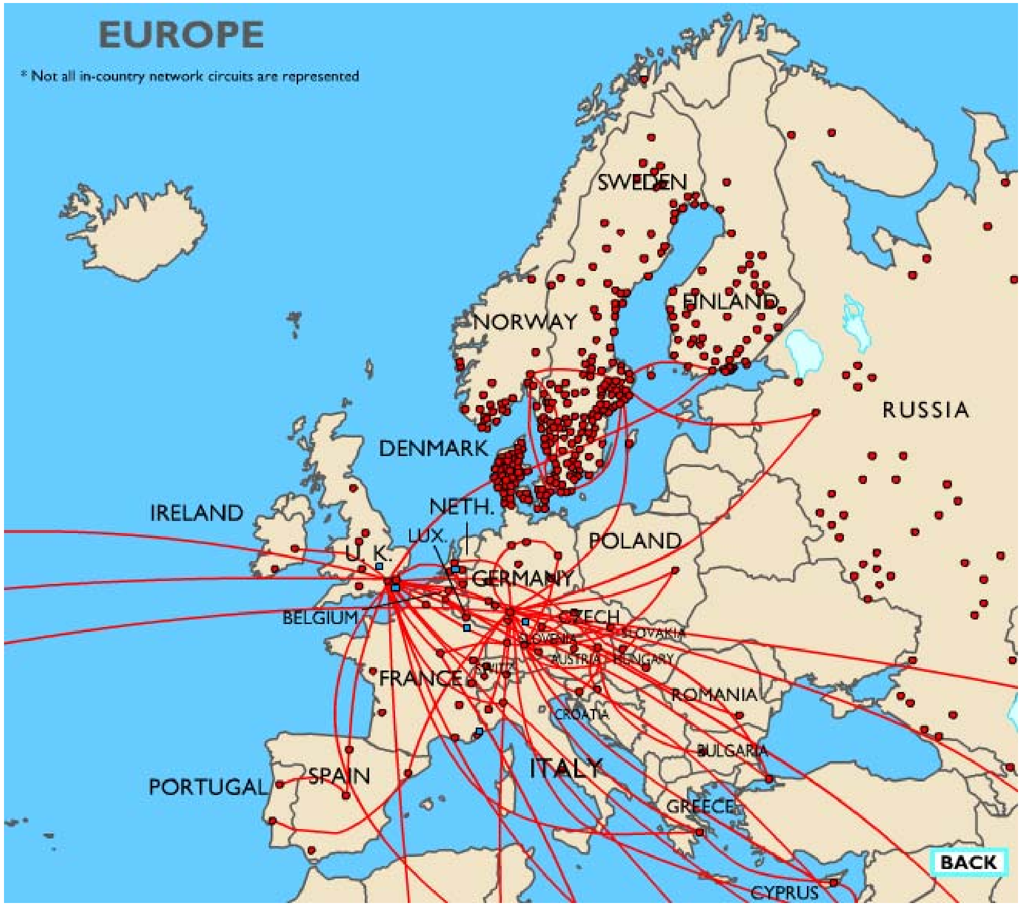
Graphes

- Applications aux réseaux de:
  - Transport:
    - Urbain
    - Routier
  - Communications
  - Energie
  - Internet



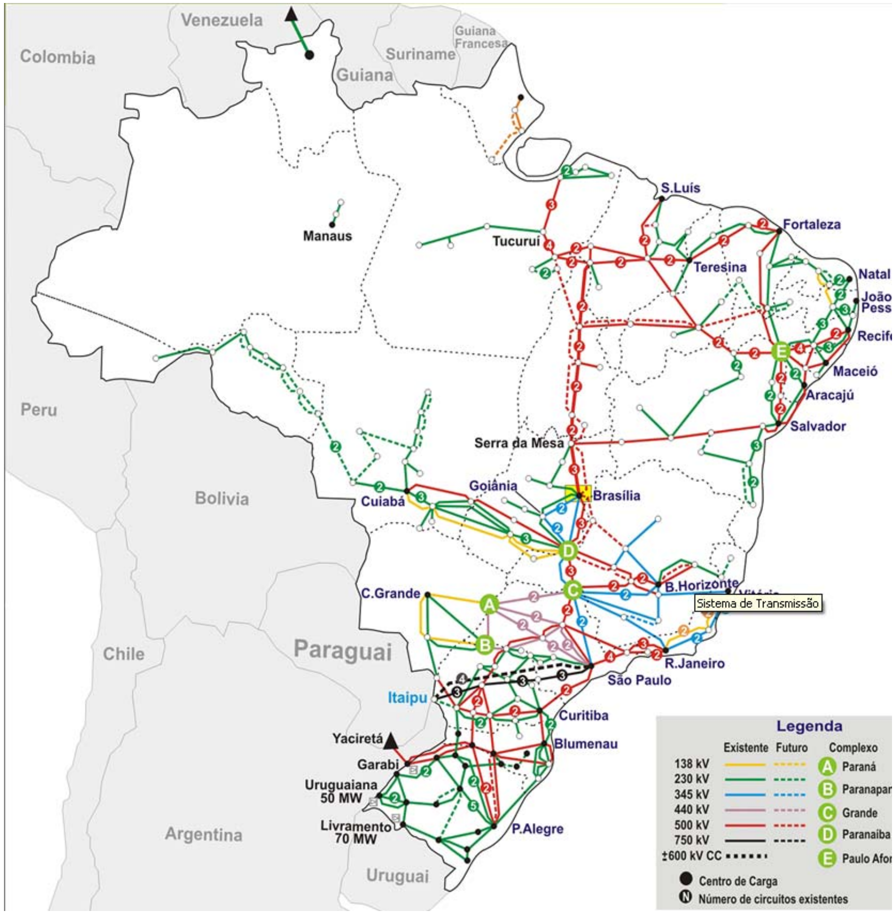
Graphes

- Applications aux réseaux de:
  - Transport:
    - Urbain
    - Routier
  - Communications
  - Energie
  - Internet



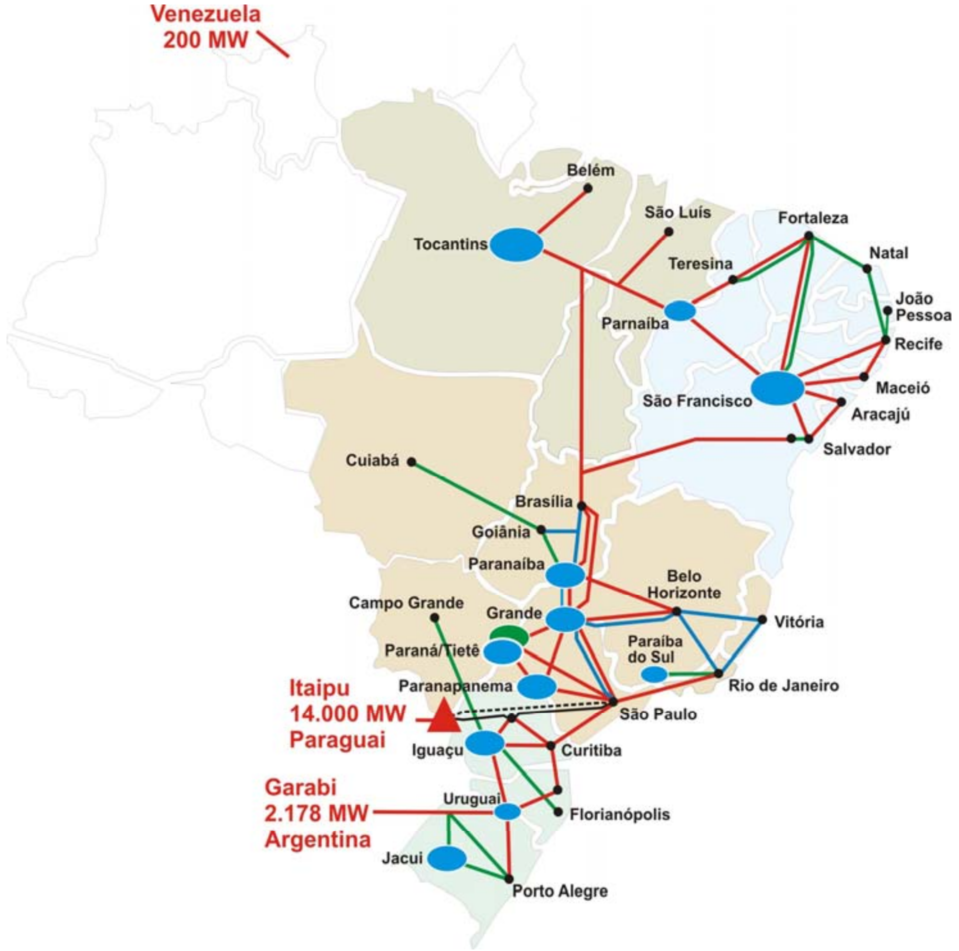
Graphes

- Applications aux réseaux de:
  - Transport:
    - Urbain
    - Routier
  - Communications
  - Energie
  - Internet



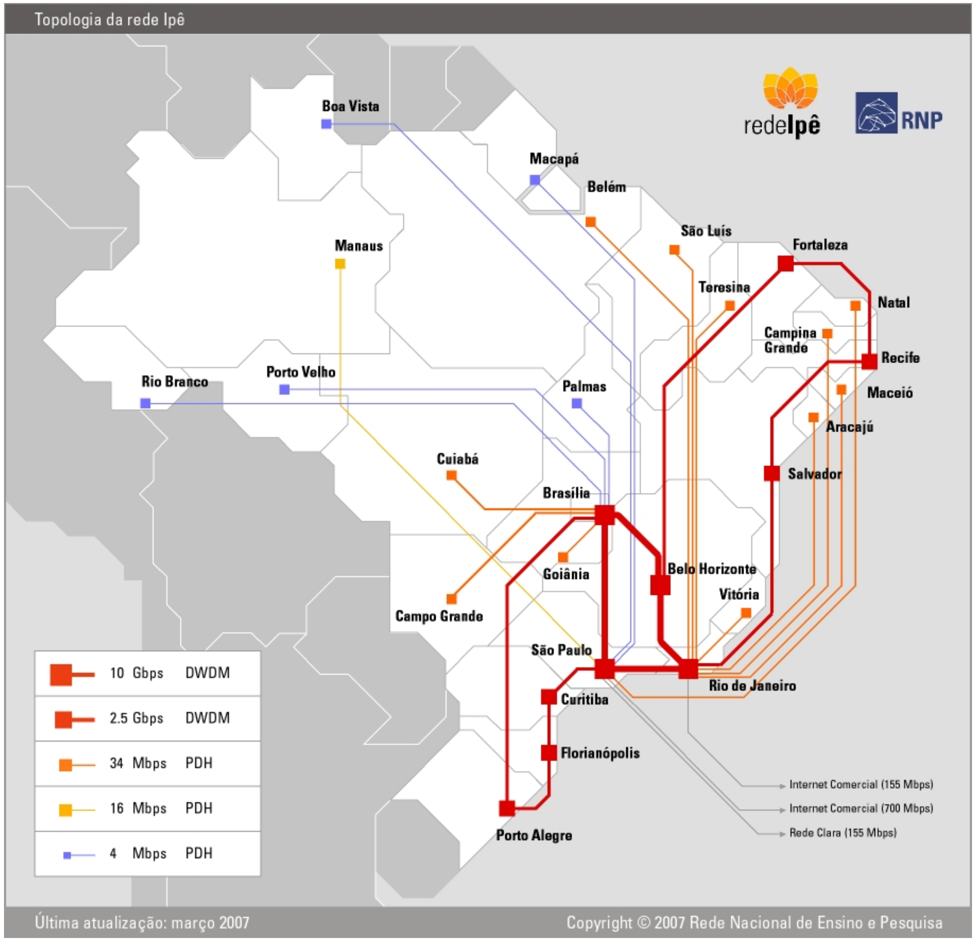
Graphes

- Applications aux réseaux de:
  - Transport:
    - Urbain
    - Routier
  - Communications
  - Energie
  - Internet



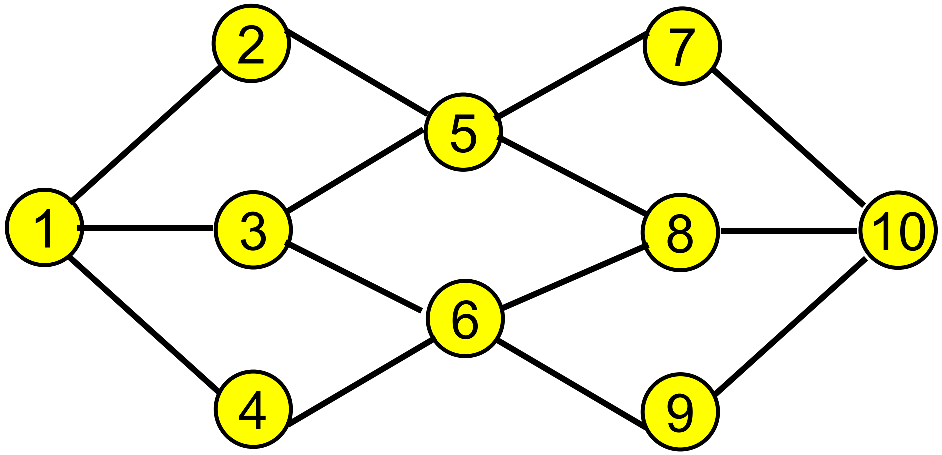
Graphes

- Applications aux réseaux de:
  - Transport:
    - Urbain
    - Routier
  - Communications
  - Energie
  - Internet



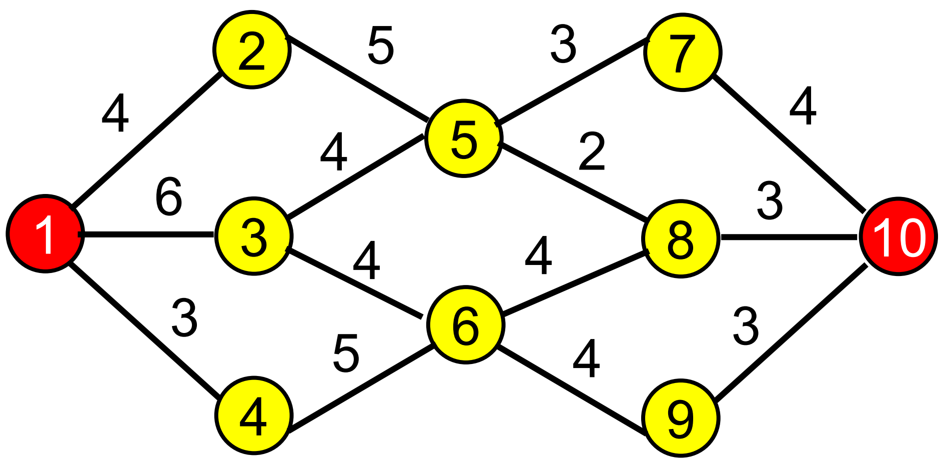
Problèmes classiques

- Plus court chemin d'un sommet à l'autre
  - Donnés:
    - Sommets de départ et de destination
    - Distance entre chaque pair de sommets



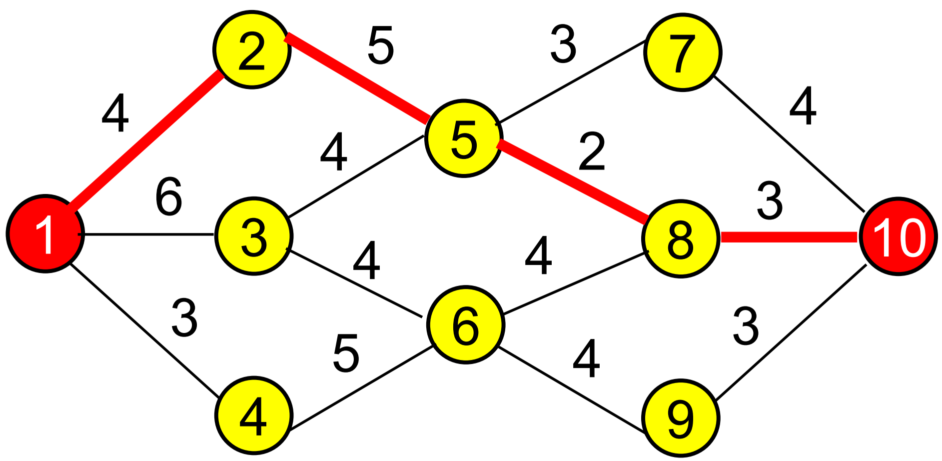
Problèmes classiques

- Plus court chemin d'un sommet à l'autre
  - Donnés:
    - Sommets de départ et de destination
    - Distance entre chaque pair de sommets



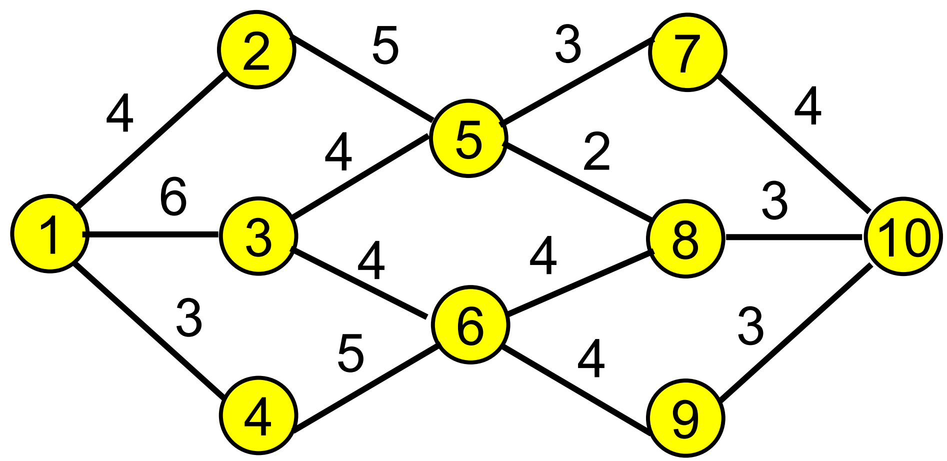
Problèmes classiques

- Plus court chemin d'un sommet à l'autre
  - Donnés:
    - Sommets de départ et de destination
    - Distance entre chaque pair de sommets



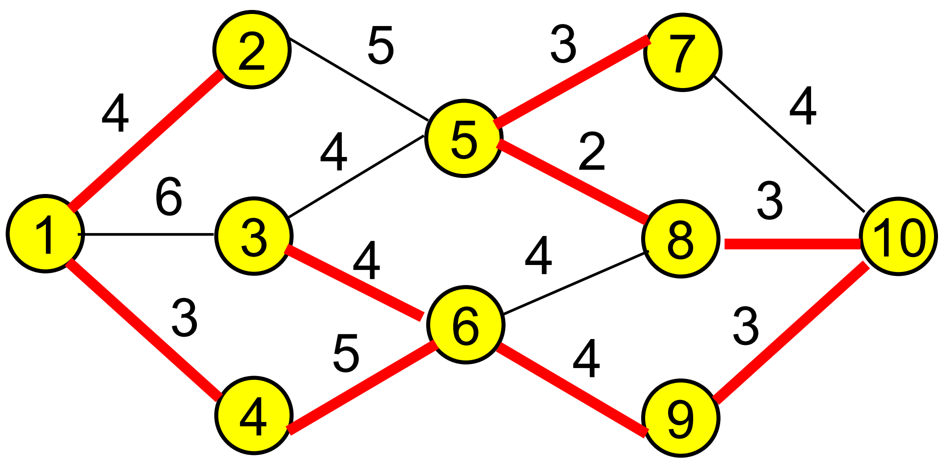
Problèmes classiques

- Arbre générateur de coût minimum
  - Donnés:
    - Coût de chaque arête



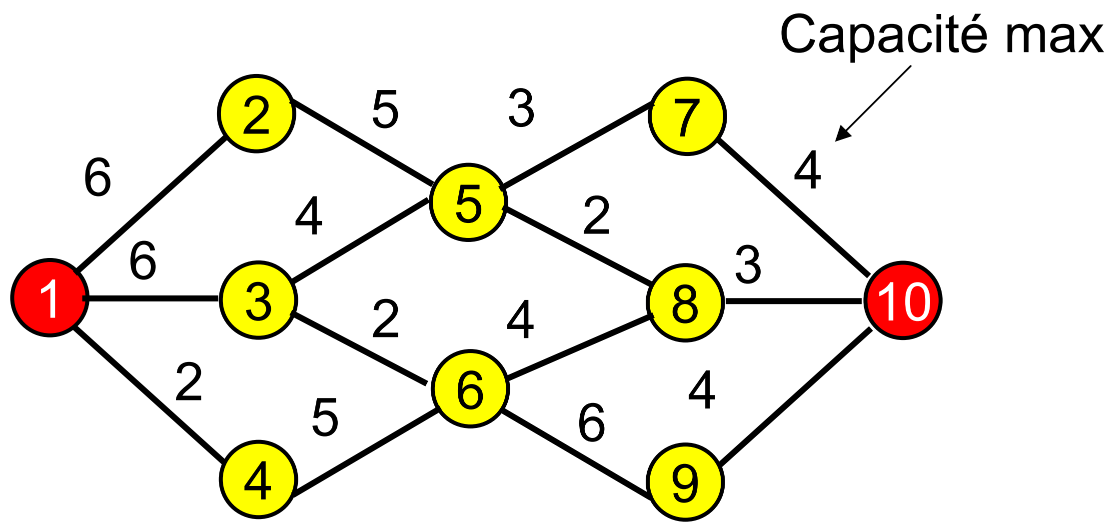
Problèmes classiques

- Arbre générateur de coût minimum
  - Donnés:
    - Coût de chaque arête



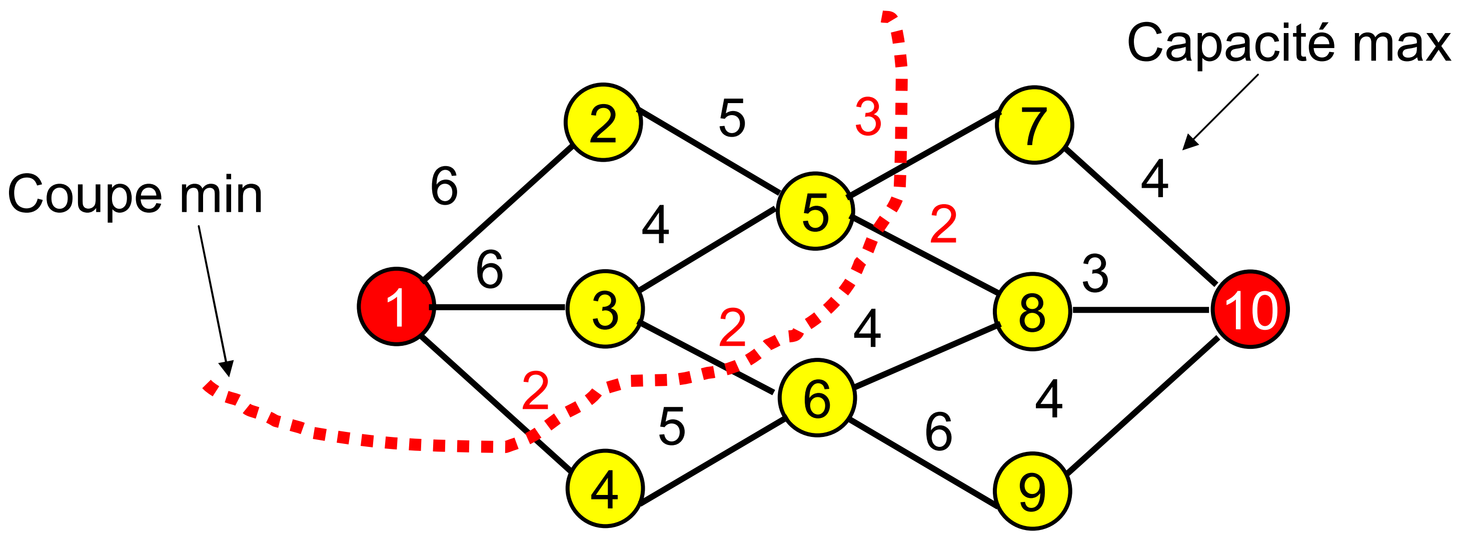
Problèmes classiques

- Flot maximum d'un sommet à l'autre
  - Donnés:
    - Sommets de départ et de destination
    - Capacité maximum (minimum) de chaque arête



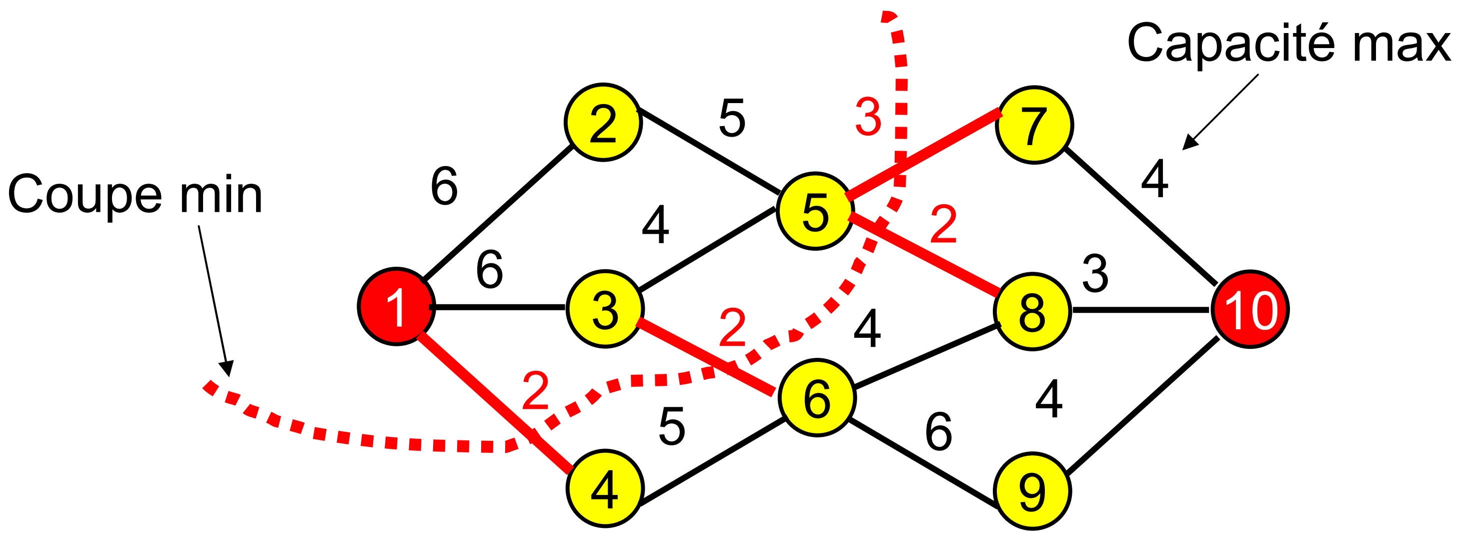
Problèmes classiques

- Flot maximum d'un sommet à l'autre
  - Donnés:
    - Sommets de départ et de destination
    - Capacité maximum (minimum) de chaque arête



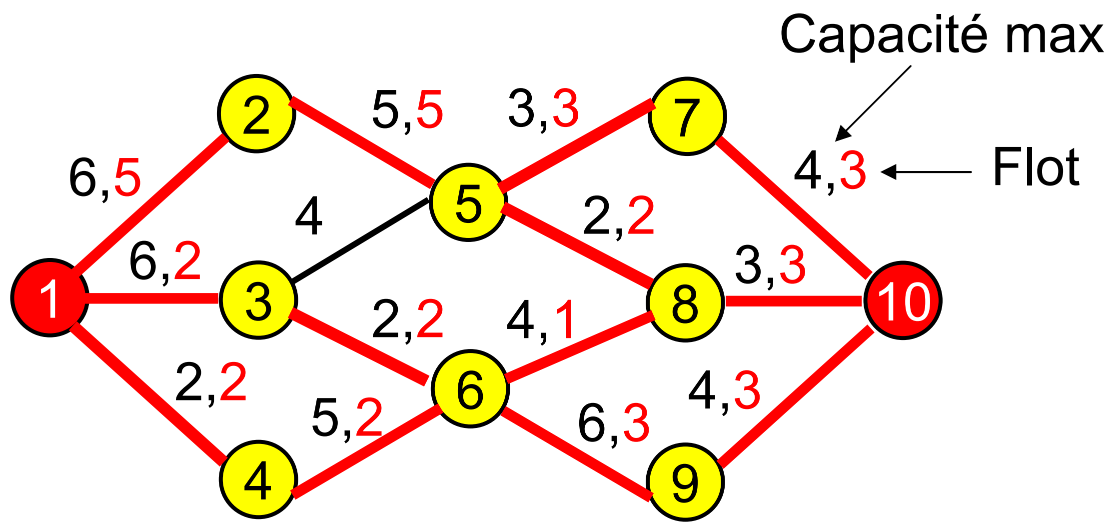
Problèmes classiques

- Flot maximum d'un sommet à l'autre
  - Donnés:
    - Sommets de départ et de destination
    - Capacité maximum (minimum) de chaque arête



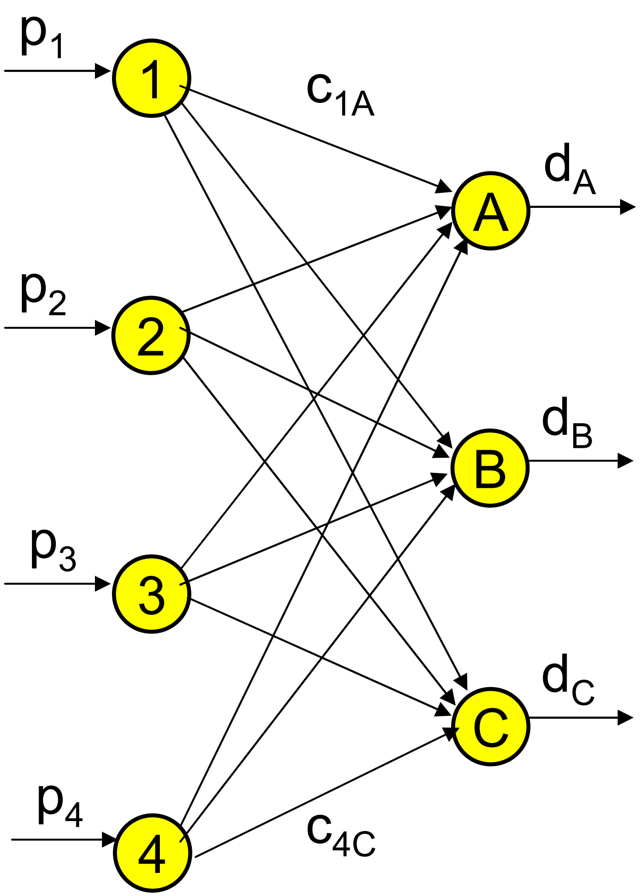
Problèmes classiques

- Flot maximum d'un sommet à l'autre
  - Donnés:
    - Sommets de départ et de destination
    - Capacité maximum (minimum) de chaque arête



Problèmes classiques

- Problème de transport
  - Donnés:
    - Capacité de production de chaque usine
    - Demande de chaque entrepôt
    - Coûts unitaires de transport
    - Déterminer les quantités à envoyer de chaque usine vers chaque entrepôt



Problèmes classiques

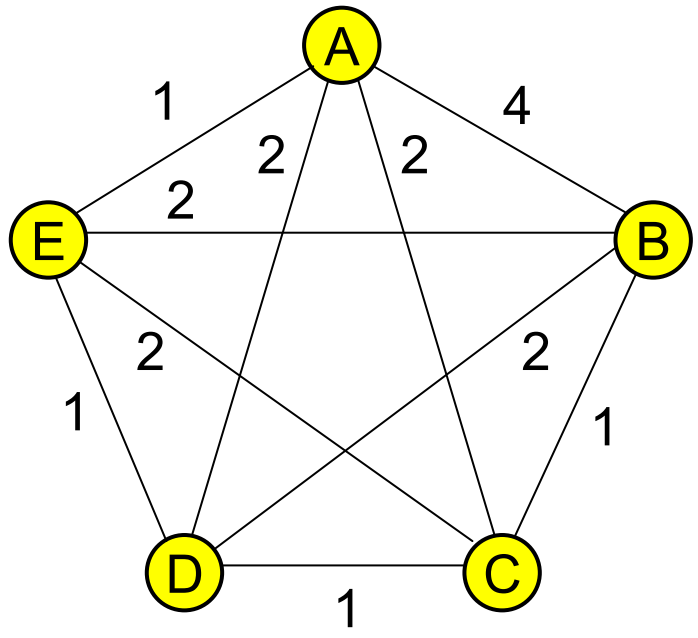
- Algorithmes polynomiaux: les temps de calcul croissent polynomialement avec la taille du problème (rapides)
  - $O(n)$, $O(n \log n)$, $O(n^2)$, $O(n^3)$
- Algorithmes rapides (efficients) →
  → problèmes "faciles" (bien résolus)
- Autres examples:
  - Problème d'afféctation
  - Problème du flot à coût minimum
  - Problème de couplage, etc.

Problèmes difficiles

- Problème du voyageur de commerce
  - Données:
    - Ensemble de n villes: graphe avec n sommets
    - Distance entre chaque paire de sommets
  - Déterminer un cycle de longueur minimale qui visite chaque sommet exactement une seule fois

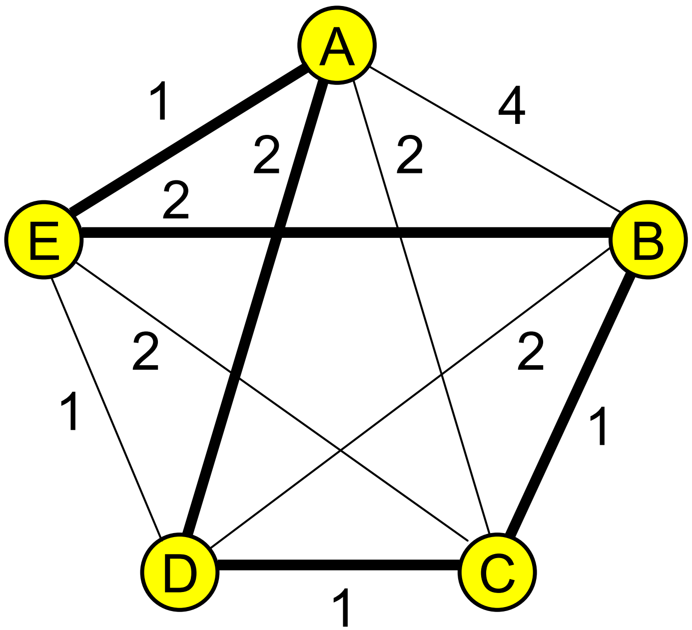
Problèmes difficiles

- Problème du voyageur de commerce



Problèmes difficiles

- Problème du voyageur de commerce



Coût minimum: 7

Problèmes difficiles

- Problème du voyageur de commerce
  - Il ny a pas d'algorithme de complexité polynomiale connu pour le problème du voyageur de commerce
  - Les temps de calcul augmentent exponentiellement avec la taille du problème: $O(n!)$, $O(2^n)$
  - Problèmes NP-complets et NP-difficiles

Problèmes difficiles

- Autres examples:
  - Problème du postier chinois
  - Problème de Steiner
  - Problème de coloriage d’un graphe
  - Problème d’affectation quadratique, etc.

Méthodes éxactes de solution

- Programmation linéaire: minimiser une fonction linéaire de variables réelles, sous un ensemble de contraintes linéaires
- Algorithmes:
  - Simplexe
  - Points intérieurs (polynomial)

$$\text{minimiser } c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

$$\text{sujeito a}:$$

$$\mathrm{a}_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1$$

$$\mathrm{a}_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2$$

$$\ldots$$

$$\mathrm{a}_{\mathrm{m}}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m$$

$$x_1, x_2, \ldots, x_n \geq 0$$

Méthodes éxactes de solution

- Programmation en nombres entiers: minimiser une fonction linéaire de variables qui prennent des valeurs entières, sous un ensemble de contraintes linéaires

$$\begin{aligned}
&\text{minimiser } c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \\
&\text{sujeito a :} \\
&\mathrm{a}_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n = b_1 \\
&\mathrm{a}_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n = b_2 \\
&\ldots \\
&\mathrm{a}_{\mathrm{m}} x_1 + a_{m2} x_2 + \ldots + a_{mn} x_n = b_m \\
&x_1, x_2, \ldots, x_n \geq 0 \text{ et entiers } (0\text{-}1)
\end{aligned}$$

- Algorithmes exacts:
  - Méthodes de coupes
  - Méthodes de recherche arborescente (branch & bound)
  - Problème NP-difficile

Méthodes éxactes de solution

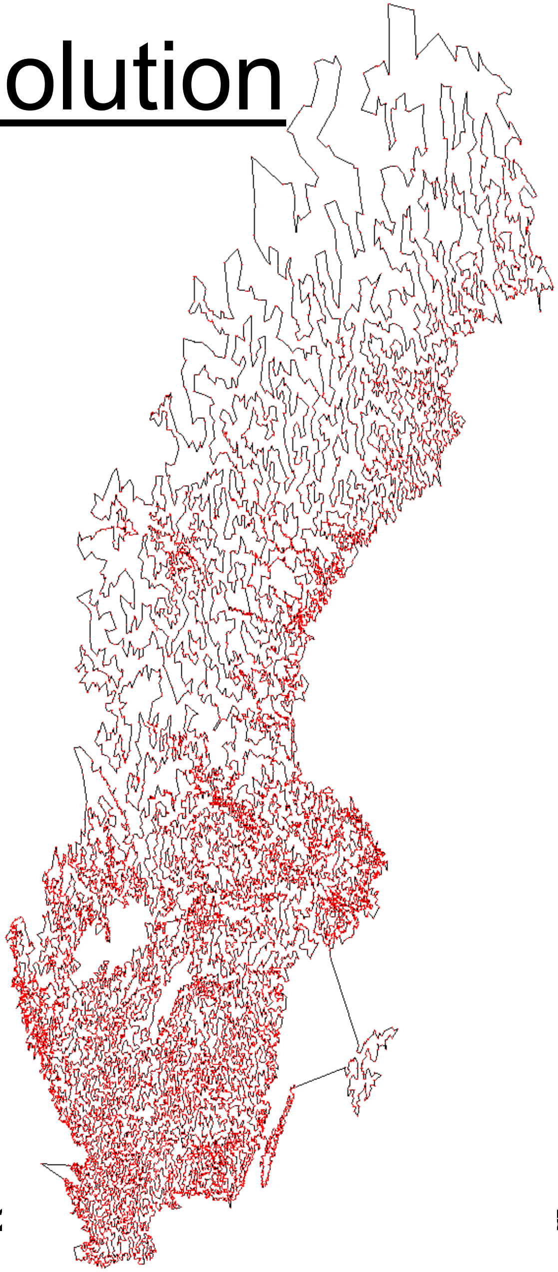
- Programmation en nombres entiers:
  - Problèmes en variables 0-1
  - Problèmes de décision
  - Problèmes d’optimisation combinatoire
- Dans le cas général, il s’agit de problèmes très difficiles, pour lesquels il n’y a pas d’algorithme connu de complexité polynomiale

Méthodes éxactes de solution

- Les parallélisme permet d'accélérer les calculs, mais pas de réduire leur complexité: contribution limitée
- Voyageur de commerce:
  - Solution exacte pour 42 villes: Dantzig, Fulkerson & Johnson (1954)
  - 24.978 villes suédoises: Applegate, Bixby, Chvátal, Cook & Helsgaun (2006)

Méthodes éxactes de solution

24.978 villes suédoises (2006)



Méthodes approchées

- Traveling tournament problem: déterminer les dates des matches d’un tournoi disputé par n équipes, de façon à minimiser la distance voyagé par les joueurs
  - Plus grand problème déjà résolu: 8 équipes
- Méthodes approchées (heuristiques): recherche d’optima locaux très proches des solutions optimales (plus rapides que les méthodes éxactes)

Méthodes approchées

- Heuristiques:
  - Constructives: construction d’une solution réalisable de départ
  - Recherche locale: amélioration de la solution courante jusqu’à ce qu’un optimum local soit trouvé
  - Metaheuristiques: recherche au delà du premier optimum local
- Compromis entre qualité et temps de calcul

Méthodes approchées

- Metaheuristiques: fondées sur différentes stratégies (principes) pour échapper des optima locaux
  - Récuit simulé
  - Recherche tabou
  - GRASP
  - Algorithmes génétiques
  - Collones de fourmis

Méthodes de trajéctoire

Méthodes de populations

- "Méthodes inspirées par la nature"

Applications aux télécommunications

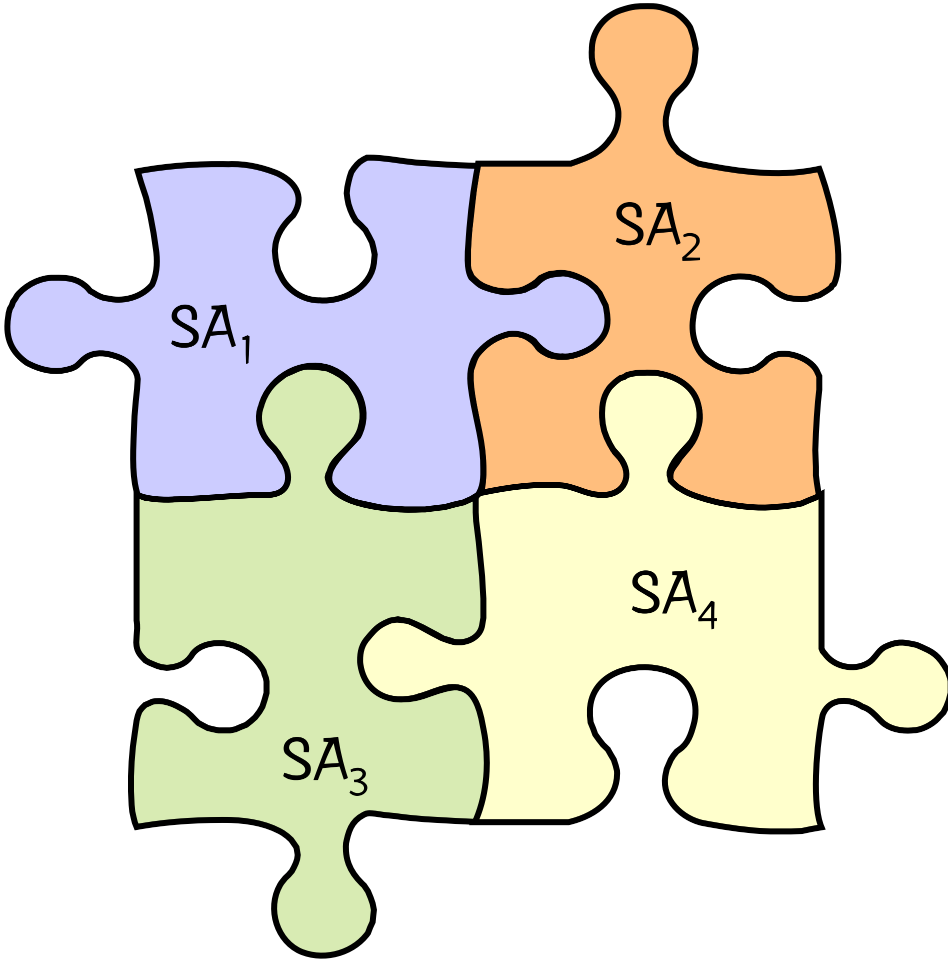
- Metaheuristiques
- Calcul parallèle (grilles)
- Problèmes de synthèse de réseaux →
  → problèmes de contrôle
- Problèmes statiques →
  → problèmes dynamiques (temps réel)

Applications aux télécommunications

- Applications:
  - Routage sous le protocole OSPF
  - Routage des circuits virtuels
  - Routage sur les réseaux optiques
  - Synthèse des réseaux d'accès
  - Réseaux ad-hoc et réseaux de senseurs

Routage sous le protocole OSPF

- Internet organisée en systèmes autonomes: routage à l'intérieur de chaque système
- Protocole OSPF de routage interne sur l'Internet: Optimal Shortest Path First

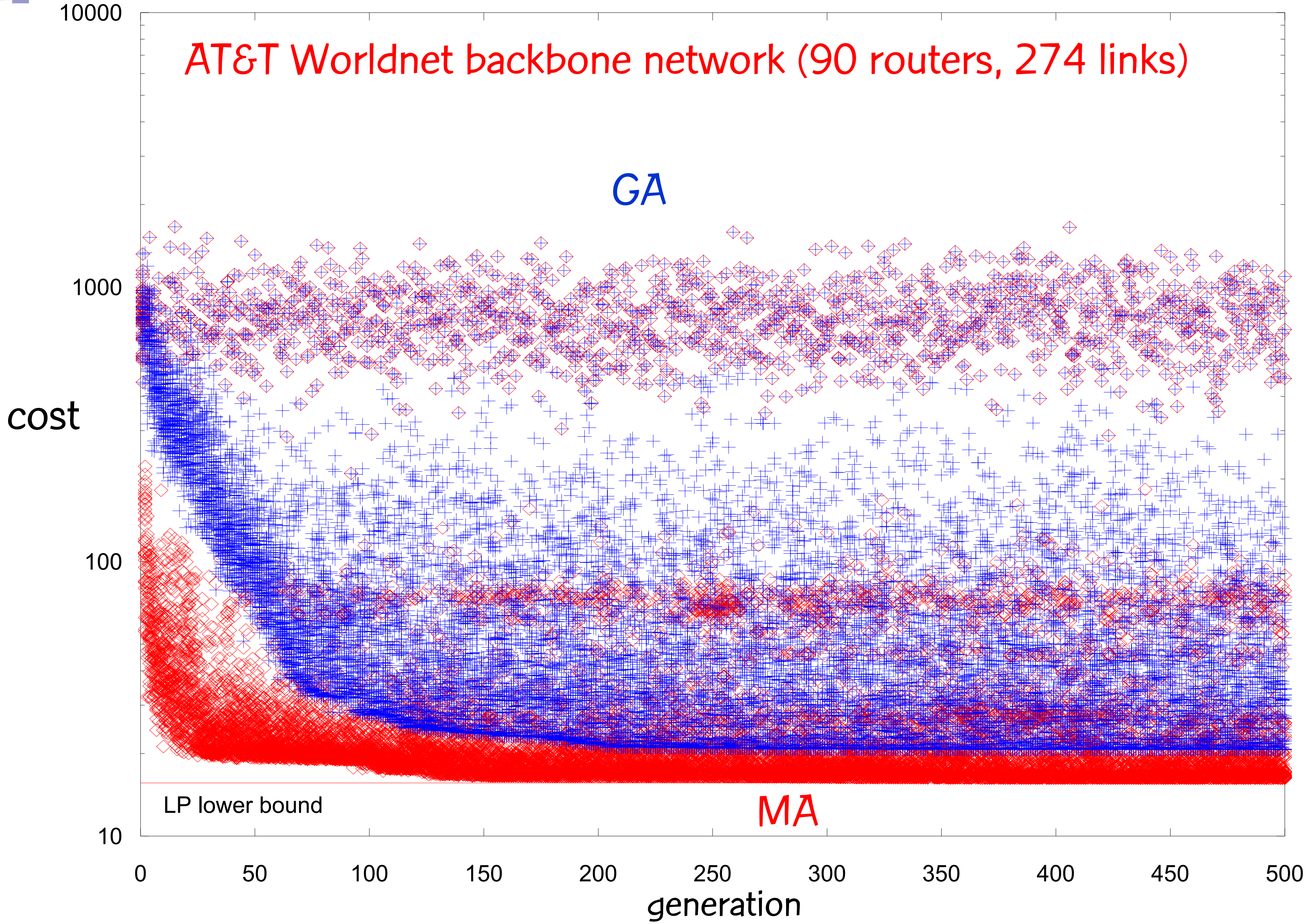


Routage sous le protocole OSPF

- Poids associés à chaque arête du réseau dans le système autonome
- Chaque routeur détérmine la plus courte route vers chaque système adjacent
- Chaque paquet réçu par un routeur est envoyé au suivant sur la plus courte route vers le système de destination

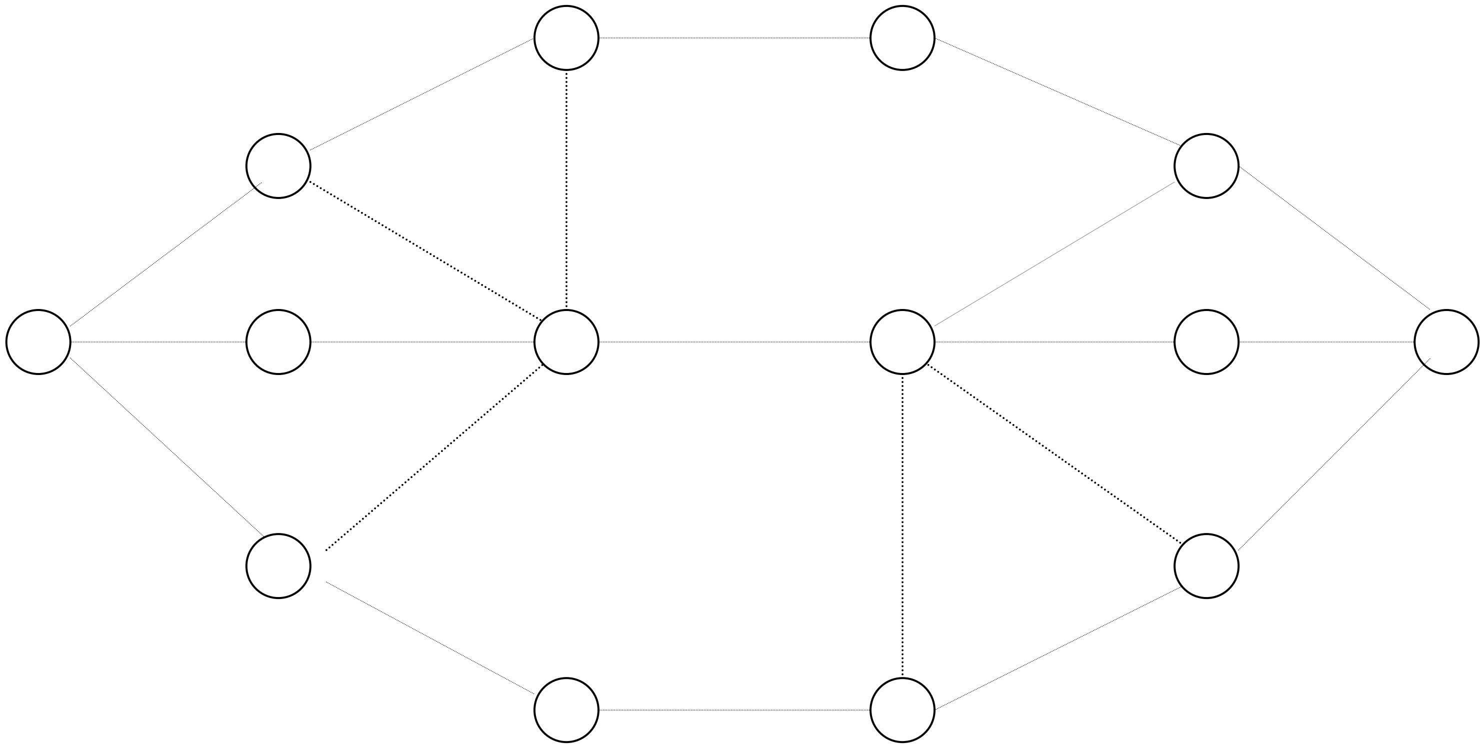
Routage sous le protocole OSPF

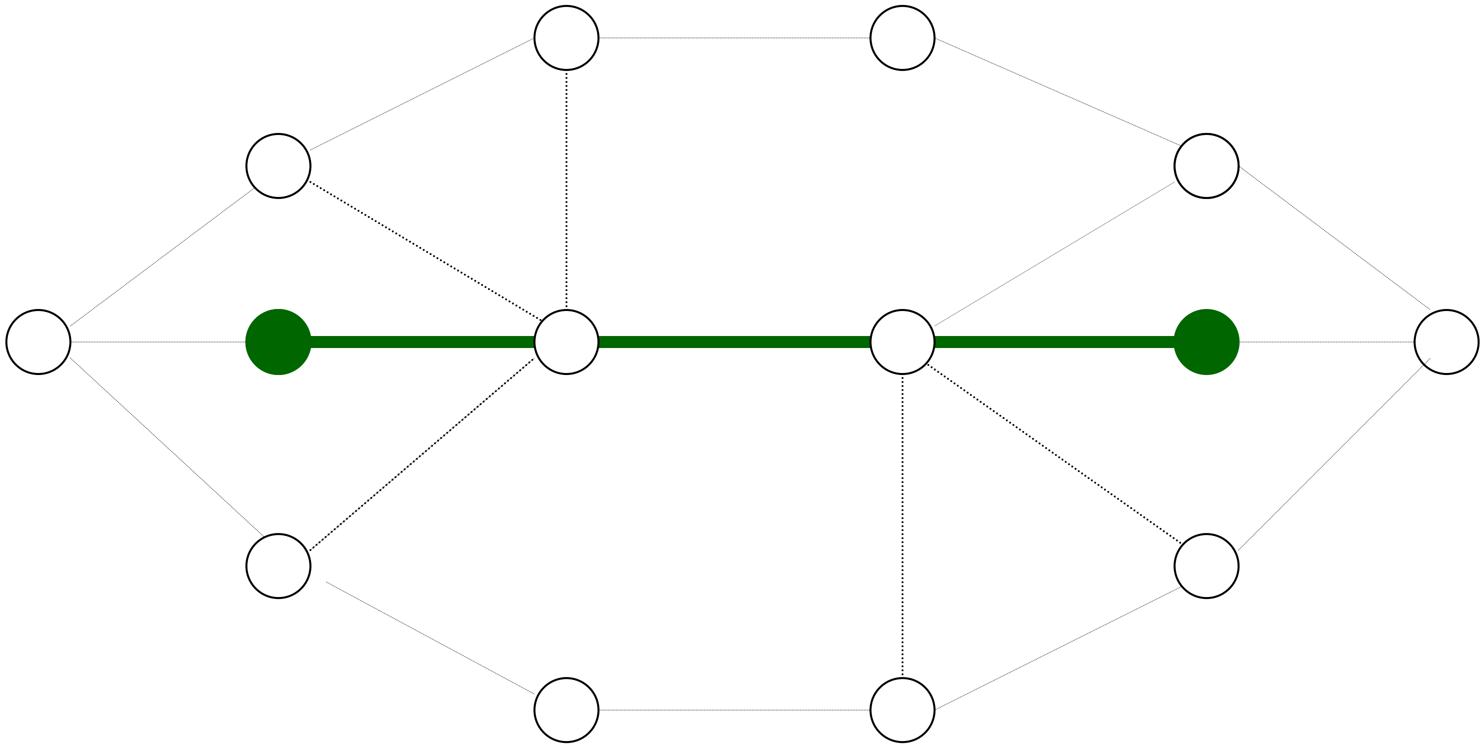
- Répartition du trafic sur multiples liens si les métriques sont identiques
- Détérminer les poids optimaux de façon à minimiser la saturation des liens
- Algorithme génétique hybride avec optimisation des croisements

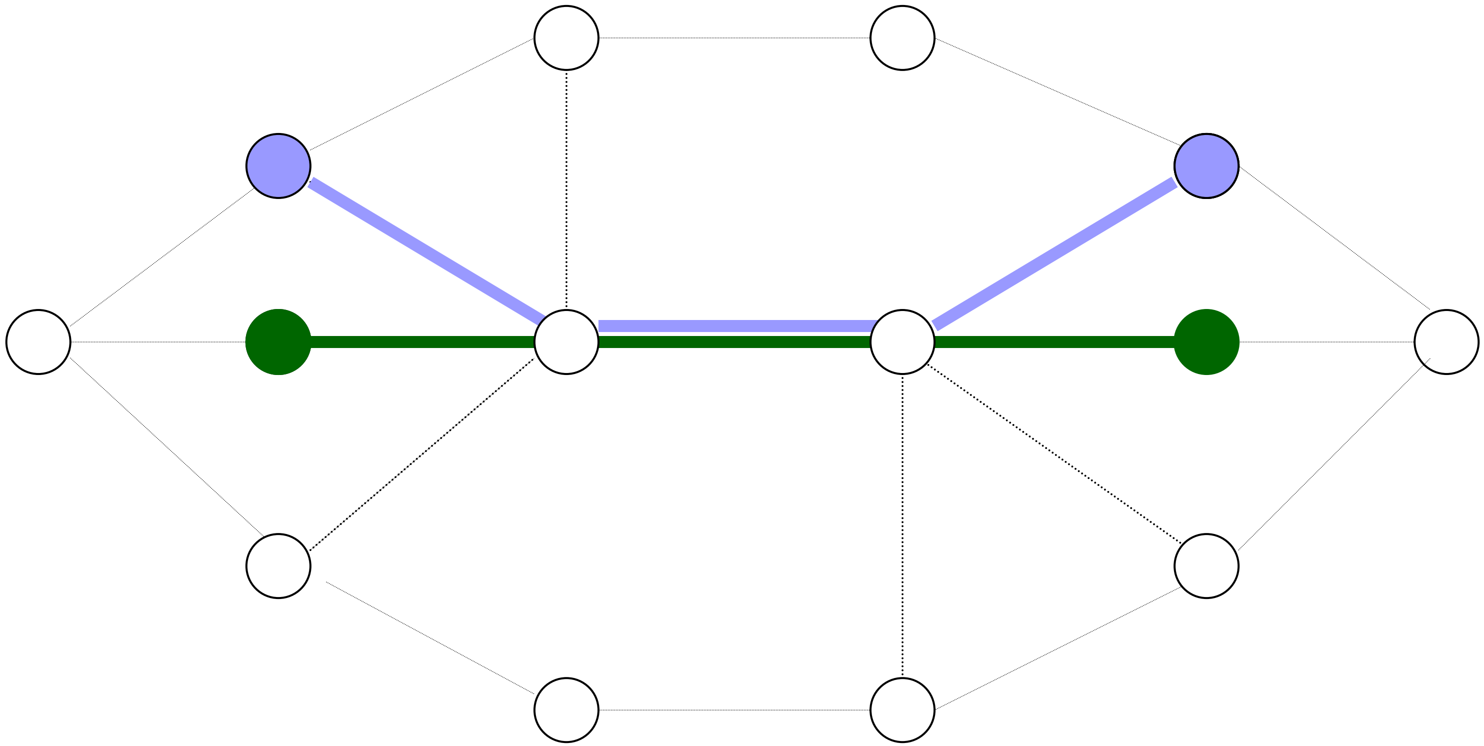

AT&T Worldnet backbone network (90 routers, 274 links)
GA
MA
cost
generation
LP lower bound
10000
1000
100
10
0
50
100
150
200
250
300
350
400
450
500

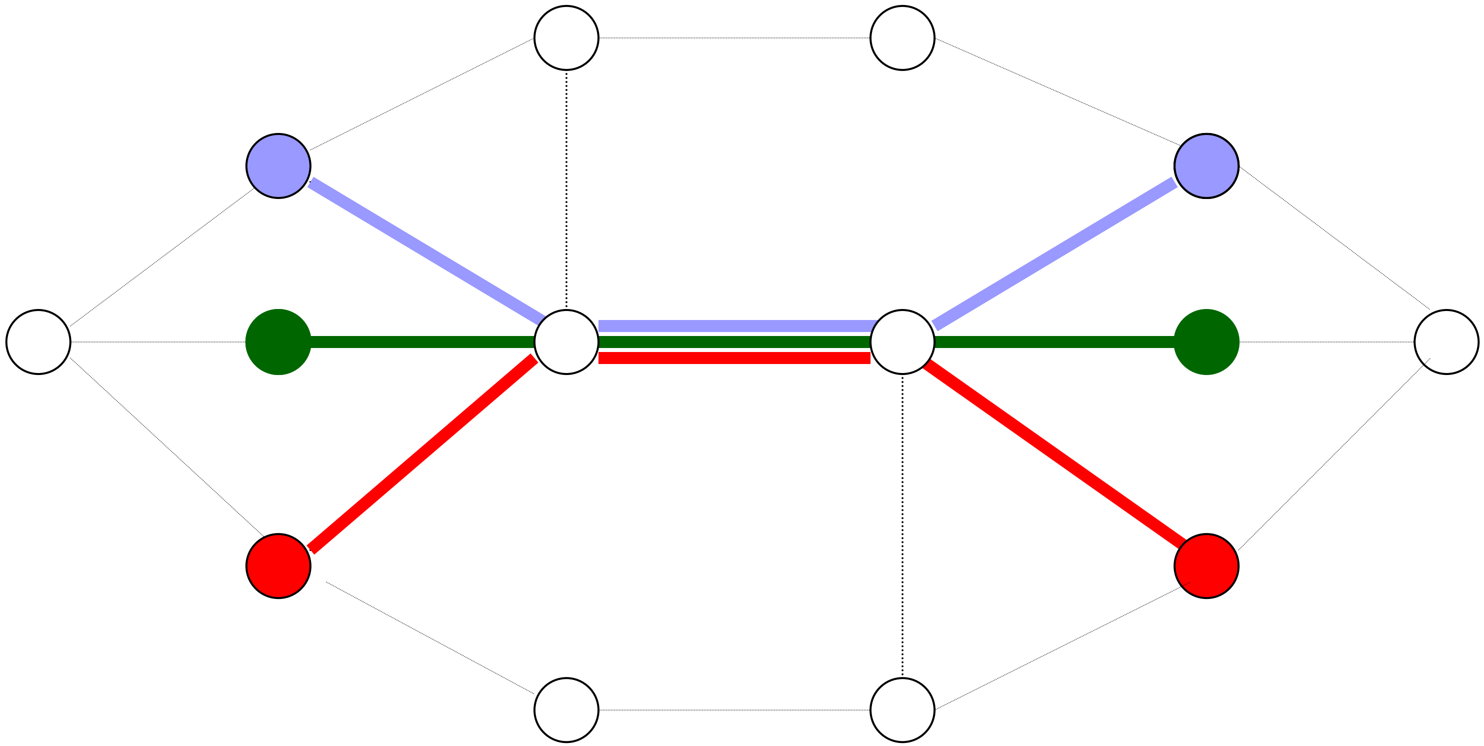
Routage des circuits virtuels

- Réseaux virtuels privés: circuits virtuels permanents (PVCs) entre chaque source et chaque destination de chaque client (la demande de chaque trafic est connue)
- Routage: détérminer les routes et les capacités affectées à chaque trafic de chaque client

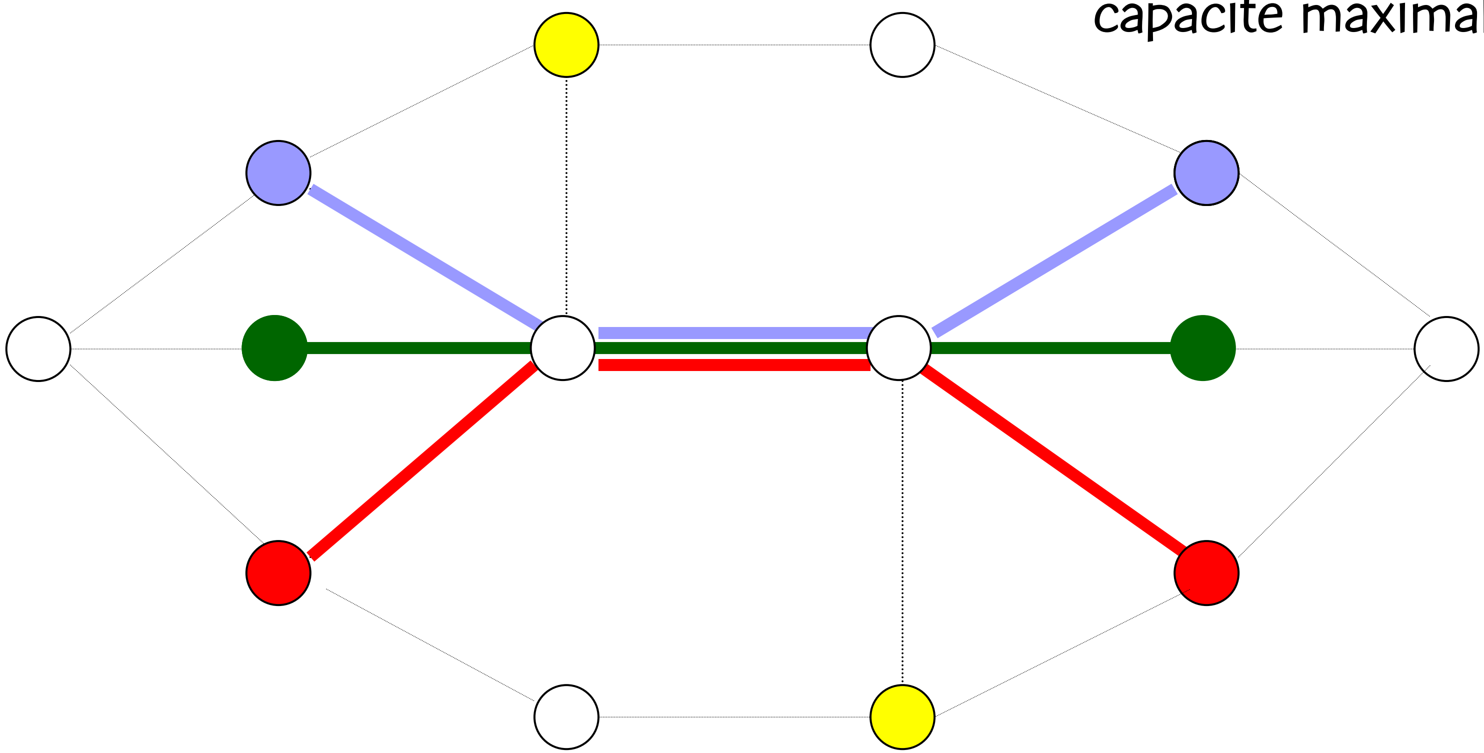


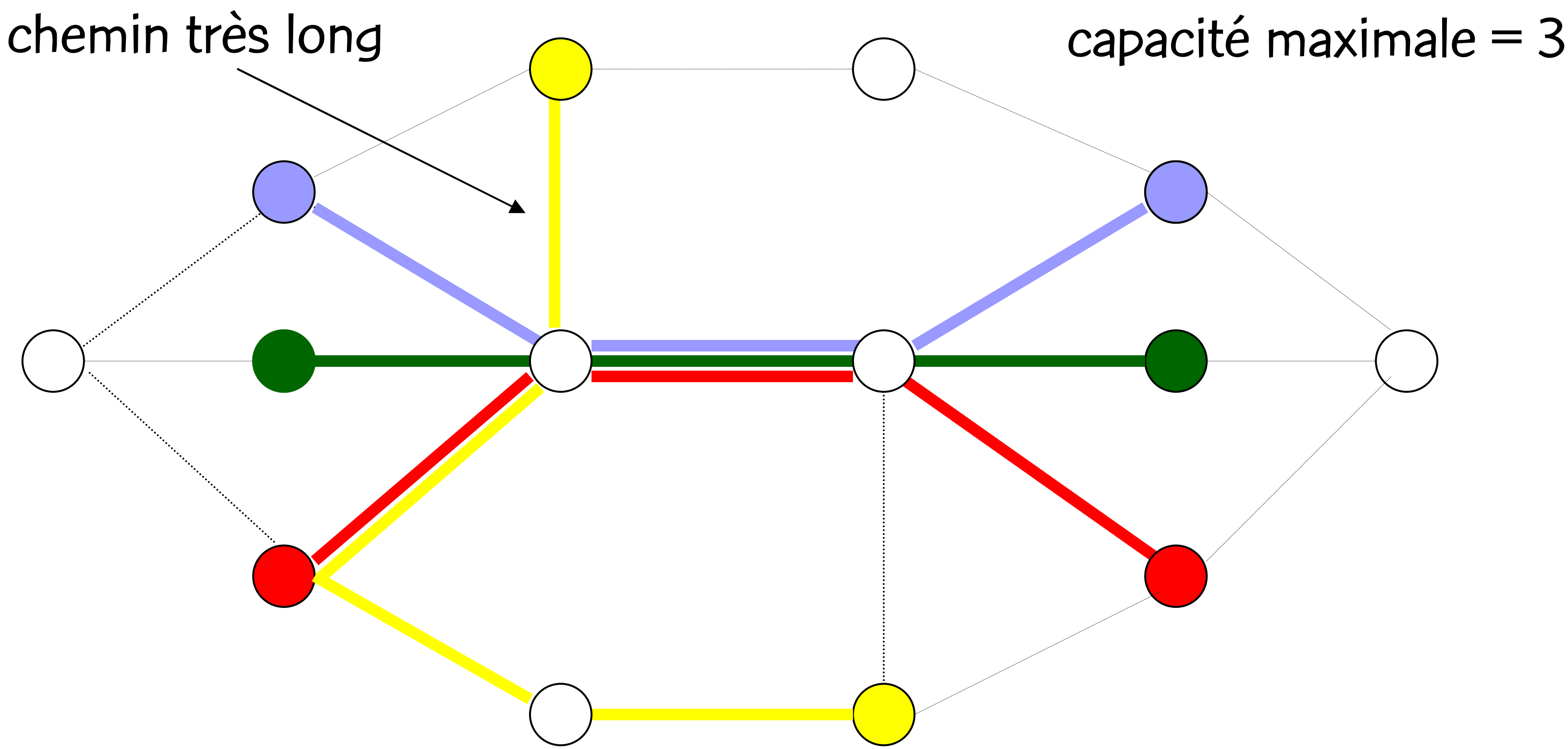


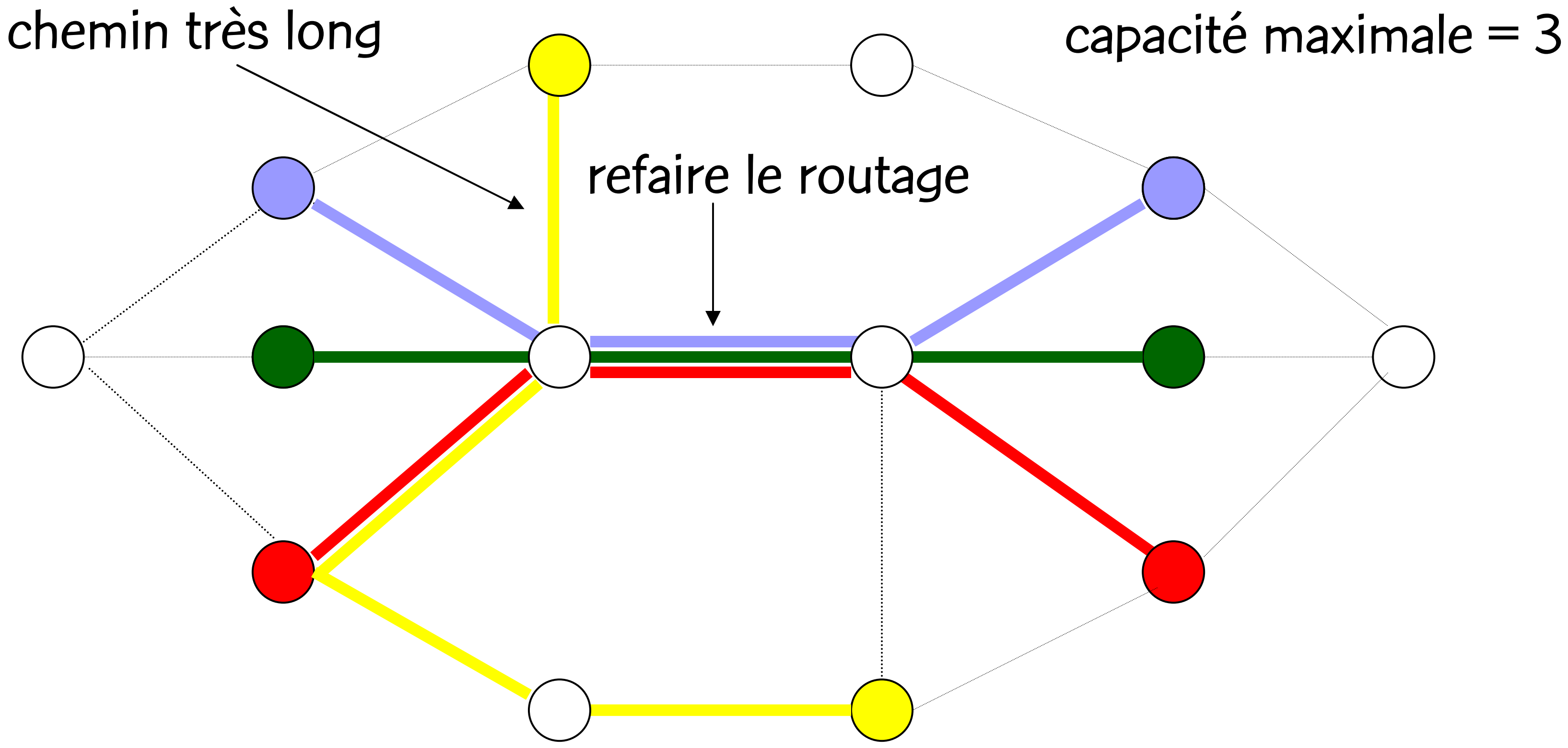


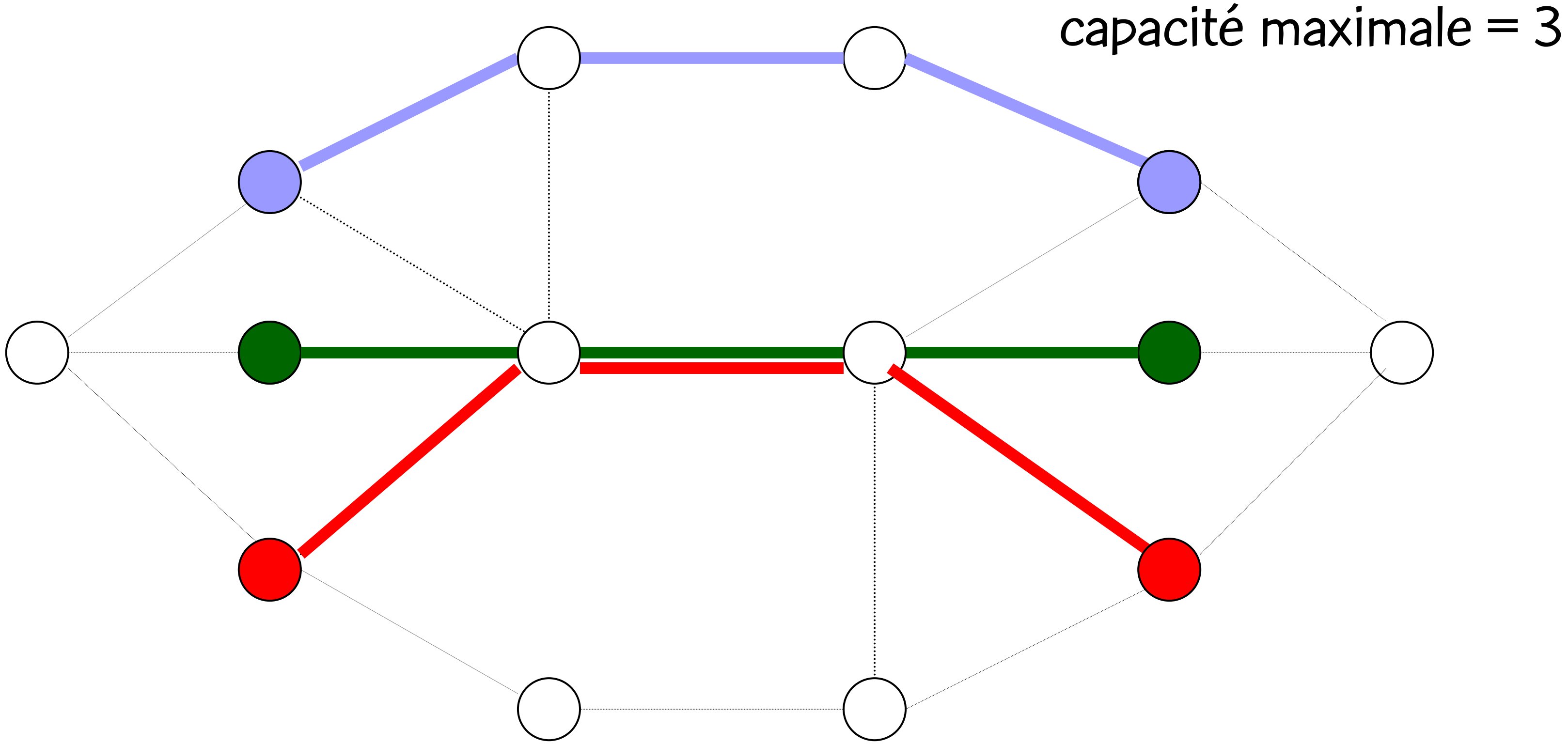


capacité maximale = 3



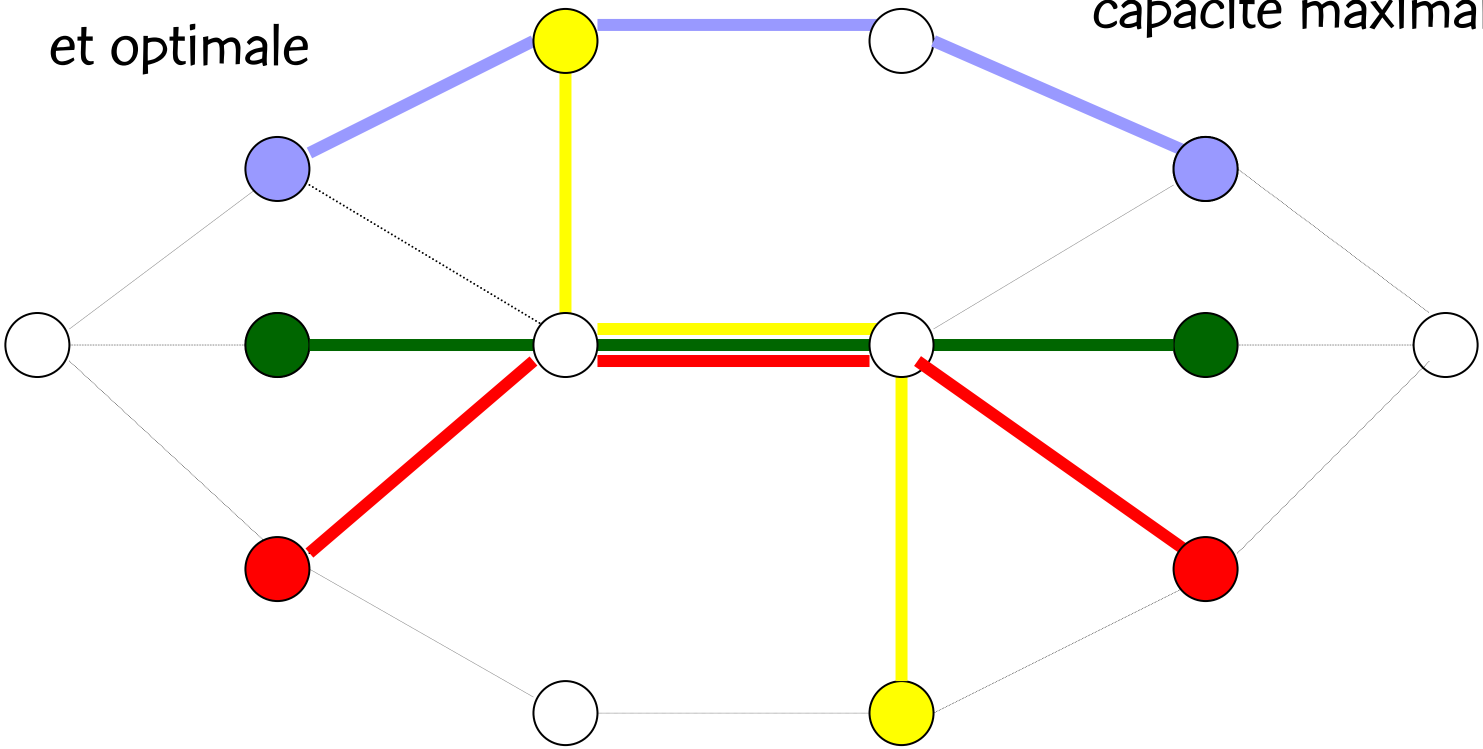

chemin très long
capacité maximale = 3


chemin très long
capacité maximale = 3
refaire le routage


capacité maximale = 3

solution réalisable
et optimale

capacité maximale = 3



Routage des circuits virtuels

- Algorithme GRASP pour routage optimal
- Outil pour l'afféctation optimale des trafics sur routes uniques dans les réseaux capacités: minimisation des délais avec équilibrage des charges
- Utilisation chez AT&T

Routage et afféctation des longueurs d’onde sur les réseaux optiques

- Ensemble de connexions: origine et destination connues pour chaque trafic
- Chaque connexion doit utiliser la même longueur d’onde de l’origine à la destination (pas de conversion)
- Plusieurs connexions peuvent utiliser la même longueur d’onde si elles n’ont pas d’arête commune

Routage et afféctation des longueurs d’onde sur les réseaux optiques

- Les connexions avec arêtes communes doivent utiliser des longueurs d’onde différentes
- Déterminer les routes associées à chaque connexion, ainsi que la longueur d’onde utilisée pour chaque connexion

Routage et afféctation des longueurs d'onde sur les réseaux optiques

- Différents critères d'optimisation, modèles de trafic et règles d'opération (conversions possibles ou non, etc.):
  - Pairs origine-destination connues
  - Pas de conversion
  - Arêtes bidiréctionnelles
  - Minimiser le nombre de longueurs d'onde utilisées

Connexions

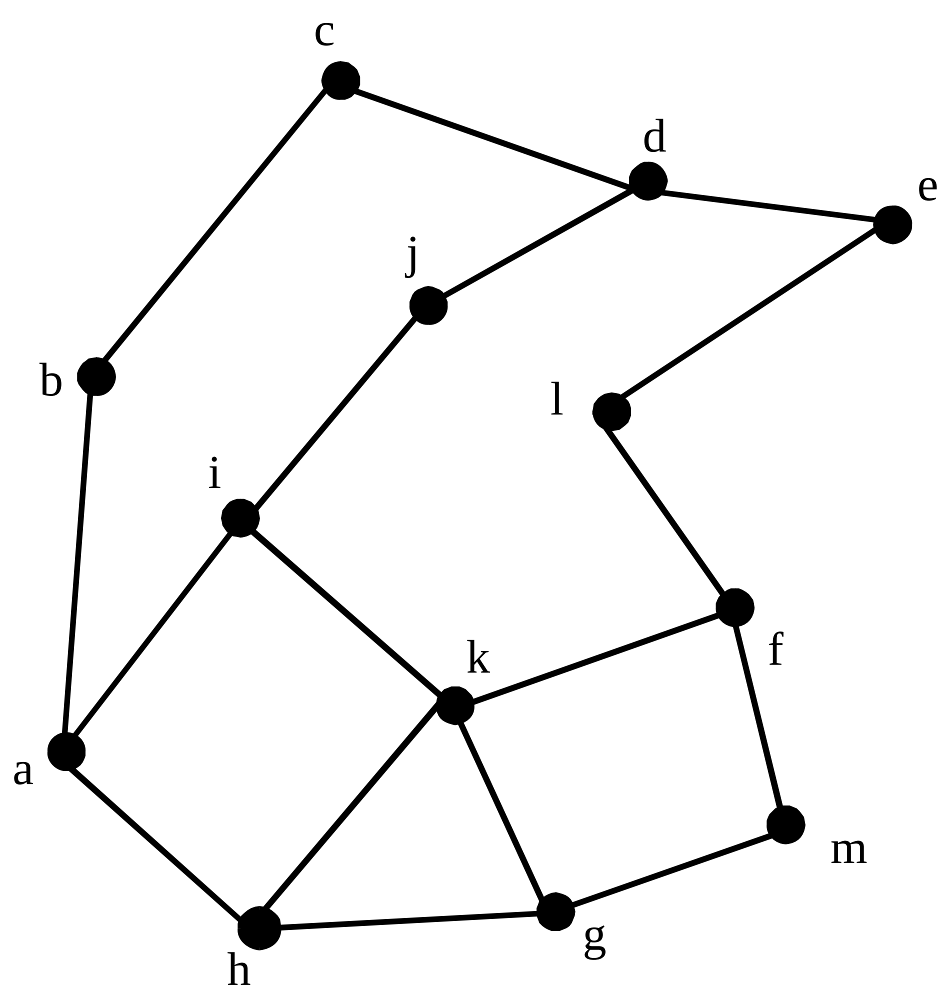
c ↔ m

d ↔ b

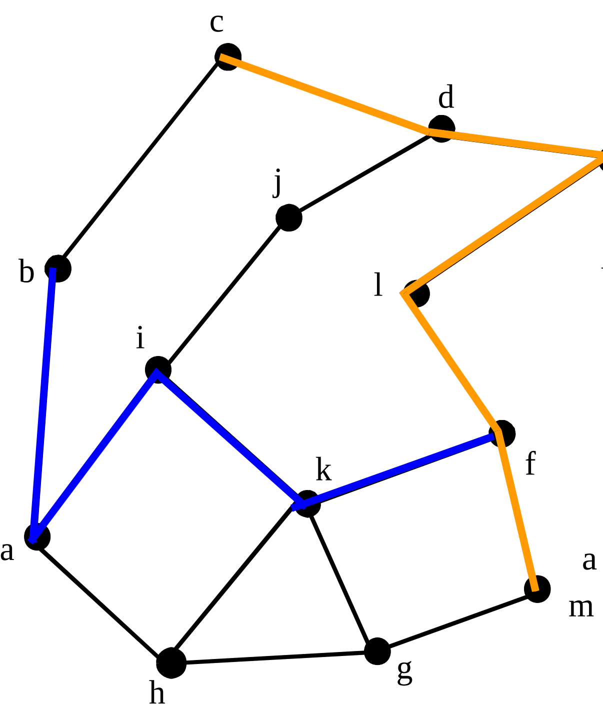
e ↔ h

a ↔ e

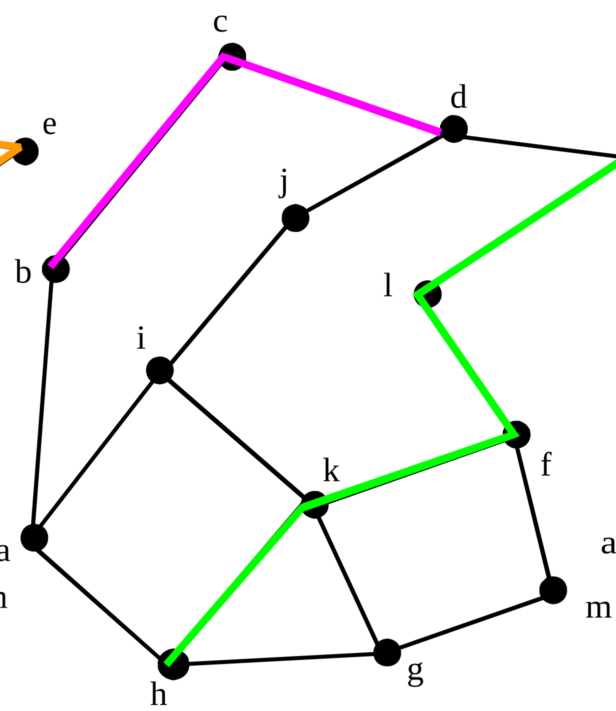
b ↔ f



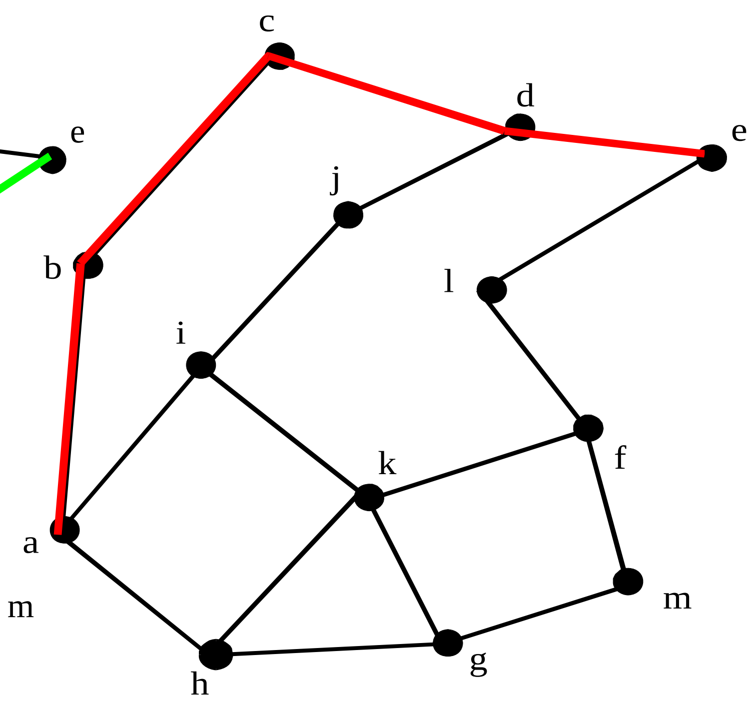
Connexions: (a ↔ e) (b ↔ f) (c ↔ m) (d ↔ b) (e ↔ h)



longueur d’onde 1



longueur d’onde 2



longueur d’onde 3

Routage et afféctation des longueurs d'onde sur les réseaux optiques

- Recherche tabou et algorithme génétique
- Extensions aux modèles plus complets, en tenant compte e.g. des contraintes de fiabilité et de la multiplicité de chemins

Synthèse des réseaux d’accès

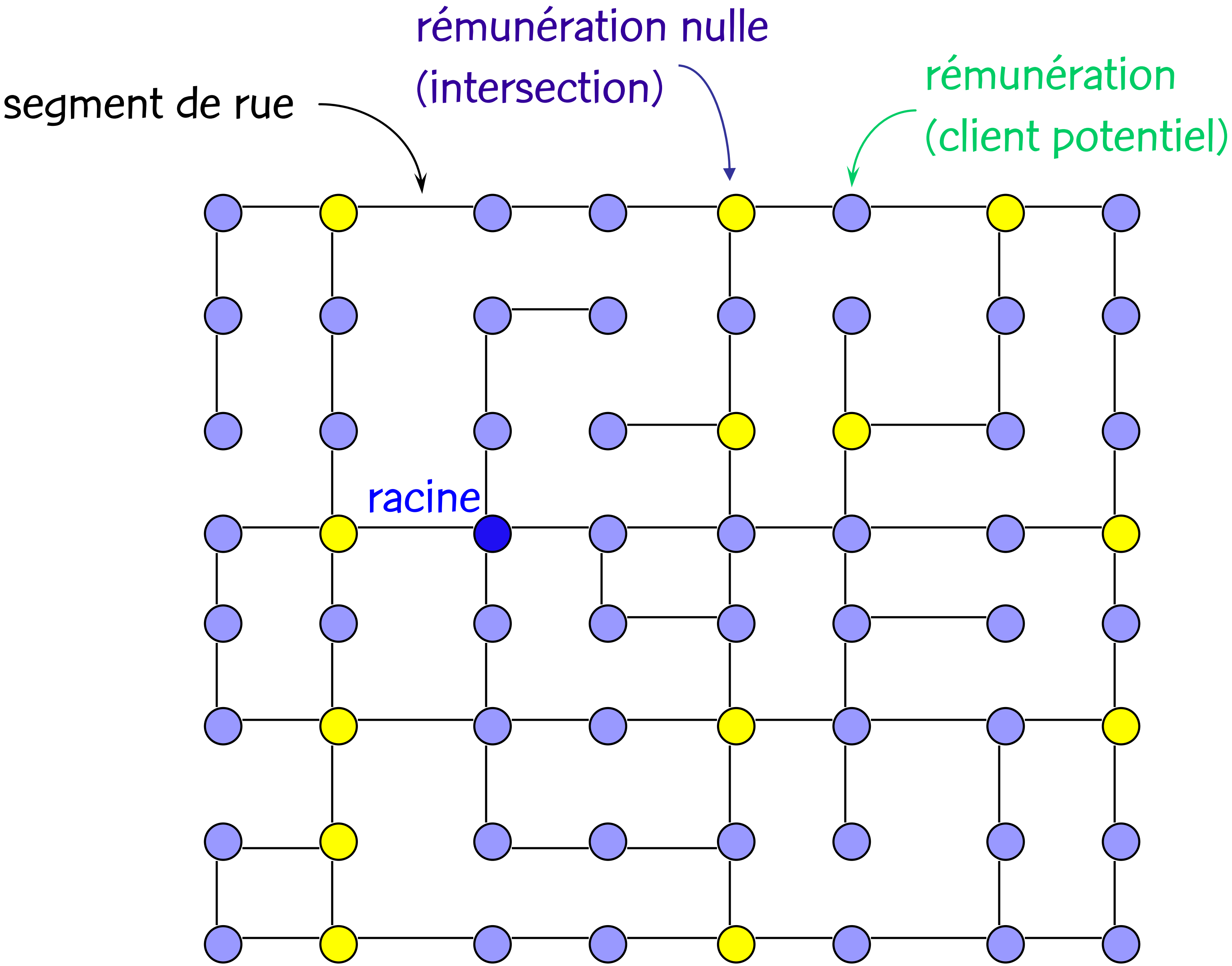
- Faire la synthèse d’un réseau optique pour donner accès aux communications à un ensemble de clients
- Prendre en considération le compromis entre:
  - coût de construction du réseau (installation des fibres optiques)
  - rémunération potentielle (ou perte de rémunération si un client n’est pas connecté)
- Choisir les clients qui seront connectés

Synthèse des réseaux d'accès

- Choisir les clients qui seront connectés
- Problème de Steiner avec récolte de primes
  - Prime: rémuneration correspondante à un client
- Application développée pour AT&T

Synthèse des réseaux d'accès

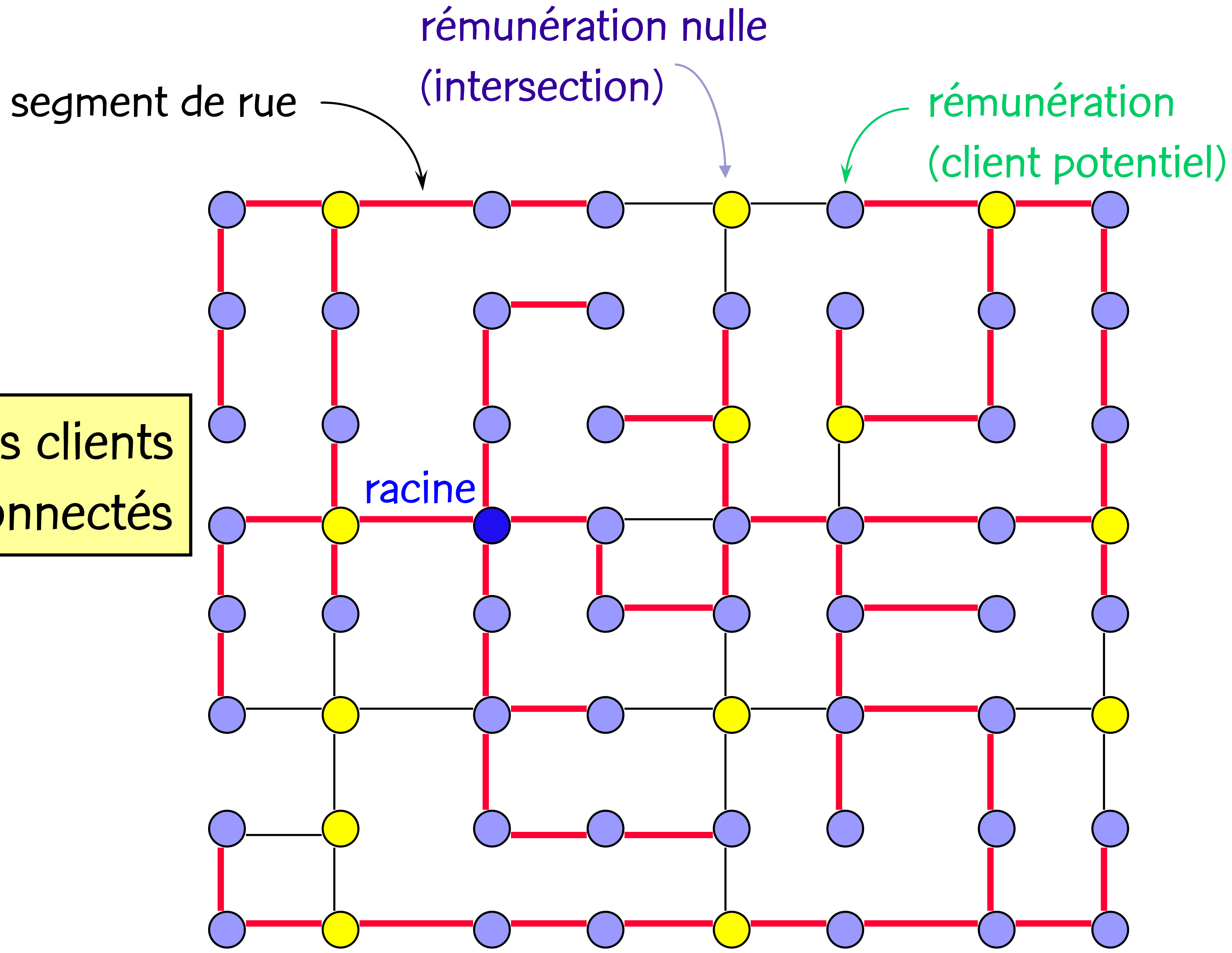
- Graphe: rues et intersections
  - Arêtes: segments de rues
    - Coût d'une arête: installation de fibres optiques sur chaque segment de rue
  - Sommets: intersections des rues et locaux des clients potentiels
    - Prime d'un sommet: rémunération correspondante à chaque client (pas de prime pour les intersections de rues)
- Minimiser une mesure de la somme des coûts de construction (fibres installées) et des primes non-recoltées (clients non-connectés)



segment de rue
rémunération nulle
(intersection)
rémunération
(client potentiel)
racine

Synthèse des réseaux d'accès

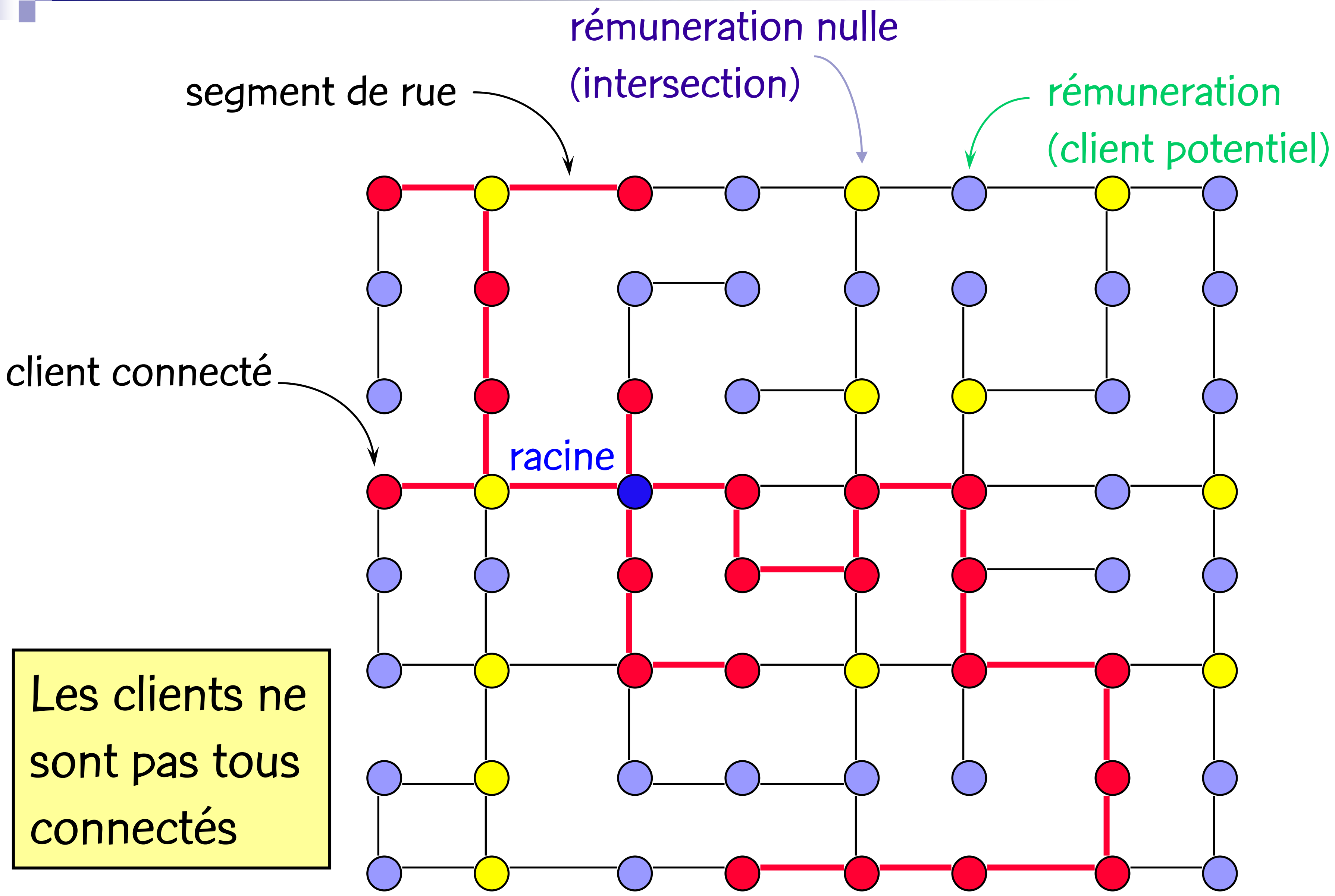
- Premier cas: tous les clients doivent être connectés
- Problème de Steiner:
  - Déterminer les connexions de façon à minimiser les coûts de construction



Tous les clients sont connectés

Synthèse des réseaux d'accès

- Deuxième cas: choisir les clients qui seront connectés
- Problème de Steiner avec primes:
  - Déterminer les connexions de façon à minimiser les coûts de construction et les primes non-recoltées



Les clients ne sont pas tous connectés

Synthèse des réseaux d'accès

- Algorithme GRASP
- Taille typique des problèmes réels:
  - 20.000 à 100.000 sommets

Recherche en cours: applications

- Réseaux ad-hoc: communication directe entre les sommets sans structure fixe
- Réséaux de senseurs: noeuds (avec batteries de capacité limitée) capables de transmettre les informations obtenues sur un ou plusieurs phénomènes
- Puissance nécéssaire pour la réception au sommet j d'un message envoyé du sommet i: $p_{ij} \sim d_{ij}^2$

Recherche en cours: applications

- Messages doivent arriver à tous les noeuds
- Sessions: unique vs. multiples
- On-line vs. statique
- Déterminer la puissance de chaque émetteur:
  - Minimiser la consommation d'énergie
  - Minimiser la consommation d'énergie en tenant compte des capacités des batteries
  - Assurer une certaine topologie (k chemins disjoints entre chaque paire de noeuds)