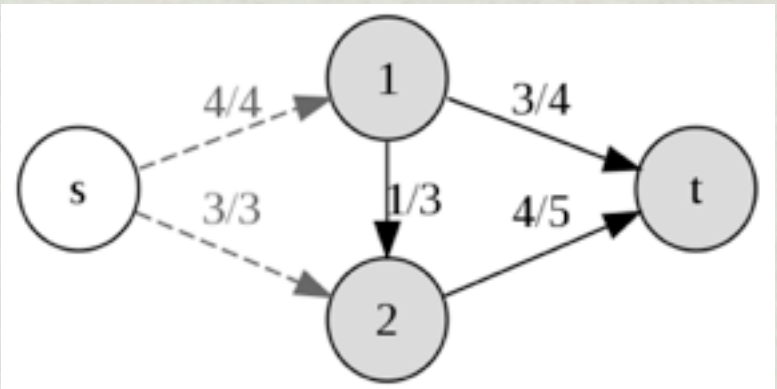
Programmation linéaire
Optimisation combinatoire

Hervé Rivano

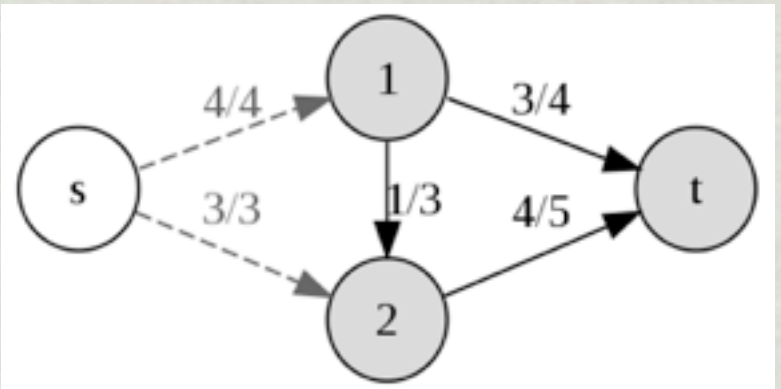
30 juin 2010

Router dans un réseau

- Un graphe :
  - V sommets, E arcs
  - c capacité
- Ensemble (source,destination)
- Router le plus de trafic possible
  - Respect des capacité



Flot maximum



Flot maximum

$$Max\ d$$

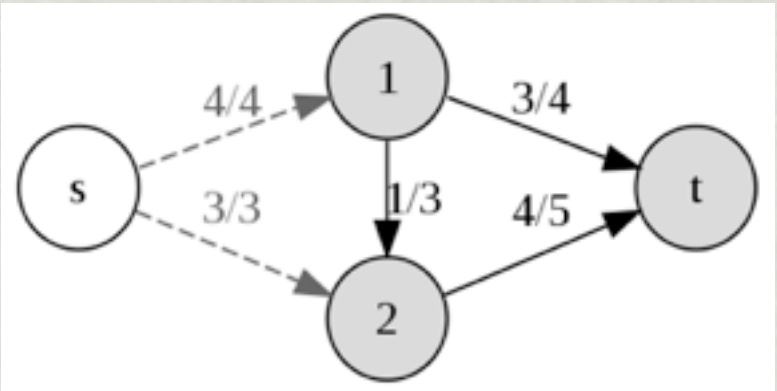
$$s.t.$$

$$\sum_v f(v,s) + d - \sum_v f(s,v) = 0$$

$$\sum_v f(v,t) - d - \sum_v f(t,v) = 0$$

$$\sum_v f(v,u) - \sum_v f(u,v) = 0,\ \forall u$$

$$f(u,v) \leq c(u,v),\ \forall (u,v)$$



Programmation linéaire

- L’optimisation convexe
- La programmation linéaire
- La programmation linéaire en nombre entiers
- Dualité
- Génération de colonnes

Optimisation convexe

$$Max\ f(z)$$

$$s.t.$$

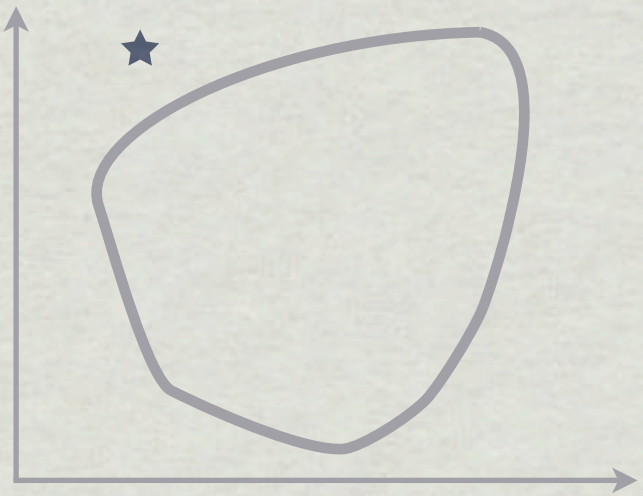
$$z \in \mathcal{C}$$

- Optimiser, c'est possible
  - Newton, ...
- Optimiser et décider, c'est «pareil»
  - Dichotomie

Séparation

Soit $\mathcal{C}$ un convexe fermé et $z \notin \mathcal{C}$

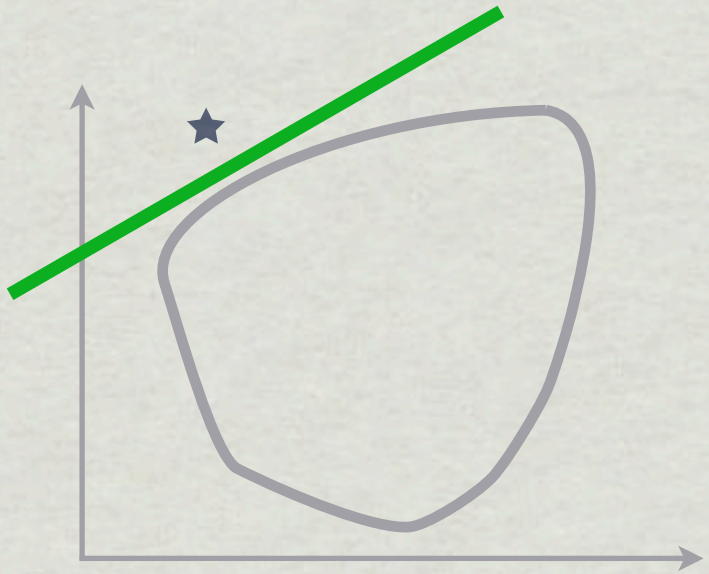
$\exists\ \mathcal{H}$ hyperplan séparant z et $\mathcal{C}$



Séparation

Soit $\mathcal{C}$ un convexe fermé et $z \notin \mathcal{C}$

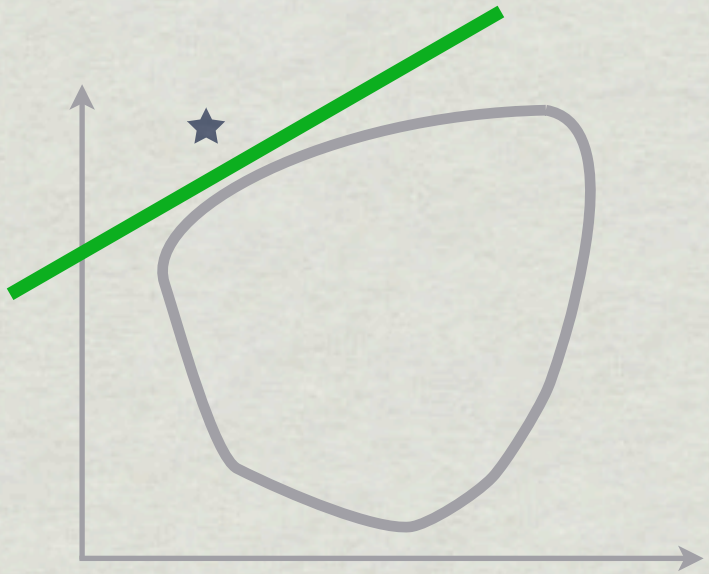
$\exists\ \mathcal{H}$ hyperplan séparant z et $\mathcal{C}$



Séparation

Soit $\mathcal{C}$ un convexe fermé et $z \notin \mathcal{C}$

$\exists\ \mathcal{H}$ hyperplan séparant z et $\mathcal{C}$

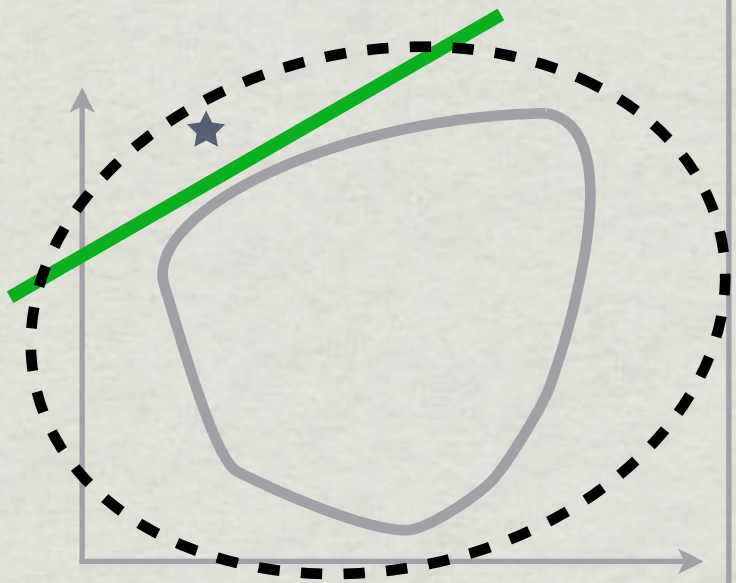


- Séparer = Optimiser
  - Algorithme de l'ellipsoïde
  - Généralisation de la dichotomie

Séparation

Soit $\mathcal{C}$ un convexe fermé et $z \notin \mathcal{C}$

$\exists\ \mathcal{H}$ hyperplan séparant z et $\mathcal{C}$

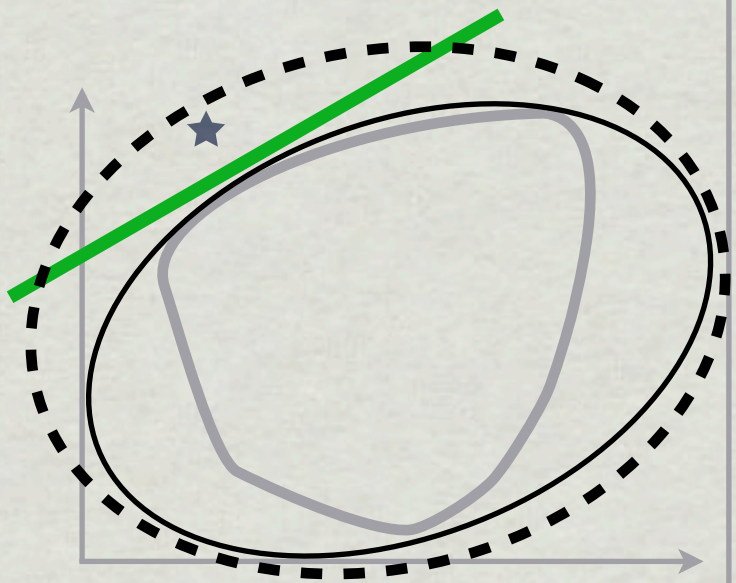


- Séparer = Optimiser
  - Algorithme de l'ellipsoïde
  - Généralisation de la dichotomie

Séparation

Soit $\mathcal{C}$ un convexe fermé et $z \notin \mathcal{C}$

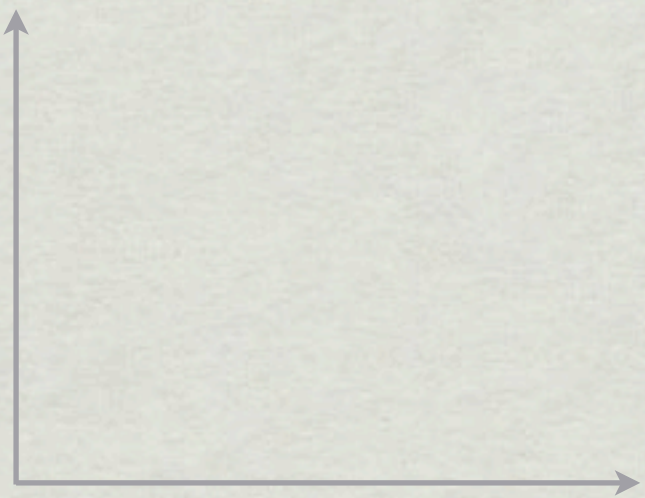
$\exists\ \mathcal{H}$ hyperplan séparant z et $\mathcal{C}$



- Séparer = Optimiser
  - Algorithme de l'ellipsoïde
  - Généralisation de la dichotomie

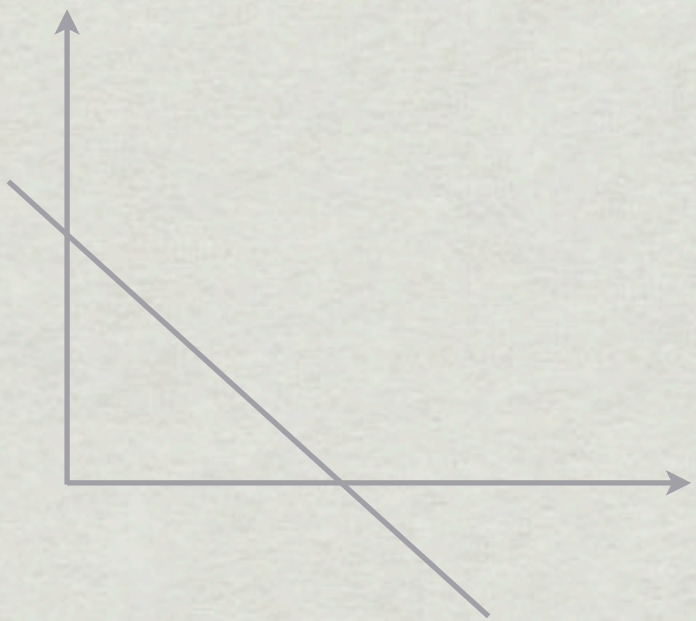
Programmation linéaire

- Forme particulière de programmation convexe
- Polytope défini par l'intersection d'hyperplan
- Optimisation d'une forme linéaire
- Points extrémaux
  - Algorithme du simplex



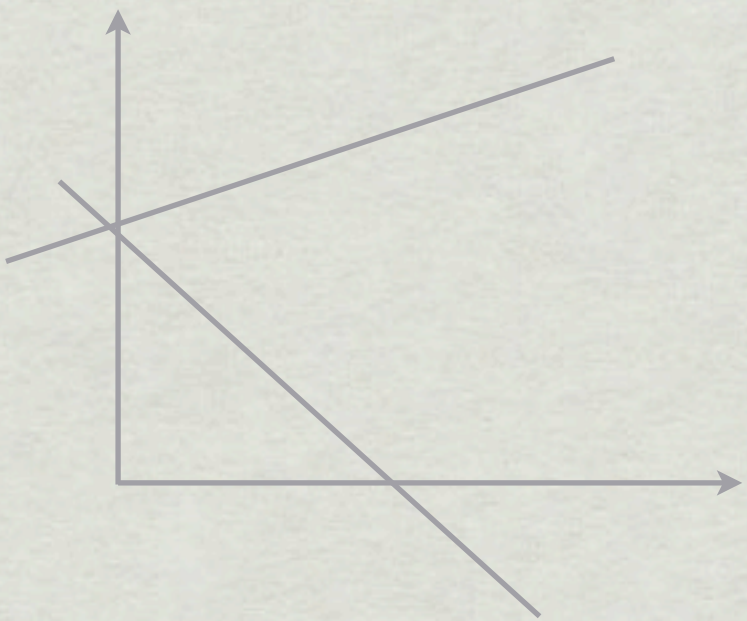
Programmation linéaire

- Forme particulière de programmation convexe
- Polytope défini par l'intersection d'hyperplan
- Optimisation d'une forme linéaire
- Points extrémaux
  - Algorithme du simplex



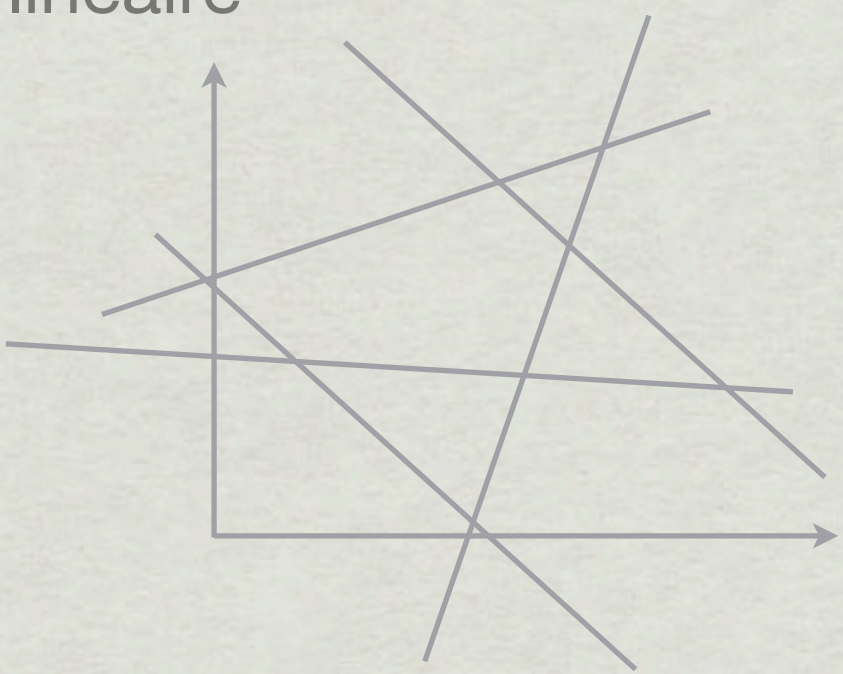
Programmation linéaire

- Forme particulière de programmation convexe
- Polytope défini par l'intersection d'hyperplan
- Optimisation d'une forme linéaire
- Points extrémaux
  - Algorithme du simplex



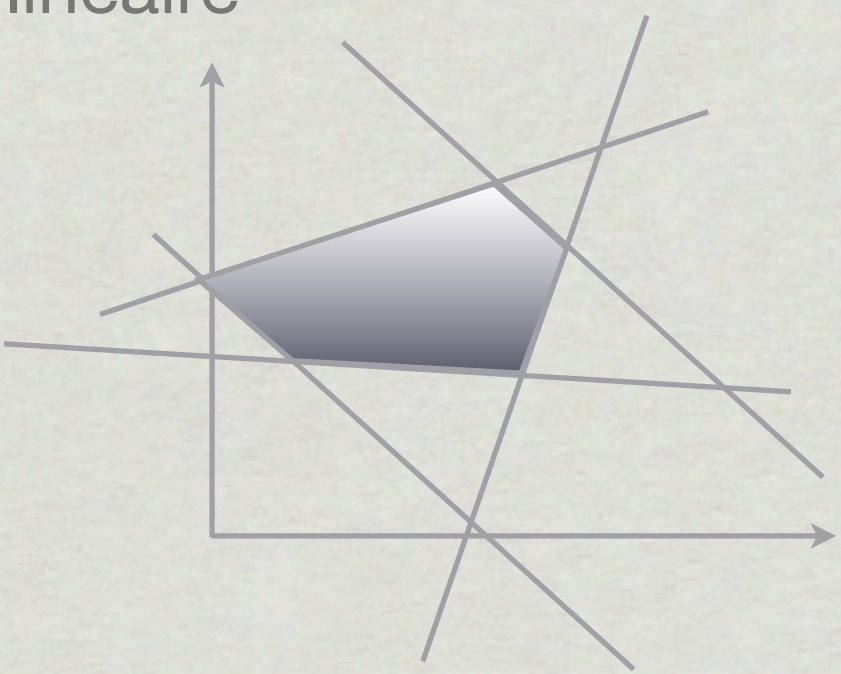
Programmation linéaire

- Forme particulière de programmation convexe
- Polytope défini par l'intersection d'hyperplan
- Optimisation d'une forme linéaire
- Points extrémaux
  - Algorithme du simplex



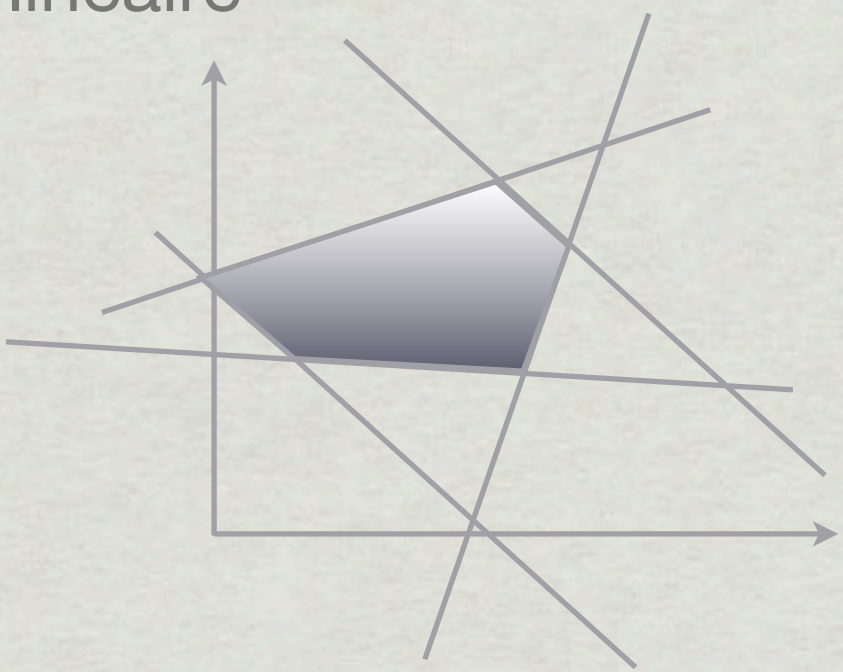
Programmation linéaire

- Forme particulière de programmation convexe
- Polytope défini par l'intersection d'hyperplan
- Optimisation d'une forme linéaire
- Points extrémaux
  - Algorithme du simplex



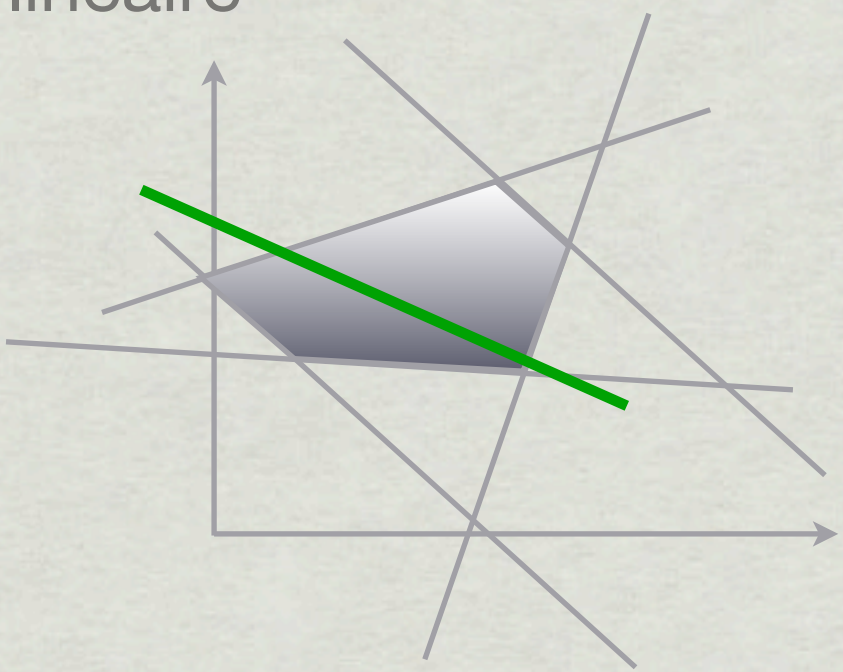
Programmation linéaire

- Forme particulière de programmation convexe
- Polytope défini par l'intersection d'hyperplan
- Optimisation d'une forme linéaire



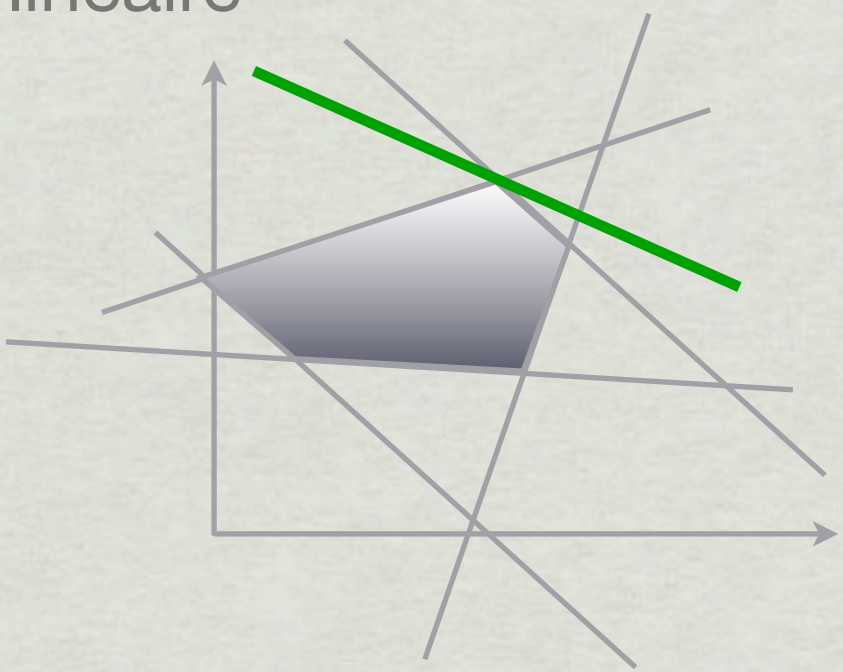
Programmation linéaire

- Forme particulière de programmation convexe
- Polytope défini par l'intersection d'hyperplan
- Optimisation d'une forme linéaire



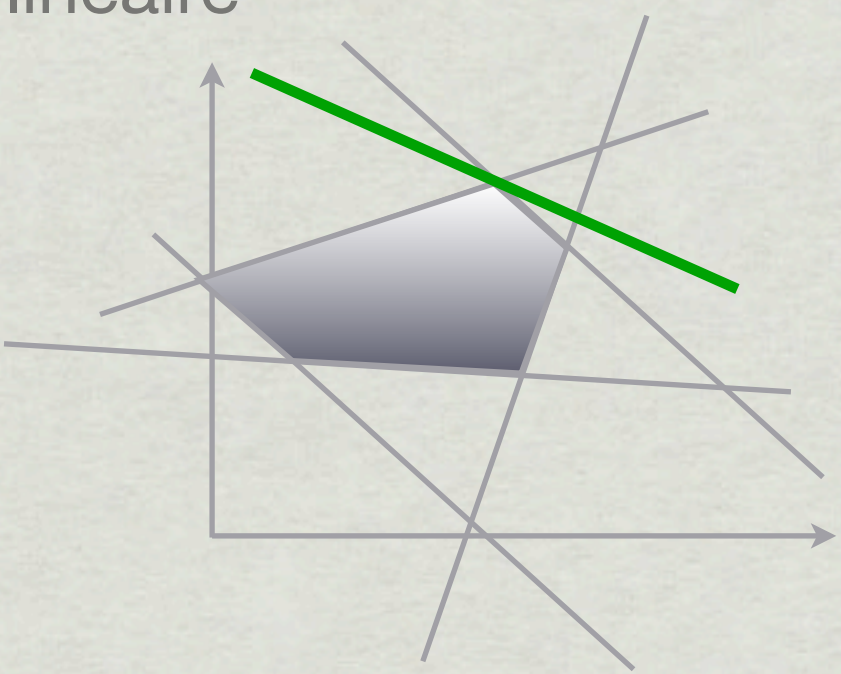
Programmation linéaire

- Forme particulière de programmation convexe
- Polytope défini par l'intersection d'hyperplan
- Optimisation d'une forme linéaire



Programmation linéaire

- Forme particulière de programmation convexe
- Polytope défini par l'intersection d'hyperplan
- Optimisation d'une forme linéaire
- Points extrémaux
  - Algorithme du simplex



Ecriture matricielle

$$Max \sum_{p \in \mathcal{P}} f(p)$$

$$s.t.$$

$$\sum_{p \ni e} f(p) \leq c(e), \ \forall e$$

$$\sum_{p \in \mathcal{P}_u} f(p) \geq d(u), \ \forall u$$

$$f(p) \geq 0, \ \forall p$$

Ecriture matricielle

$$Max \sum_{p \in \mathcal{P}} f(p)$$

$$s.t.$$

$$\sum_{p \ni e} f(p) \leq c(e), \ \forall e$$

$$\sum_{p \in \mathcal{P}_u} f(p) \geq d(u), \ \forall u$$

$$f(p) \geq 0, \ \forall p$$

$$Max \ c.x$$

$$s.t.$$

$$A.x \leq b$$

Ecriture matricielle

$$Max \sum_{p \in \mathcal{P}} f(p)$$

$$s.t.$$

$$\sum_{p \ni e} f(p) \leq c(e), \ \forall e$$

$$\sum_{p \in \mathcal{P}_u} f(p) \geq d(u), \ \forall u$$

$$f(p) \geq 0, \ \forall p$$

$$Max \ c.x$$

$$s.t.$$

$$A.x \leq b$$

$$x \in \mathbb{R}^n$$

$$c \in \mathbb{R}^n$$

$$b \in \mathbb{R}^m$$

$$A \in \mathbb{R}^{m \times n}$$

Ecriture matricielle

$$Max \sum_{p \in \mathcal{P}} f(p)$$

$$s.t.$$

$$\sum_{p \ni e} f(p) \leq c(e), \ \forall e$$

$$\sum_{p \in \mathcal{P}_u} f(p) \geq d(u), \ \forall u$$

$$f(p) \geq 0, \ \forall p$$

$$Max \ c.x$$

$$s.t.$$

$$A.x \leq b$$

$$x \in \mathbb{R}^n$$

$$c \in \mathbb{R}^n$$

$$b \in \mathbb{R}^m$$

$$A \in \mathbb{R}^{m \times n}$$

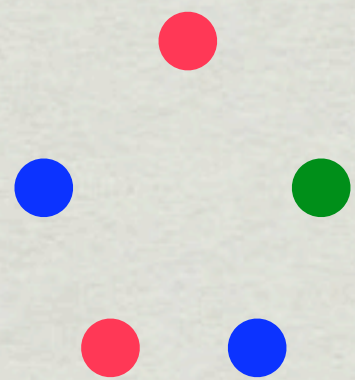
$$Min \ c_2.x$$

$$s.t.$$

$$A_2.x \geq b_2$$

Prog. Lin. en nb entiers

- Si les variables sont toutes dans Z : ILP
- Si certaines sont dans Z, les autres dans R : MILP
- En général NP-difficile, non approximable



Prog. Lin. en nb entiers

- Si les variables sont toutes dans Z : ILP
- Si certaines sont dans Z, les autres dans R : MILP
- En général NP-difficile, non approximable

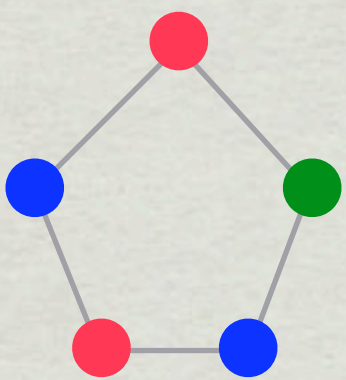
$$Min \sum_C y(c)$$

$$s.t.$$

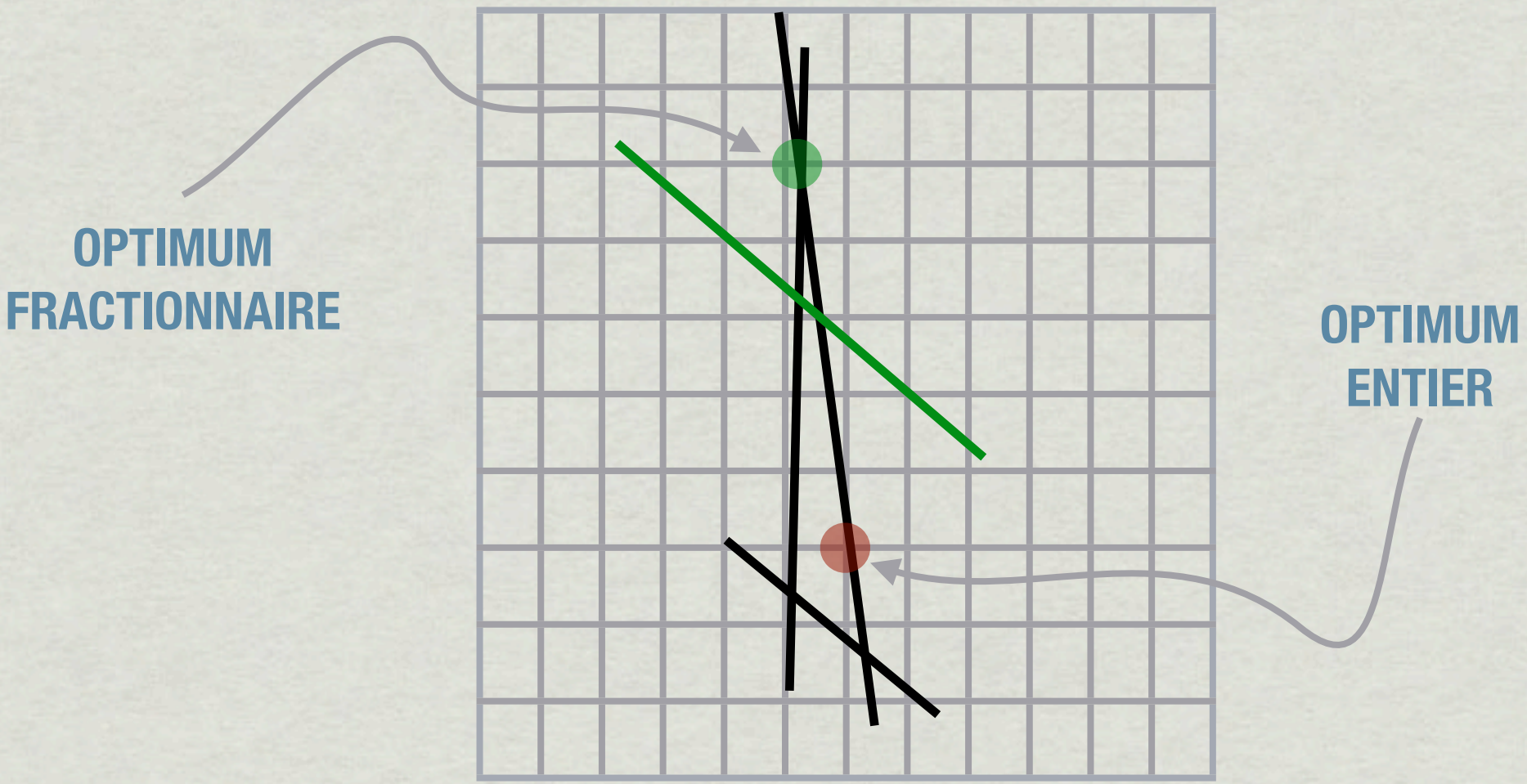
$$\sum_c x_c(u) \geq 1, \ \forall u$$

$$x_c(u) + x_c(v) \leq 1, \ \forall c, u, v$$

$$y(c) - x_c(u) \geq 0, \ \forall c, u$$



Intuition de la difficulté



Dualité

- Posons le problème de décision

$$\exists ?x \ s.t.$$
$$c.x \geq C \ \& \ Ax \leq b$$

Dualité

- Posons le problème de décision

$$\exists ?x \ s.t.$$
$$c.x \geq C \ \& \ Ax \leq b$$

- Cherchons à prouver que non

$$\forall x,$$
$$Ax \leq b \Rightarrow c.x \leq C$$

Dualité

- Posons le problème de décision
- Cherchons à prouver que non
- C'est vrai si

$$\exists ?x \ s.t. \\ c.x \geq C \ \& \ Ax \leq b$$

$$\forall x, \\ Ax \leq b \Rightarrow c.x \leq C$$

$$\exists y \ s.t. \ \forall x \\ c.x \leq y.A.x \ \& \ y.b \leq C$$

Dualité

- Posons le problème de décision

$$\exists ?x \ s.t. \\ c.x \geq C \ \& \ Ax \leq b$$

- Cherchons à prouver que non

$$\forall x, \\ Ax \leq b \Rightarrow c.x \leq C$$

- C'est vrai si

$$\exists y \ s.t. \ \forall x \\ c.x \leq y.A.x \ \& \ y.b \leq C$$

- Autrement dit

$$\exists y \ s.t. \\ {}^{t}At \geq c \ \& \ b.y \leq C$$

Dualité

- Posons le problème de décision

$$\exists ?x \ s.t. \\ c.x \geq C \ \& \ Ax \leq b$$

- Cherchons à prouver que non

$$\forall x, \\ Ax \leq b \Rightarrow c.x \leq C$$

- C'est vrai si

$$\exists y \ s.t. \ \forall x \\ c.x \leq y.A.x \ \& \ y.b \leq C$$

- Autrement dit

$$\exists y \ s.t. \\ {}^{t}At \geq c \ \& \ b.y \leq C$$

- Conséquence

Dualité

- Posons le problème de décision

$$\exists ?x \ s.t. \\ c.x \geq C \ \& \ Ax \leq b$$

- Cherchons à prouver que non

$$\forall x, \\ Ax \leq b \Rightarrow c.x \leq C$$

- C'est vrai si

$$\exists y \ s.t. \ \forall x \\ c.x \leq y.A.x \ \& \ y.b \leq C$$

- Autrement dit

$$\exists y \ s.t. \\ {}^{t}At \geq c \ \& \ b.y \leq C$$

- Conséquence $Max\left\{c.x \ s.t. \ Ax \leq b\right\} \leq Min\left\{b.y \ s.t. \ {}^{t}Ay \geq c\right\}$

Retour au flot maximum

$$Max\ d$$

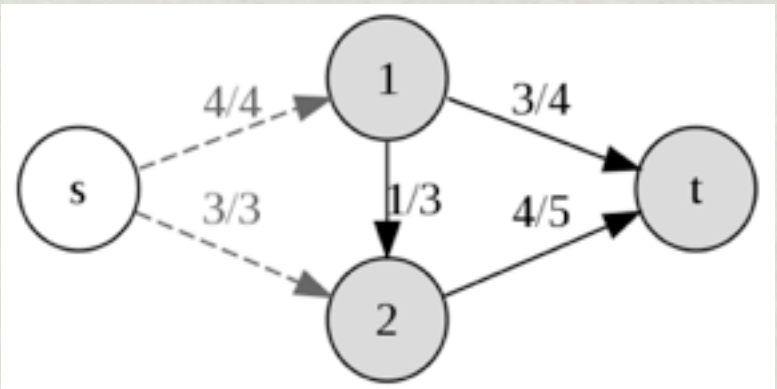
$$s.t.$$

$$\sum_v f(v,s) + d - \sum_v f(s,v) = 0$$

$$\sum_v f(v,t) - d - \sum_v f(t,v) = 0$$

$$\sum_v f(v,u) - \sum_v f(u,v) = 0,\ \forall u$$

$$f(u,v) \leq c(u,v),\ \forall (u,v)$$



Trichons un peu

$$Max\ d$$

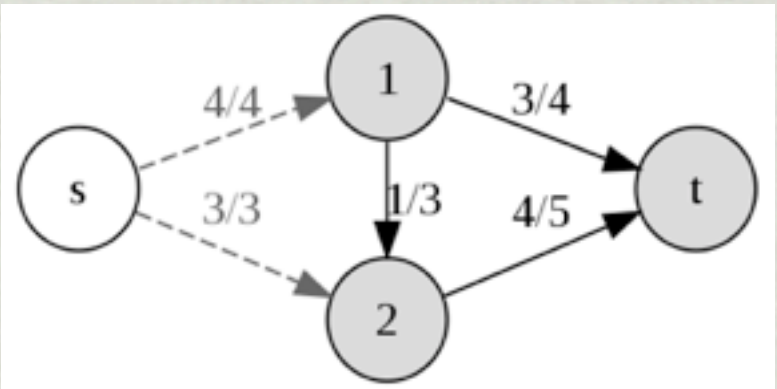
$$s.t.$$

$$\sum_v f(v,s) + d - \sum_v f(s,v) \leq 0$$

$$\ldots$$

$$\sum_v f(v,u) - \sum_v f(u,v) \leq 0,\ \forall u$$

$$f(u,v) \leq c(u,v),\ \forall (u,v)$$



$$Min \sum_e \lambda(e)c(e)$$

$$s.t.$$

$$p_s \geq 1$$

$$\lambda(u,v) - p_u + p_v \geq 0,\ \forall (u,v)$$

$$p_t = 0$$

Trichons un peu

$$Max\ d$$

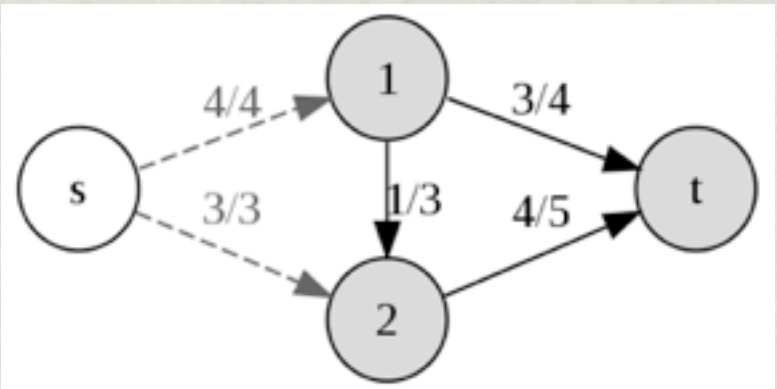
$$s.t.$$

$$\sum_v f(v,s) + d - \sum_v f(s,v) \leq 0$$

$$\ldots$$

$$\sum_v f(v,u) - \sum_v f(u,v) \leq 0,\ \forall u$$

$$f(u,v) \leq c(u,v),\ \forall (u,v)$$



$$Min \sum_e \lambda(e)c(e)$$

$$s.t.$$

$$p_s \geq 1$$

$$\lambda(u,v) - p_u + p_v \geq 0,\ \forall (u,v)$$

$$p_t = 0$$

DUAL : COUPE MINIMUM

A quoi ça sert ?

- Le dual donne une borne supérieure
  - Algorithmes primal/dual : garantie sur la solution

A quoi ça sert ?

- Le dual donne une borne supérieure
  - Algorithmes primal/dual : garantie sur la solution
- Pb de décision : primal plein = dual vide
  - Solution sous-optimale = contrainte duale violée

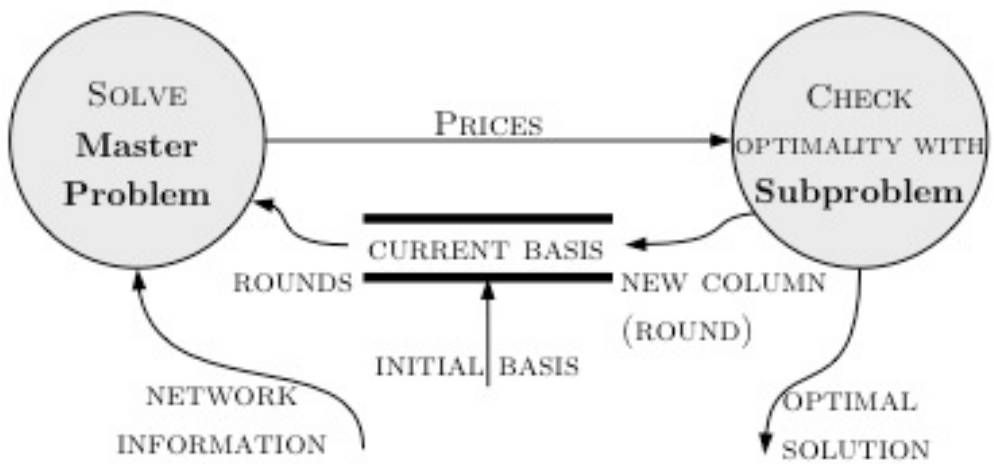
A quoi ça sert ?

- Le dual donne une borne supérieure
  - Algorithmes primal/dual : garantie sur la solution
- Pb de décision : primal plein = dual vide
  - Solution sous-optimale = contrainte duale violée
- Génération de colonnes...

Génération de colonne

$$Min\ c.x$$

$$Ax \geq b$$



CG Algorithm:
repeat
GET CURRENT BASIS
SOLVE MASTER PROBLEM
CHECK OPTIMALITY
if (not OPTIMAL)
SAVE NEW COLUMN
until (OPTIMAL SOLUTION)

$$Max\ b^t.y$$

$$A^t y \leq c^t$$

Retour sur le flot

$$Max\ d$$

$$s.t.$$

$$\sum_v f(v,s) + d - \sum_v f(s,v) = 0$$

$$\sum_v f(v,t) - d - \sum_v f(t,v) = 0$$

$$\sum_v f(v,u) - \sum_v f(u,v) = 0,\ \forall u$$

$$f(u,v) \leq c(u,v),\ \forall (u,v)$$

Retour sur le flot

$$Max\ d$$

$$s.t.$$

$$\sum_v f(v,s) + d - \sum_v f(s,v) = 0$$

$$\sum_v f(v,t) - d - \sum_v f(t,v) = 0$$

$$\sum_v f(v,u) - \sum_v f(u,v) = 0,\ \forall u$$

$$f(u,v) \leq c(u,v),\ \forall (u,v)$$

$$Max \sum_{p \in \mathcal{P}} f(p)$$

$$s.t.$$

$$\sum_{p \ni e} f(p) \leq c(e),\ \forall e$$

Retour sur le flot

$$Max\ d$$

$$s.t.$$

$$\sum_v f(v,s) + d - \sum_v f(s,v) = 0$$

$$\sum_v f(v,t) - d - \sum_v f(t,v) = 0$$

$$\sum_v f(v,u) - \sum_v f(u,v) = 0,\ \forall u$$

$$f(u,v) \leq c(u,v),\ \forall (u,v)$$

$$Max \sum_{p \in \mathcal{P}} f(p)$$

$$s.t.$$

$$\sum_{p \ni e} f(p) \leq c(e),\ \forall e$$

NOMBRE EXPONENTIEL DE VARIABLES

Des colonnes de flot

$$Max \sum_{p \in \mathcal{P}} f(p)$$

$$s.t.$$

$$\sum_{p \ni e} f(p) \leq c(e), \ \forall e$$

Des colonnes de flot

$$Max \sum_{p \in \mathcal{P}} f(p)$$

$$s.t.$$

$$\sum_{p \ni e} f(p) \leq c(e), \ \forall e$$

$$Min \sum_{e} \lambda(e) c(e)$$

$$s.t.$$

$$\sum_{e \in p} \lambda(e) \geq 1, \ \forall p$$

Des colonnes de flot

$$Max \sum_{p \in \mathcal{P}} f(p)$$

$$s.t. \sum_{p \ni e} f(p) \leq c(e), \ \forall e$$

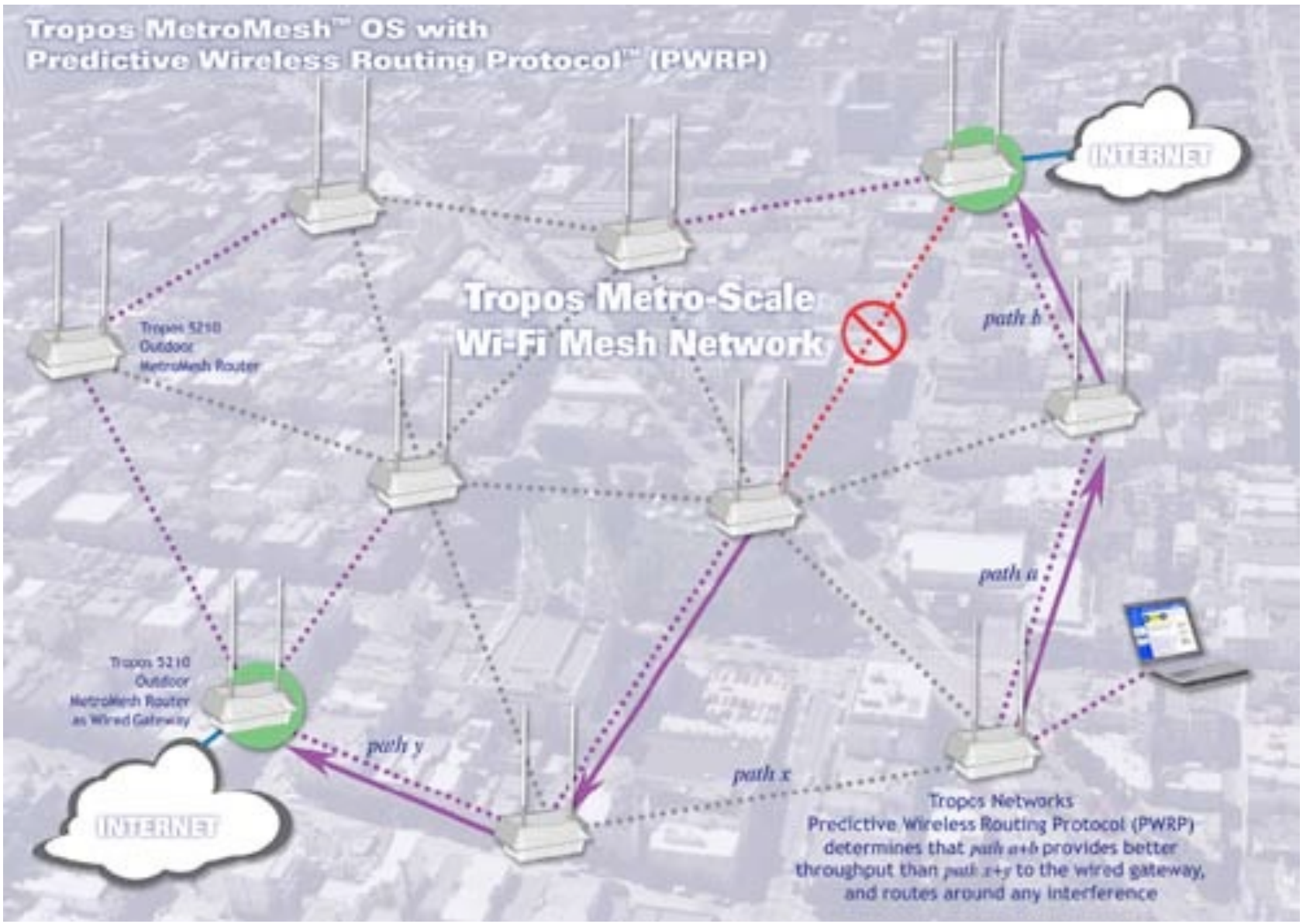
$$Min \sum_{e} \lambda(e) c(e)$$

$$s.t. \sum_{e \in p} \lambda(e) \geq 1, \ \forall p$$

- Génération de colonnes :
  - Démarrer avec un ensemble de chemins triviaux
  - Calculer le flot sous optimal, récupérer les lambda
- Trouver un plus court chemin
- S'il est trop court, on l'insère et on boucle
- Sinon, l'optimalité est prouvée

Application aux Mesh

Radio Mesh Networks



Model and objectives

- Synchronous network, periodic steady state
  - Slotted network
  - Throughput : #packets sent during each period/T

- Transport capacity
  - Known demand, maximum concurrent “flow”
  - Optim Throughput ==minimizing the period

Model and objectives

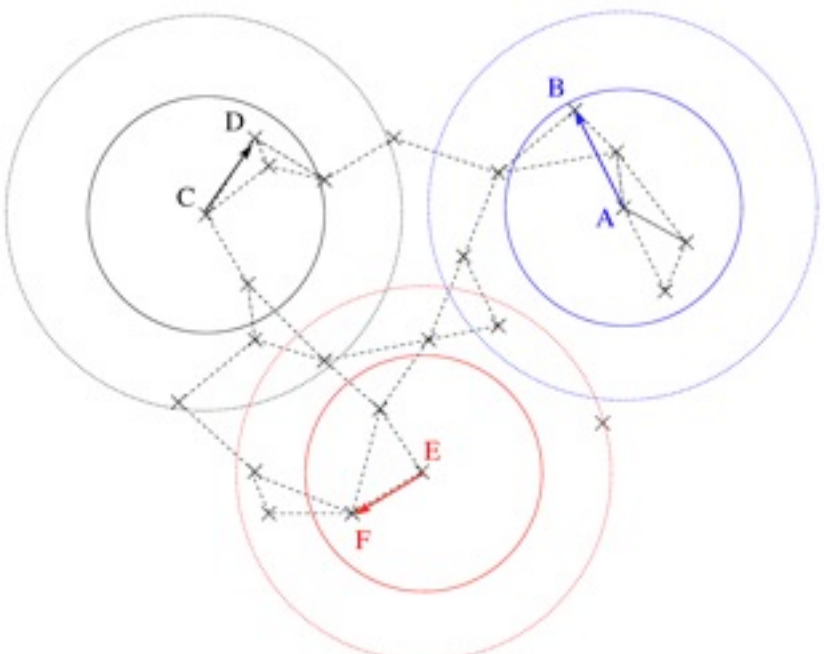
- Synchronous network, periodic steady state
  - Slotted network
  - Throughput : #packets sent during each period/T

$$\forall u \in V,\ D(u) \equiv \frac{\sum_{P \in \mathcal{P}_u} \Phi(P)}{T}$$

- Transport capacity
  - Known demand, maximum concurrent “flow”
  - Optim Throughput ==minimizing the period

The Rounds

- Round : set of simultaneouly activable links
  - Binary, distance-wise model
    - Sending at distance r, jamming at distance r(1+Δ)
  - Signal to noise model
    - Sending with power Pu, reception if Puα(u,v)>S∑(N°+Pu'α(u',v))



Minimization MILP

- Objective : minimize nb of APs

$$\min \sum_{i \in V_c} s_i, \tag{1}$$

$$\sum_{e \in E_v \cup E_u} a_e^t \leqslant 1, \forall v \in V, u \in N_v, t \in T \tag{2}$$

$$\sum_{v \in V} x_e^v \leqslant c_e \cdot \sum_{t \leqslant T} a_e^t / T, \forall e \in E \tag{3}$$

$$\sum_{e=(u,v) \in E} x_e^r + \mathbb{1}_{\{v=r\}} d_v = \sum_{e=(v,u) \in E} x_e^r + y_{r,v}, \forall r, v \in V \tag{4}$$

$$\sum_{i \in V_c} y_{r,i} = d_r, \forall r \in V \tag{5}$$

$$y_{r,i} \leqslant s_i \cdot C, \forall i \in V_c, r \in V \tag{6}$$

Minimization MILP

- Objective : minimize nb of APs

$$\min \sum_{i \in V_c} s_i, \tag{1}$$

Admissibles calls

$$\sum_{e \in E_v \cup E_u} a_e^t \leqslant 1, \forall v \in V, u \in N_v, t \in T \tag{2}$$

Link capacities

$$\sum_{v \in V} x_e^v \leqslant c_e \cdot \sum_{t \leqslant T} a_e^t / T, \forall e \in E \tag{3}$$

Routing

$$\sum_{e=(u,v) \in E} x_e^r + \mathbb{1}_{\{v=r\}} d_v = \sum_{e=(v,u) \in E} x_e^r + y_{r,v}, \forall r, v \in V \tag{4}$$

Demand transport

$$\sum_{i \in V_c} y_{r,i} = d_r, \forall r \in V \tag{5}$$

$$y_{r,i} \leqslant s_i \cdot C, \forall i \in V_c, r \in V \tag{6}$$

Minimization MILP

- Objective : minimize nb of APs

$$\min \sum_{i \in V_c} s_i, \tag{1}$$

Admissibles calls

$$\sum_{e \in E_v \cup E_u} a_e^t \leqslant 1, \forall v \in V, u \in N_v, t \in T \tag{2}$$

Link capacities

$$\sum_{v \in V} x_e^v \leqslant c_e \cdot \sum_{t \leqslant T} a_e^t / T, \forall e \in E \tag{3}$$

Routing

$$\sum_{e=(u,v) \in E} x_e^r + \mathbb{1}_{\{v=r\}} d_v = \sum_{e=(v,u) \in E} x_e^r + y_{r,v}, \forall r, v \in V \tag{4}$$

Demand transport

$$\sum_{i \in V_c} y_{r,i} = d_r, \forall r \in V \tag{5}$$

$$y_{r,i} \leqslant s_i \cdot C, \forall i \in V_c, r \in V \tag{6}$$

Very hard to solve !

Routing and scheduling

- Obj : minimize round activation = max Tput

$$Min \sum_{R \in \mathcal{R}} w(R)$$

$$C_w(e) = \sum_{R \in \mathcal{R}, e \in R} w(R)$$

$$C_w(e) \geq \sum_{P \in \mathcal{P}, e \in P} \Phi(P) \qquad \forall e \in E$$

$$\sum_{P \in \mathcal{P}_v} \Phi(P) = t(v) \qquad \forall v \in V_r$$

Routing and scheduling

- Obj : minimize round activation = max Tput

$$Min \sum_{R \in \mathcal{R}} w(R)$$

$$C_w(e) = \sum_{R \in \mathcal{R}, e \in R} w(R)$$

Activating rounds gives capacity

$$C_w(e) \geq \sum_{P \in \mathcal{P}, e \in P} \Phi(P) \qquad \forall e \in E$$

In order to transport the flow

$$\sum_{P \in \mathcal{P}_v} \Phi(P) = t(v) \qquad \forall v \in V_r$$

And route the demand

Routing and scheduling

- Obj : minimize round activation = max Tput

$$Min \sum_{R \in \mathcal{R}} w(R)$$

$$C_w(e) = \sum_{R \in \mathcal{R}, e \in R} w(R)$$

Activating rounds gives capacity

$$C_w(e) \geq \sum_{P \in \mathcal{P}, e \in P} \Phi(P) \qquad \forall e \in E$$

In order to transport the flow

$$\sum_{P \in \mathcal{P}_v} \Phi(P) = t(v) \qquad \forall v \in V_r$$

And route the demand

Sets of rounds and paths are exponential
Column generation is necessary

Dualité, encore

$$Min \sum_{R \in \mathcal{R}} w(R)$$

$$C_w(e) = \sum_{R \in \mathcal{R}, e \in R} w(R)$$

$$C_w(e) \geq \sum_{P \in \mathcal{P}, e \in P} \Phi(P) \qquad \forall e \in E$$

$$\sum_{P \in \mathcal{P}_v} \Phi(P) = t(v) \qquad \forall v \in V_r$$

$$Min \sum_{e} \lambda(e) c(e)$$

$$s.t.$$

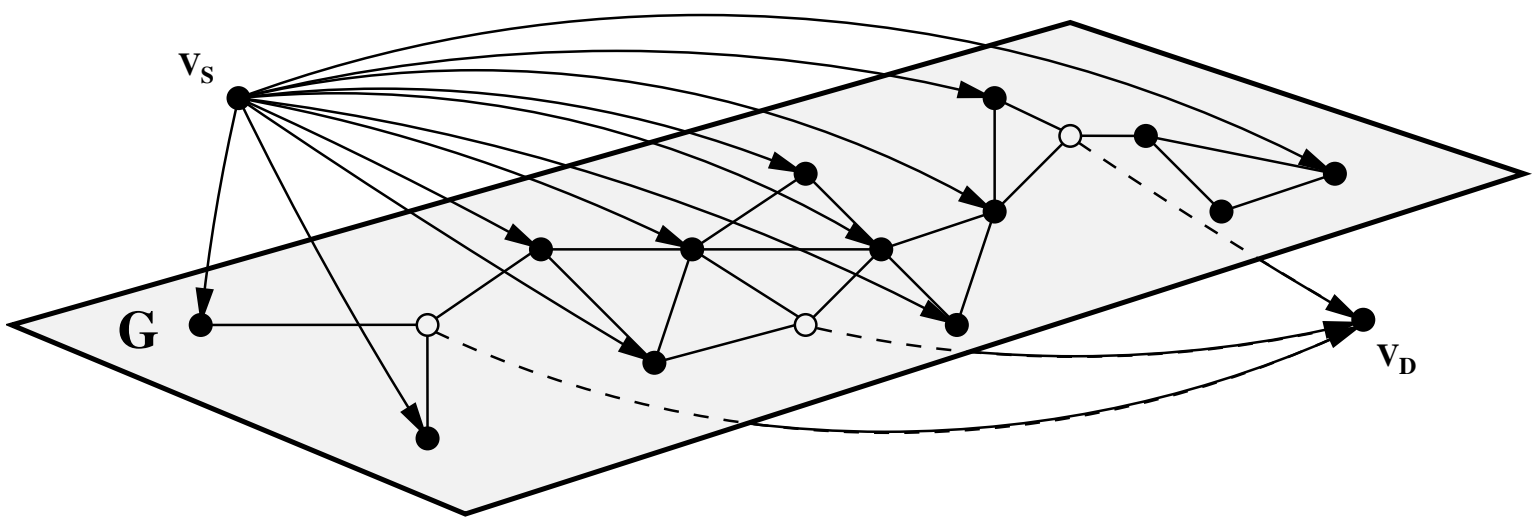
$$\sum_{e \in R} \lambda(e) \leq 1, \ \forall R$$

$$\sum_{e \in p} \lambda(e) \geq 1, \ \forall p$$

- Génération de colonnes :
- Démarrer avec un ensemble de chemins et de rounds triviaux
- Calculer le RWP sous optimal, récupérer les lambda

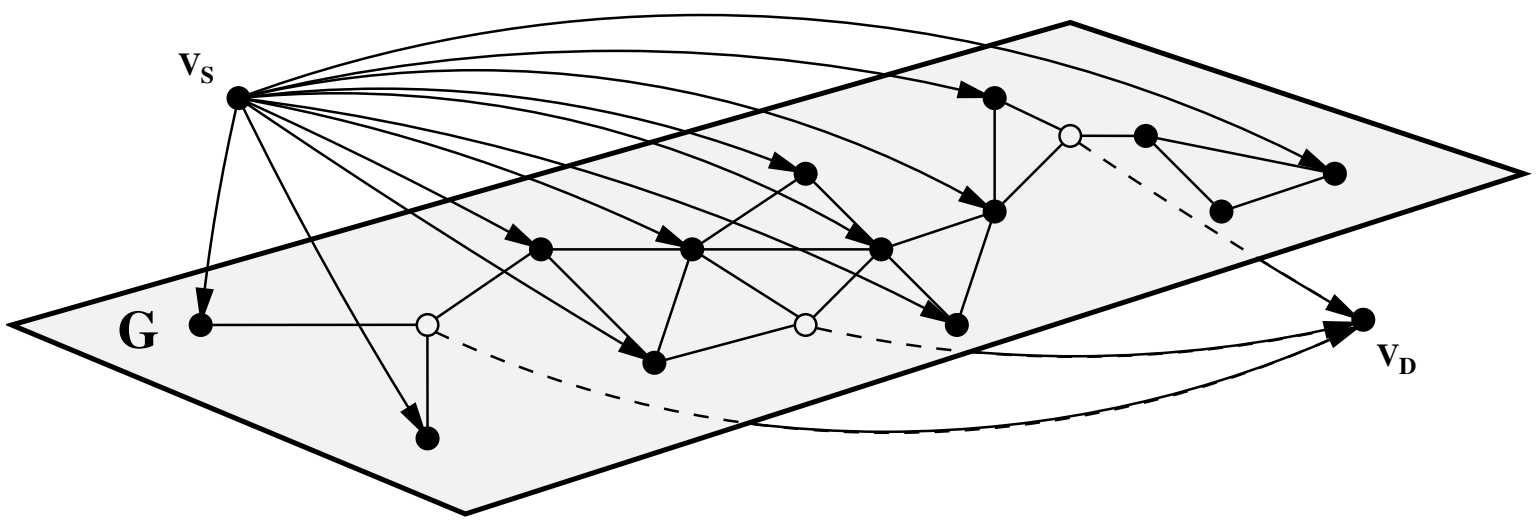
- Trouver un plus court chemin ou un round le plus lourd
- **Toute la difficulté est réduite au calcul du round**

Max-flow & Min-cut



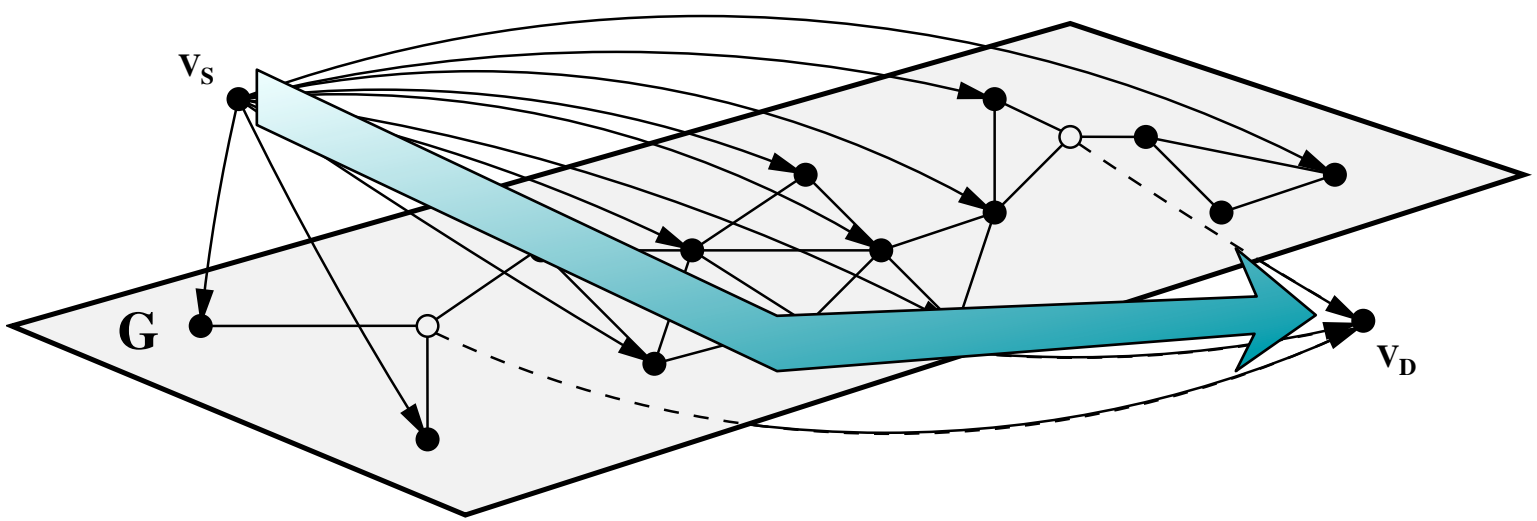
Max-flow & Min-cut

- If it exists a flow Φ,



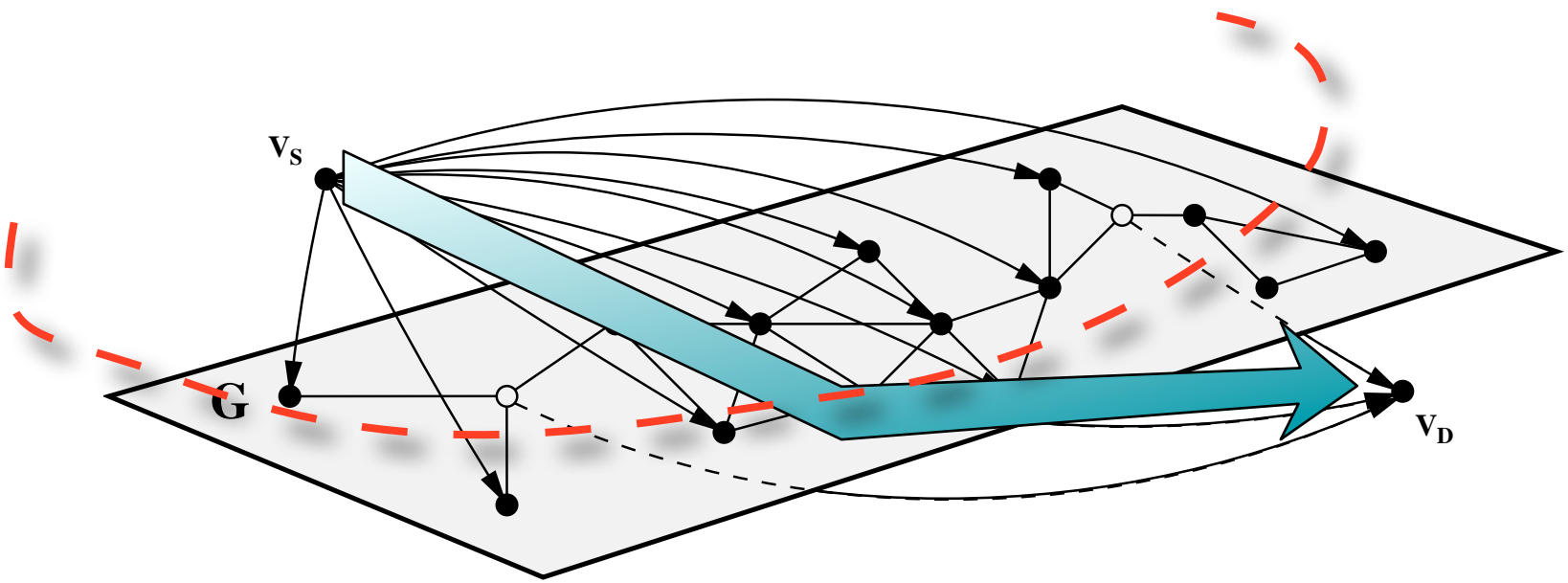
Max-flow & Min-cut

- If it exists a flow Φ,
  - Φ sends ∑t(v) from Vs to Vd in G



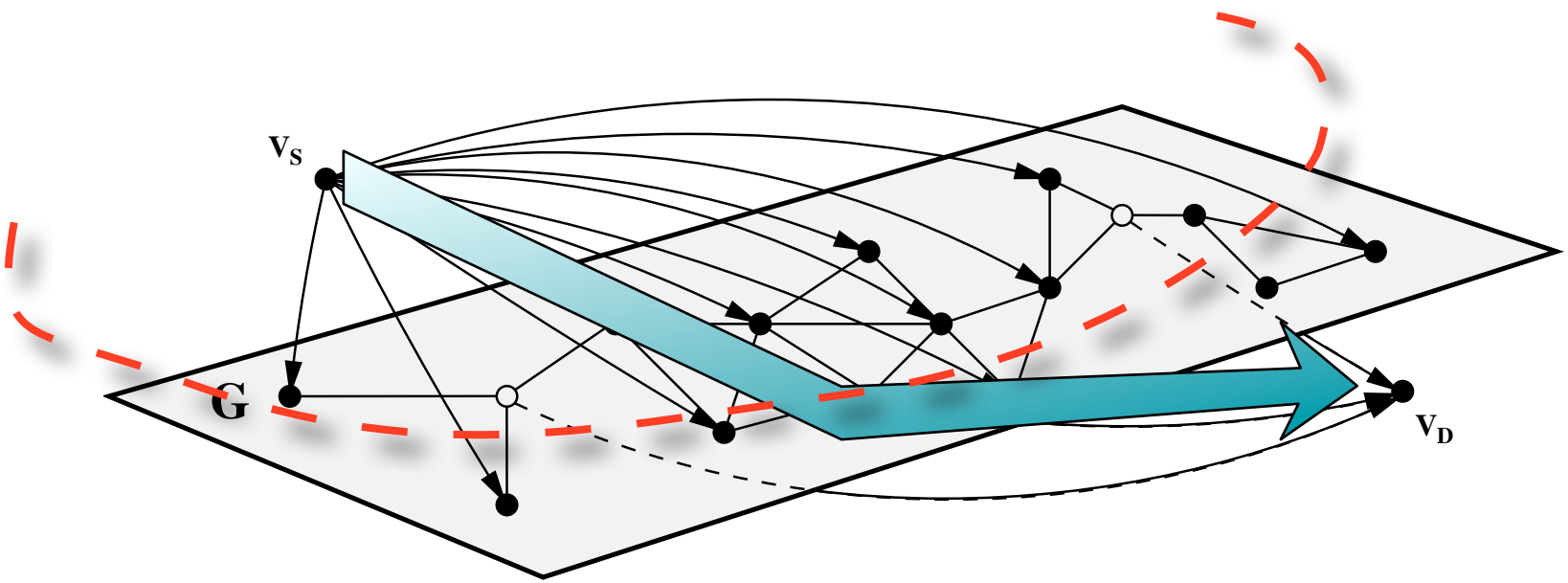
Max-flow & Min-cut

- If it exists a flow Φ,
  - Φ sends ∑t(v) from Vs to Vd in G
  - S is a cut separating Vs and Vd : Cw(S) ≥ ∑t(v)



Max-flow & Min-cut

- If it exists a flow Φ,
  - Φ sends ∑t(v) from Vs to Vd in G
  - S is a cut separating Vs and Vd : Cw(S) ≥ ∑t(v)
  - Cw(S) = ∑δ(R,S)w(R)



Round/Cut Formulation

- Obj : min round activation

$$Min \sum_{R \in \mathcal{R}} w(R)$$

$$\sum_{R \in \mathcal{R}} \delta(R, S) w(R) \geq t(S) \qquad \forall S \in \mathcal{S}$$

Round/Cut Formulation

- Obj : min round activation

$$Min \sum_{R \in \mathcal{R}} w(R)$$

$$\sum_{R \in \mathcal{R}} \delta(R, S) w(R) \geq t(S) \qquad \forall S \in \mathcal{S}$$

providing enough capacity for the traffic

Round/Cut Formulation

- Obj : min round activation

$$Min \sum_{R \in \mathcal{R}} w(R)$$

$$\sum_{R \in \mathcal{R}} \delta(R, S) w(R) \geq t(S) \qquad \forall S \in \mathcal{S}$$

providing enough capacity for the traffic

Covering of the cuts by the rounds

Set of the rounds : as before
Set of the cuts : line generation

Line and column gen.

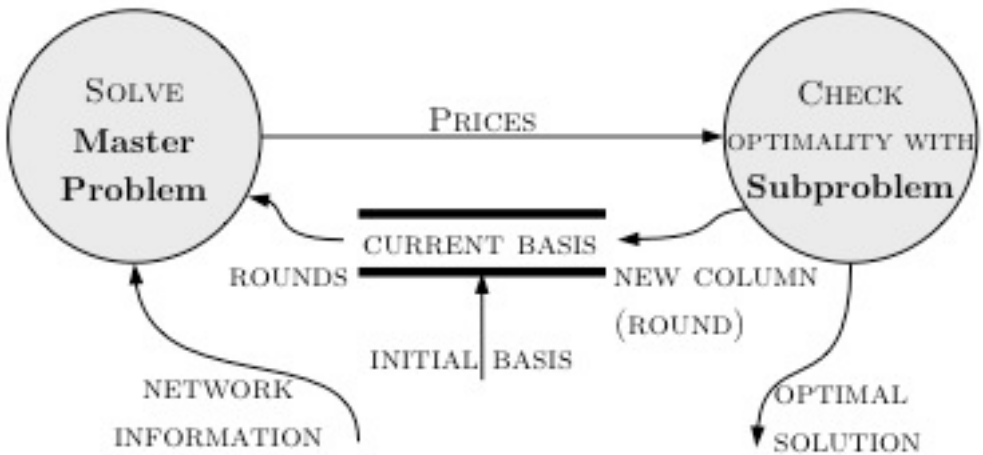
Compute an initial set S

Solve the primal with S

Line and column gen.

Compute an initial set S

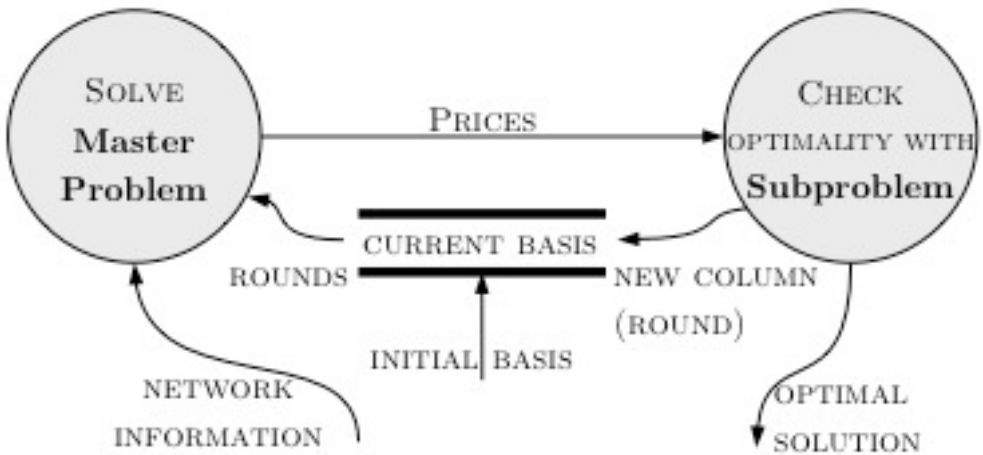
Solve the primal with S



Line and column gen.

Compute an initial set S

Solve the primal with S



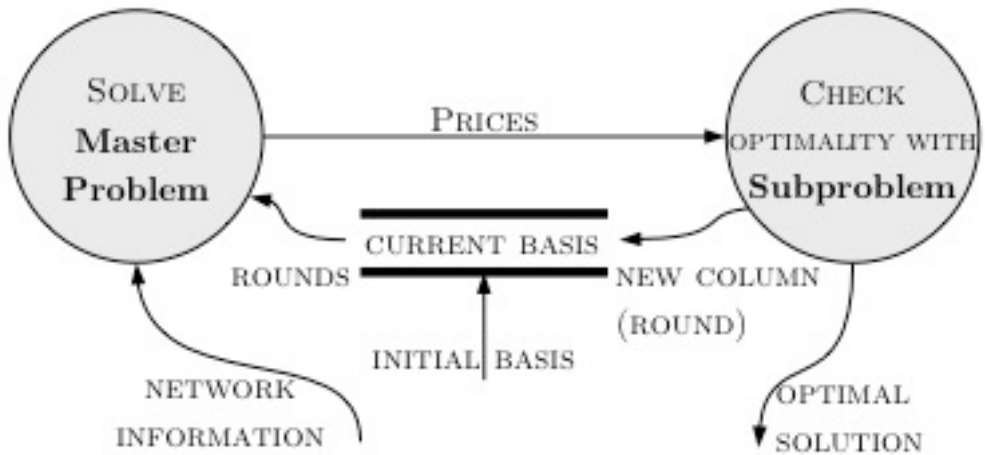
Look for a violated constraint in the primal

Compute a minimum cut, update S

Line and column gen.

Compute an initial set S

Solve the primal with S



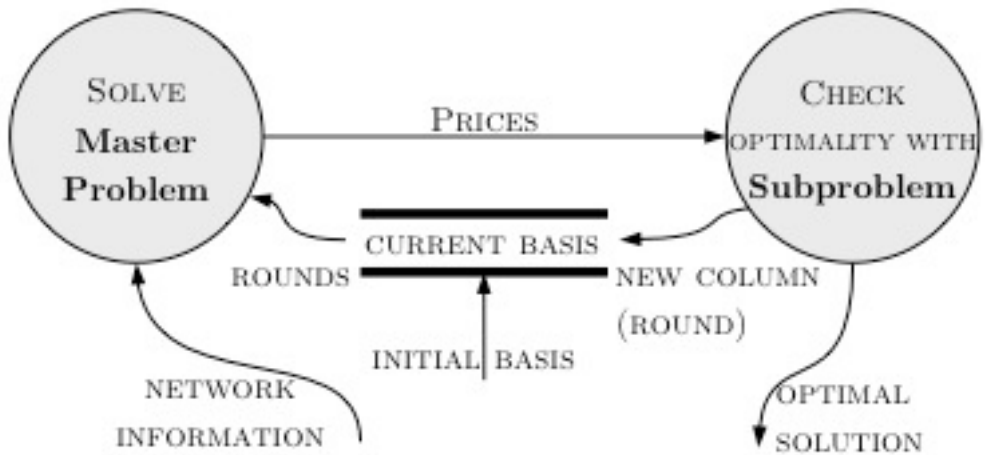
Look for a violated constraint in the primal

Compute a minimum cut, update S

Line and column gen.

Compute an initial set S

Solve the primal with S

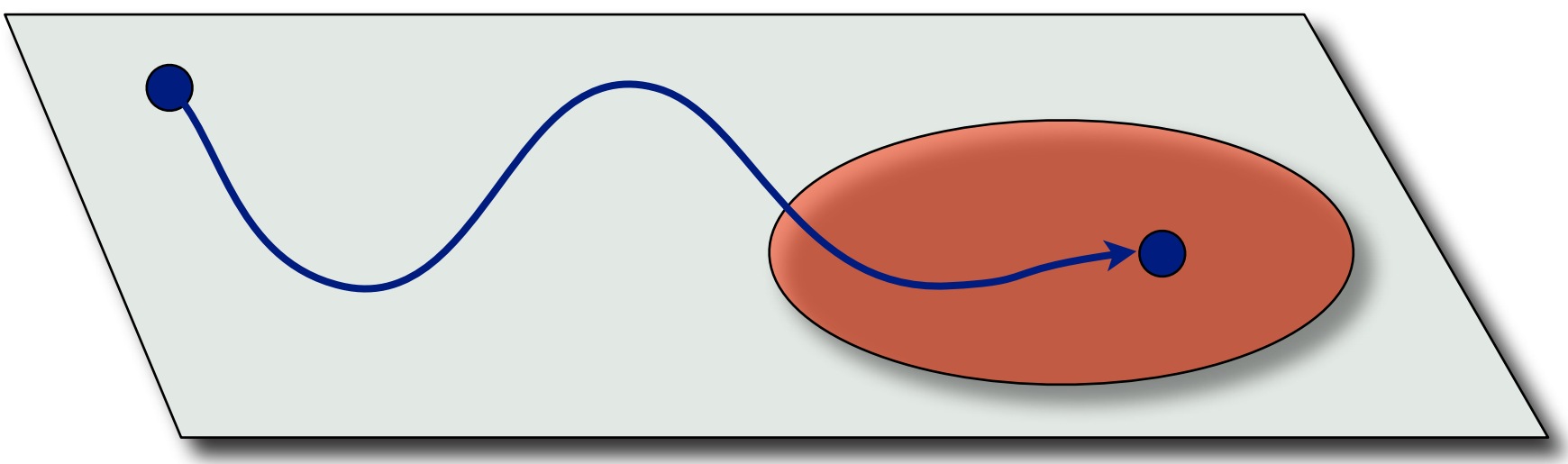


Look for a violated constraint in the primal

Compute a minimum cut, update S

Optimal solution

Interest of two models

- Flow : control on the kind of paths
- Cut : zone constraints related to the problem

- Similar complexity
  - for the whole network
  - local optimization might be enough