

COMPOSITIONS  
HYBRIDATIONS  
pour  
l'OPTIMISATION  
COMBINATOIRE

**Sophie Demasse**  
CMA, Mines ParisTech

1. contexte

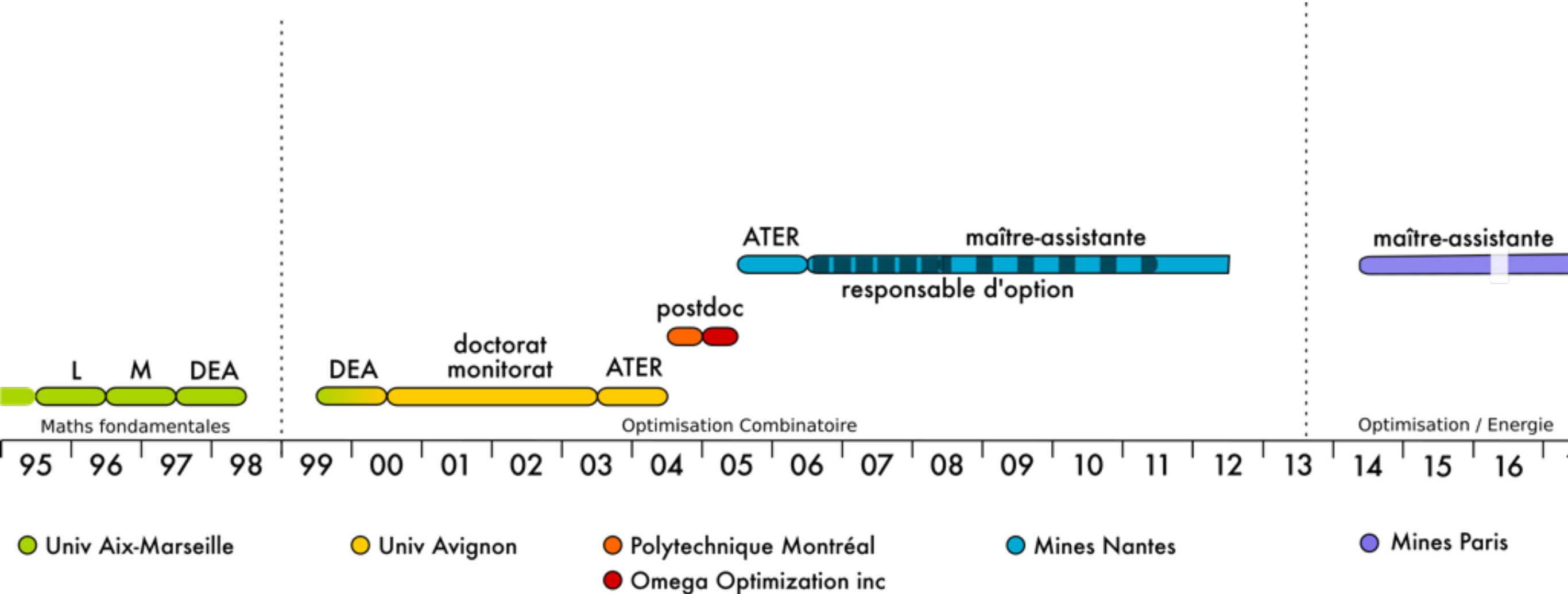
2. optimisation combinatoire flexible

3. BtrPlace et les contraintes globales

4. contraintes, automates et coûts

5. perspectives énergétiques

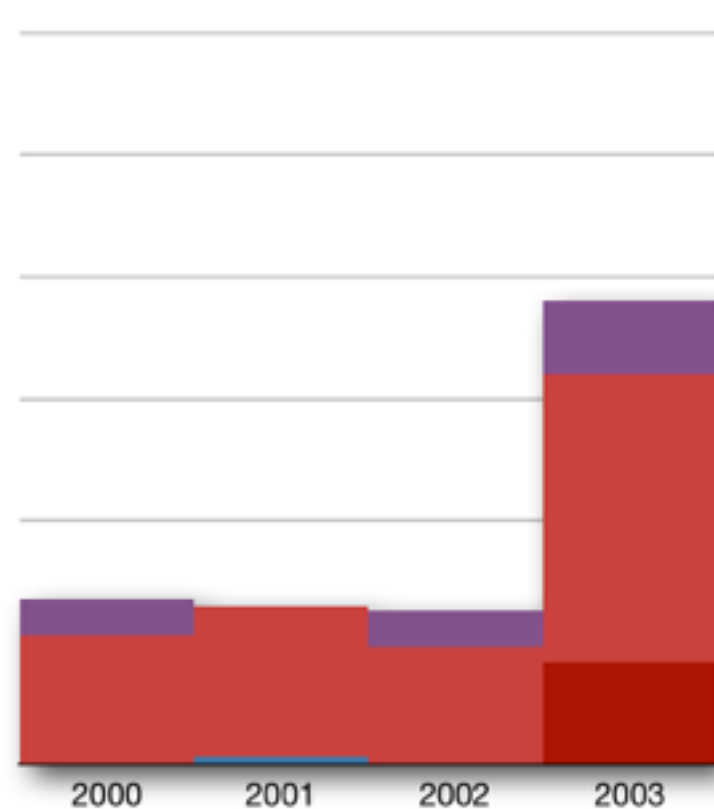
# PARCOURS



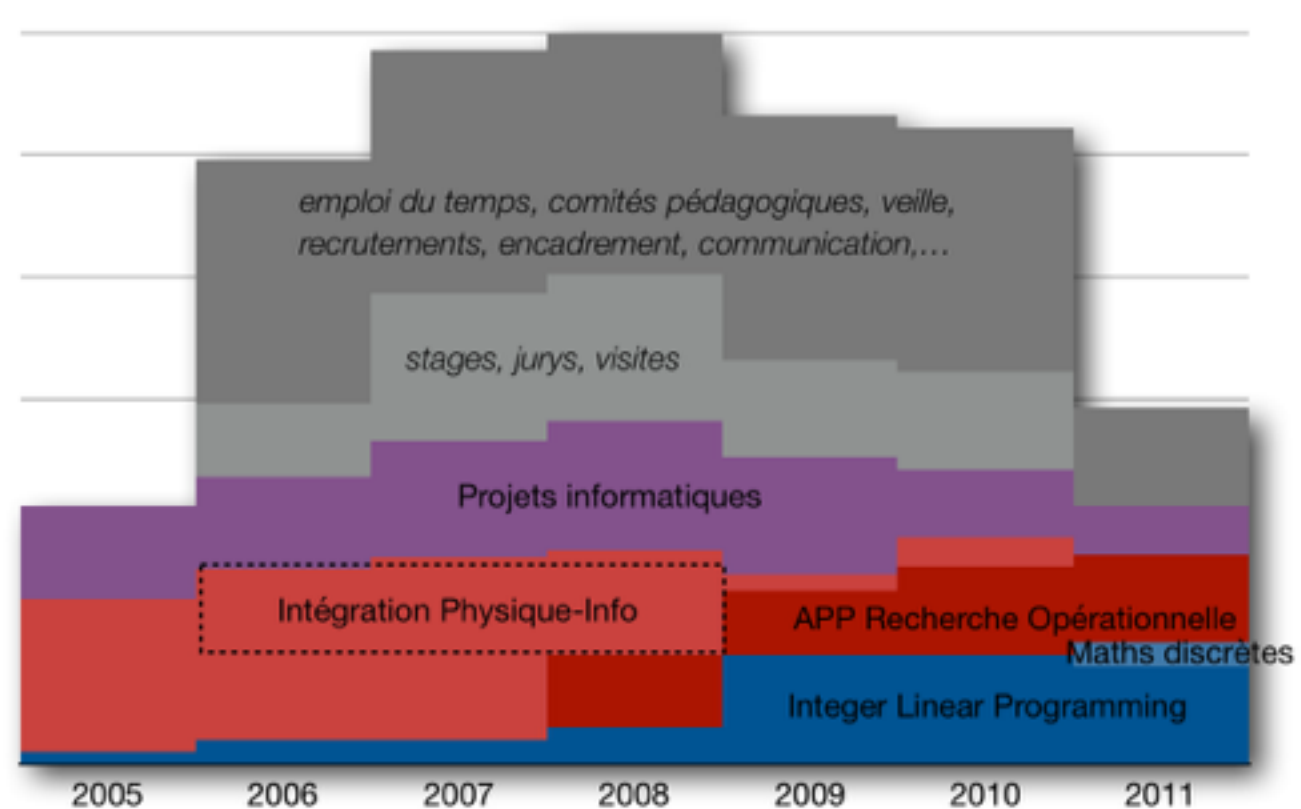
# ACTIVITÉS PÉDAGOGIQUES

heures

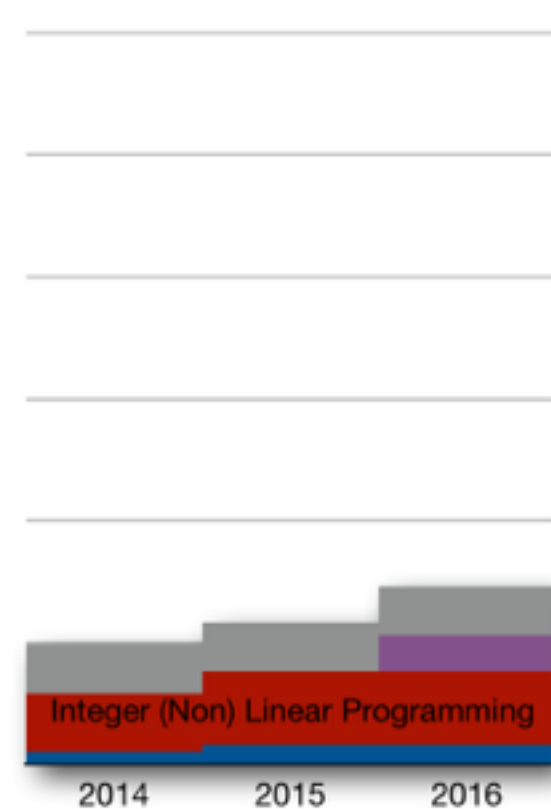
**Avignon: IUP (300h) /  
Fac Sciences (90h)**



**Nantes: Mines (760h hors resp.) /  
Fac Sciences (90h)**



**Mines Paris: MS OSE (150h)**



■ cours M ■ cours L ■ TD M ■ TD L ■ projets ■ entreprise ■ resp. GIPAD

## EXERCICE

12 années chercheuse + enseignante + responsable pédagogique

LIA, CIRRELT, LINA, CMA

## PRODUCTION

1 à 2 articles originaux par an (revues RO & conf PPC; appliqués)

contributions aux logiciels libres Choco, BtrPlace/Entropy, NRP

co-encadrement de 6 thèses et de 7 masters

## ANIMATION

expertise pour 2/3 articles de revues par an, 13 PC de conférences

coordination de Contraintes & RO et LigéRO; bureau Roadef

webmaster Global Constraint Catalog

FLEXIBILITÉ  
en  
OPTIMISATION  
COMBINATOIRE

ordonnancement d'instructions sur architectures parallèles

gestion de projets à moyens limités

planification de personnel

saturation de la capacité d'un noeud ferroviaire

placement de blocs 3D

gestion des ressources dans un centre de données

localisation de cibles par des radars

routage dans les réseaux logistiques

relocalisation de conteneurs à quai dans les ports maritimes

allumage des pompes dans un réseau de distribution d'eau

dimensionnement d'un microgrid

gestion de la charge en réponse à un signal dynamique

modélisation prospective d'un système énergétique

# PROBLÈMES

- ▶ difficiles
- ▶ protéiformes
- **composites**

# MÉTHODES

- ▶ performantes
- ▶ polyvalentes
- **modulaires**
- **hybrides**
- **déclaratives**

# (DÉ)COMPOSITION

résolution modulaire et intégrée des sous-problèmes

## Performance

- passage à l'échelle
- hybridation

## Flexibilité

- robustesse
- réutilisabilité

dans des paradigmes déclaratifs



# DÉCLARATIVE

# IMPÉRATIVE

$$\begin{aligned} \min \quad & \sum_{j=1}^m s_j^+ + s_j^- \\ \text{s.t.} \quad & \sum_{i=1}^n a_{ij} x_i + s_j^+ - s_j^- = \frac{d_j}{2} \quad j = 1..m \\ & x_i \in \{0, 1\} \quad i = 1..n \\ & s_j^+ \geq 0, s_j^- \geq 0 \quad j = 1..m \end{aligned}$$

```
[sofdem:~/Documents/Code/gurobi]$ gurobi.sh mymip.py markshare_5_0.mps.gz
Optimize a model with 5 rows, 45 columns and 203 nonzeros
Found heuristic solution: objective 5335
Presolve time: 0.00s
Presolved: 5 rows, 45 columns, 203 nonzeros
Variable types: 0 continuous, 45 integer (40 binary)

Root relaxation: objective 0.000000e+00, 15 iterations, 0.00 seconds
```

Nodes		Current Node			Objective Bounds			Work	
Expl	Unexpl	Obj	Depth	IntInf	Incumbent	BestBd	Gap	It/Node	Time
	0	0	0.00000	0	5 5335.00000	0.00000	100%	-	0s
H	0	0			320.0000000	0.00000	100%	-	0s
	0	0	0.00000	0	6 320.00000	0.00000	100%	-	0s
	0	0	0.00000	0	5 320.00000	0.00000	100%	-	0s
	0	0	0.00000	0	6 320.00000	0.00000	100%	-	0s
	0	0	0.00000	0	5 320.00000	0.00000	100%	-	0s
H	0	0			239.0000000	0.00000	100%	-	0s
	0	0	0.00000	0	5 239.00000	0.00000	100%	-	0s
*	36	0		29	96.0000000	0.00000	100%	2.7	0s
*	99	32		34	58.0000000	0.00000	100%	2.1	0s
H	506	214			53.0000000	0.00000	100%	1.9	0s
H30682	442				1.0000000	1.00000	0.00%	2.1	0s

```
Cutting planes:
Cover: 26
```

```
Explored 30682 nodes (65348 simplex iterations) in 0.70 seconds
Thread count was 4 (of 4 available processors)
```

```
Optimal solution found (tolerance 1.00e-04)
Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0%
Optimal objective: 1
```

```
public int greedy(int first, int[] whichStep, int[] whichCont, int[] whichStack) {

    int step = first;
    assert whichStep[0] == step;
    int greedy = 0;
    int n = 0;
    int nextStep = 0;

    initialize(first);

    while (step < nContainers) {
        boolean hasNotMoved = true;
        int pos = firstPred - 1;
        int firstMove = lastPred+1;

        // find the position of step in pred if it has already moved at least once
        if (step == pred[nextStep]) {
            hasNotMoved = false;
            nextStep = nContainers + 1;
            for (int p = firstPred; pos < firstPred && p <= lastPred; p++) {
                if (pred[p] == step) {
                    pos = p;
                    pred[pos] = 0;
                } else if (pred[p] > 0 && pred[p] < nextStep) {
                    nextStep = pred[p];
                }
            }
        }

        // repush in pred all the containers currently above step after being moved at least once
        assert pos >= firstPred - 1;
        sz[stk[step]]--;
        for (int p = lastPred; p > pos; p--) {
            if (pred[p] > 0) {
                int i = pred[p];
                if (stk[i] == stk[step]) {
                    greedy++;
                    sz[stk[step]]--;
                    stk[i] = -1;
                    pred[++lastPred] = i;
                    pred[p] = 0;
                }
            }
            if (i < nextStep) {
                nextStep = i;
            }
        }
        while (pred[firstPred] == 0) {
            firstPred++;
        }

        // if step has never moved then push in pred all the containers above step that never
        // when possible assign these containers according to the solution of the MIP
        if (hasNotMoved) {
            stepMove[step] = step;
        }
    }
}
```

# DÉCLARATIVE VS. IMPÉRATIVE

séparation modèle / algorithme



- modèle flexible
- préoccupation haut niveau
- expertise réduite en algorithmique
- validité, vérification, réutilisabilité



- séparation partielle
- paramétrage ou réimplémentation
- expertise requise en modélisation
- solveurs coûteux, boîte noire

# DÉCLARATIVE HYBRIDE

programmation  
mathématique

$$\begin{aligned}
 & \min \sum_{t=ES_{n+1}, \dots, T} ty_{(n+1)t} \\
 & \text{subject to:} \\
 & \sum_{t=ES_j}^{LS_j} y_{jt} = 1 \quad \forall j \in V \\
 & \sum_{t=ES_j}^{LS_j} ty_{jt} - \sum_{t=ES_i}^{LS_i} ty_{it} \geq b_{ij} \quad \forall (i, j) \in V^2 \\
 & \sum_{j \in V} (r_{jk} \cdot \sum_{\tau=\max(ES_j, t-p_j+1)}^{\min(LS_j, t)} y_{j\tau}) \leq R_k \quad \forall k \in \mathcal{R}, \forall t \in \{0, \dots, T\} \\
 & y_{jt} \in \{0, 1\} \quad \forall j \in V, \forall t \in \{ES_j, \dots, LS_j\}
 \end{aligned}$$

global  
optimisation  
scalable  
décomposition externe

programmation  
par contraintes

$$\begin{aligned}
 & \text{gcc}(\langle \sigma_a | a \in A \rangle, \langle a \in A \rangle, \langle s_1, \dots, s_T \rangle) \\
 & \sigma == \sum_{a \in W} \sigma_a \\
 & \sigma < 24 \Rightarrow (\sigma_l == 0 \wedge \sigma_p == 1) \\
 & \sigma \geq 24 \Rightarrow (\sigma_l == 4 \wedge \sigma_p == 2) \\
 & \text{regular}(\langle s_1, \dots, s_T \rangle, \Pi) \\
 & s_t \in A \setminus F_t, \quad \forall t = 1, \dots, T \\
 & \sigma_a \in \{0, \dots, 32\}, \quad \forall a \in W, \quad \sigma \in \{12, 32\} \\
 & \sigma_l \in \{0, 4\}, \quad \sigma_p \in \{1, 2\}, \quad \sigma_o \in \{58, \dots, 83\}
 \end{aligned}$$

local  
satisfaction  
compact  
décomposition interne

# VERROUS

## MODULAIRE

- quel degré de décomposition ?

## HYBRIDE

- quelles techniques ? comment les adapter ? comment assurer la compatibilité ?

## DÉCLARATIVE

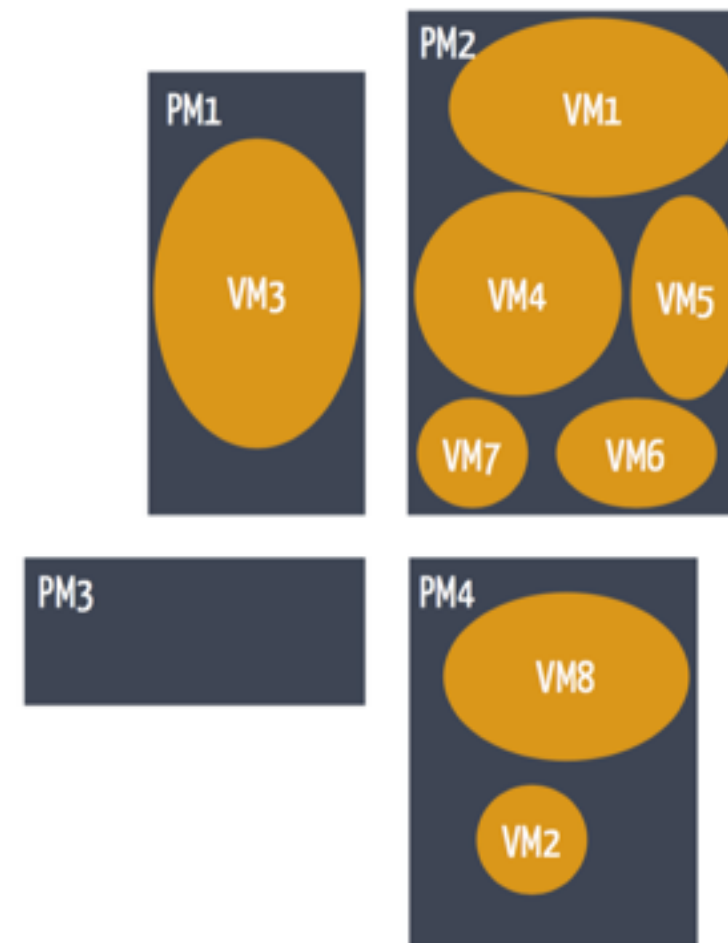
- comment modéliser de manière flexible et efficace ?

# BTRPLACE

## COMPOSABILITÉ ET ERGONOMIE DES CONTRAINTES GLOBALES

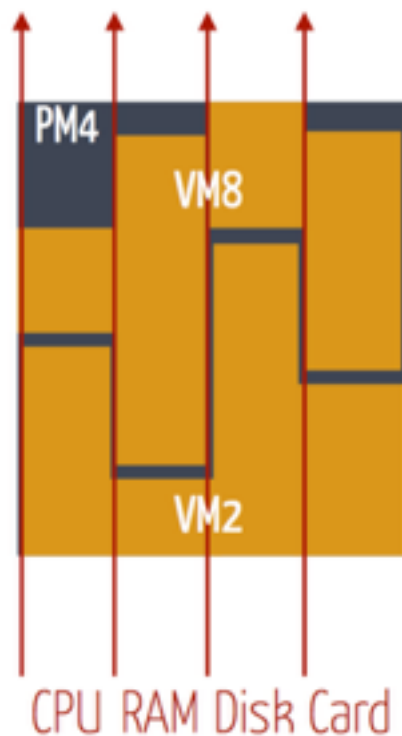
# GESTION DE RESSOURCES DANS LES CENTRE DE DONNÉES

- ré-affecter les tâches (VM) aux machines (PM) en minimisant l'impact sur la performance



# GESTION DE RESSOURCES DANS LES CENTRE DE DONNÉES

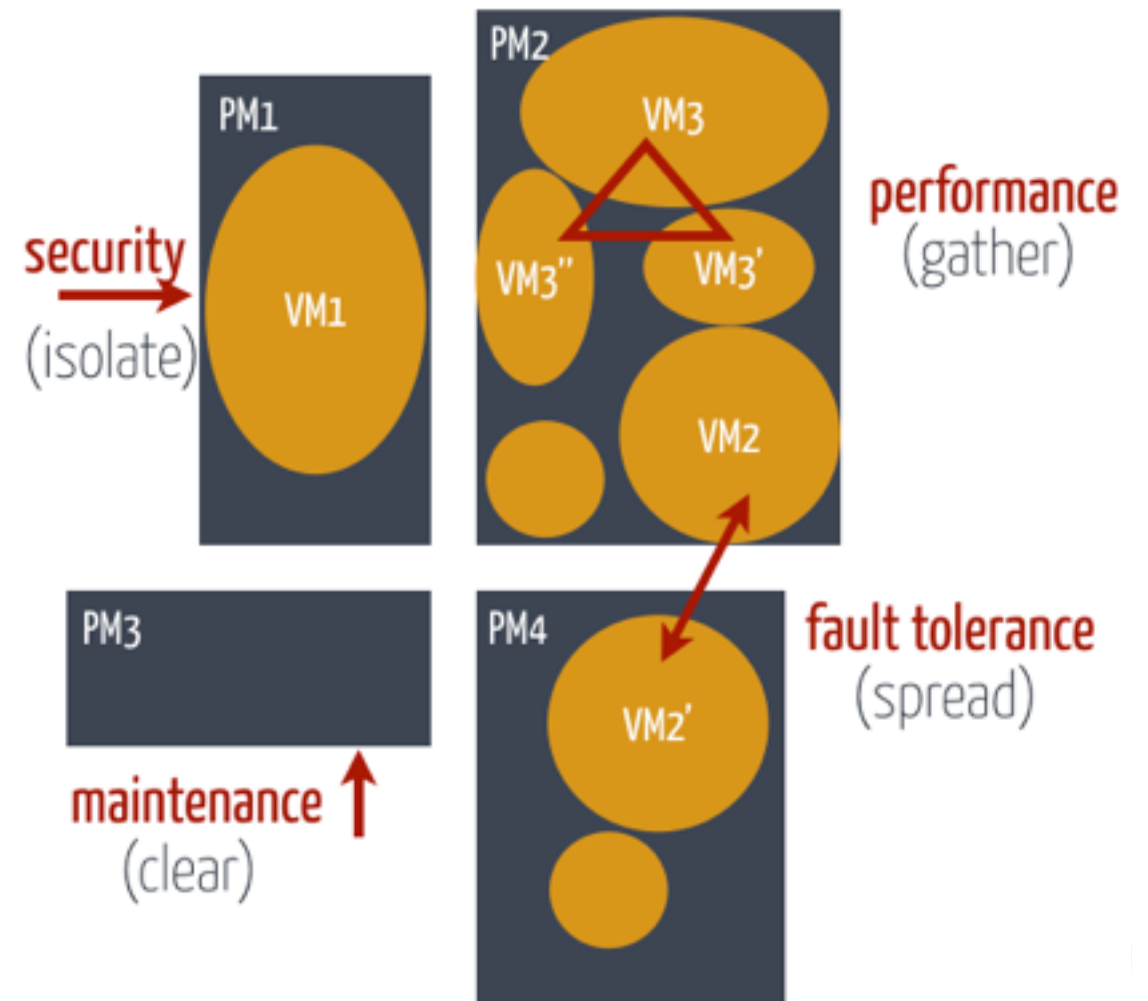
- ré-affecter les tâches (VM) aux machines (PM) en minimisant l'impact sur la performance
- vector packing
- ordonnancement producteur/consommateur



# GESTION DE RESSOURCES DANS LES CENTRE DE DONNÉES

- ré-affecter les tâches (VM) aux machines (PM) en minimisant l'impact sur la performance

- vector packing
- ordonnancement producteur/consommateur
- temps réel
- données hétérogènes et dynamiques
- contraintes utilisateur hétérogènes et dynamiques







# BtrPlace

An Open-Source flexible virtual machine scheduler

- modèle de contraintes sur le solveur Choco
- 2 contraintes globales spécifiques: vector packing & ordonnancement producteur/consommateur
- contraintes utilisateurs auto-composables via un DSL extensible
- mode heuristique de réparation des conflits
  - ➔ spécialisation aux instances petites et difficiles
  - ➔ décomposition packing / ordo, couplage packing + cardinalités, bornes, back-propagation
  - ➔ auto-paramétrisation



# CONTRAINTES GLOBALES



fonctionnel: déclaratif  
réutilisable



explosion du langage

flexible: composable  
extensible

adaptabilité limitée

implémentation complexe

efficace: spécialisable  
contexte d'hybridation

communication limitée

traitement limité du coût

MCR/NRP

METACONTRAINTES - AUTOMATES - COÛTS

# CONTRAINTE = LANGAGE

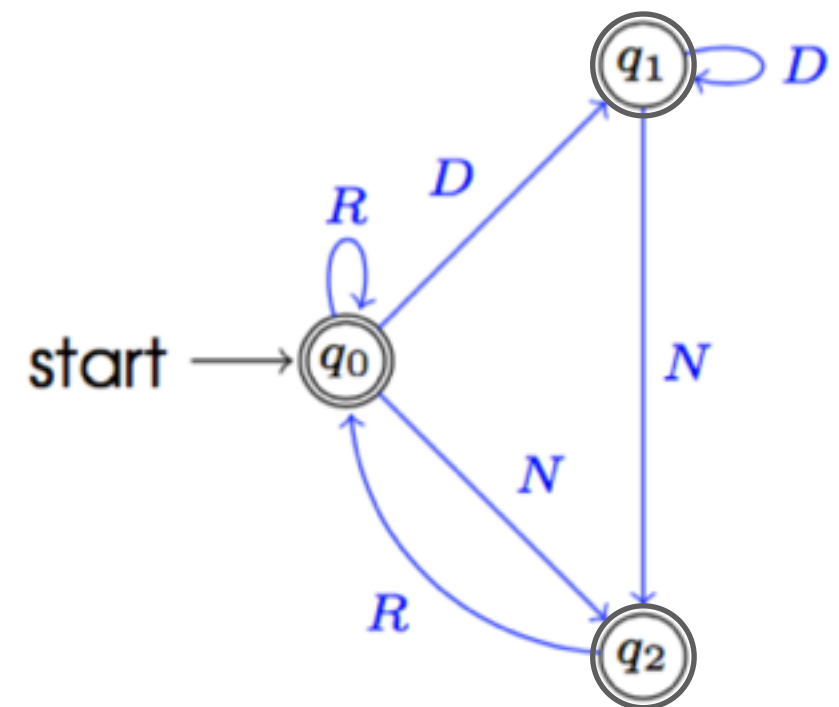
- solution (discrète) = mot (de taille fixe)
- contrainte = langage

exemple Nurse Scheduling: RRDDNRD

“repos obligatoire après un quart de nuit”

automate fini = modèle implicite compact

contraintes globales regular / automaton



# CONTRAINTE = LANGAGE

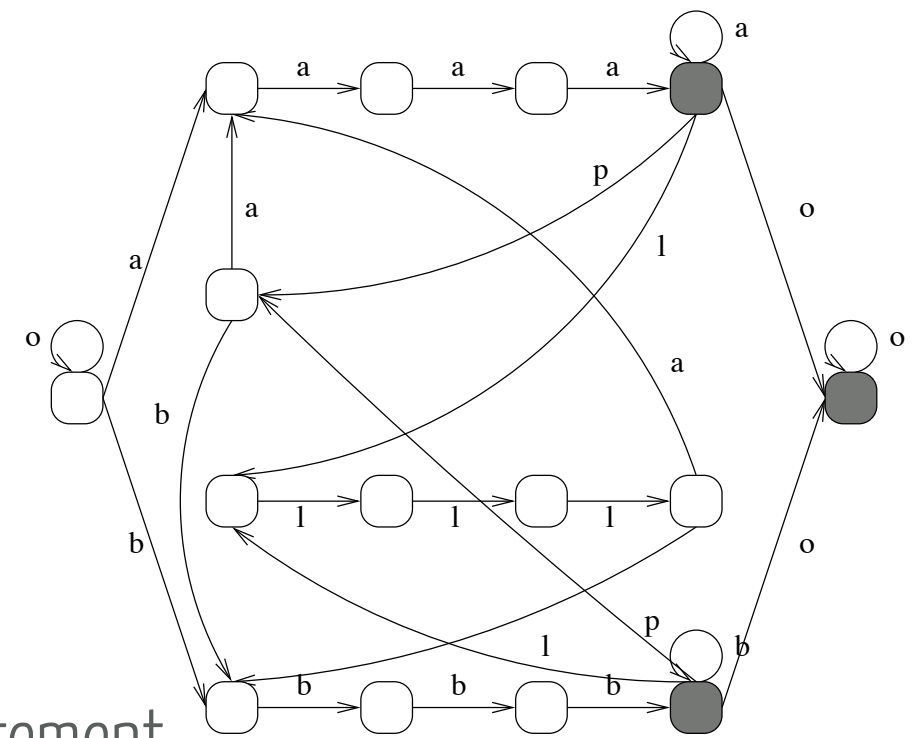
- solution (discrète) = mot (de taille fixe)
- contrainte = langage

exemple Nurse Scheduling: **RRDDNRD**

“repos obligatoire après un quart de nuit”

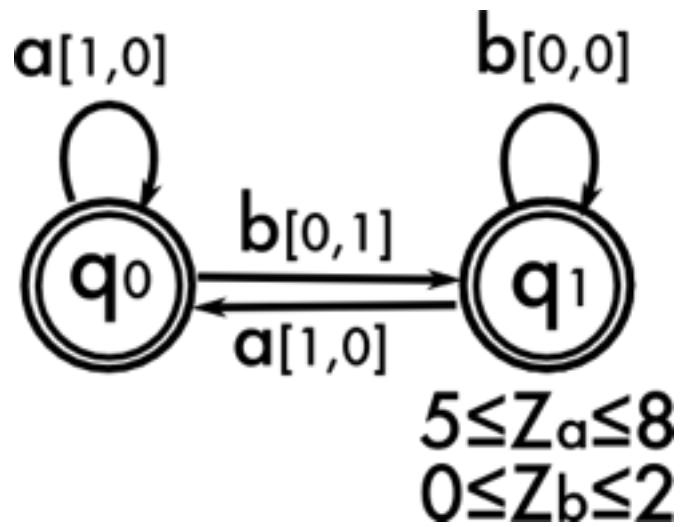
automate fini = modèle implicite compact **composable**

contraintes globales `regular/automaton`

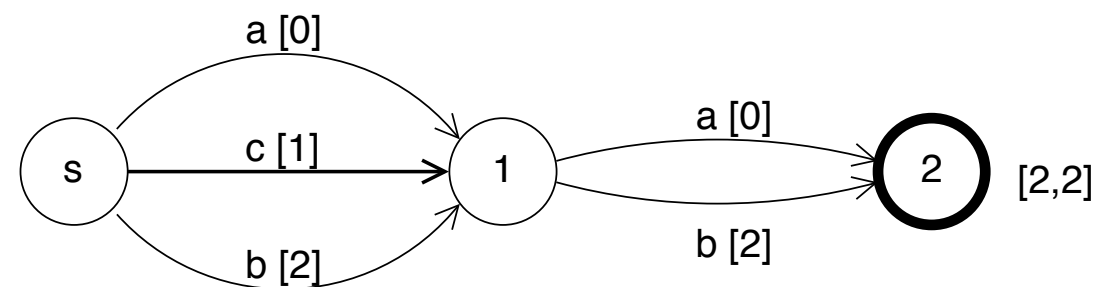


1. expressivité limitée —> ajout de pondérations+bornes
2. modélisation complexe —> outil de modélisation et prétraitement
3. coût prépondérant —> hybridations à des méthodes d'optimisation

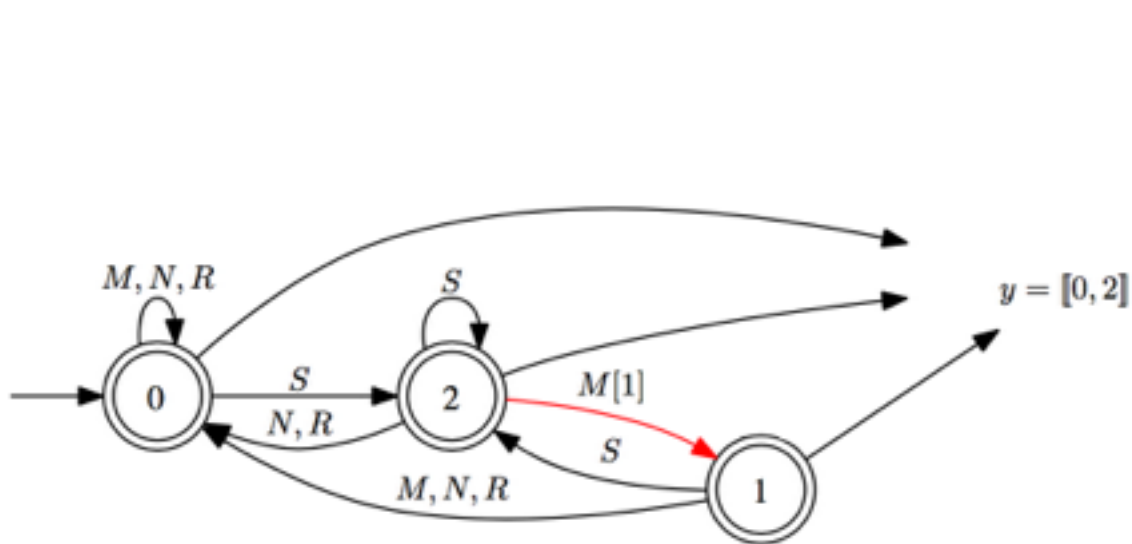
# I. AUTOMATES PONDÉRÉS



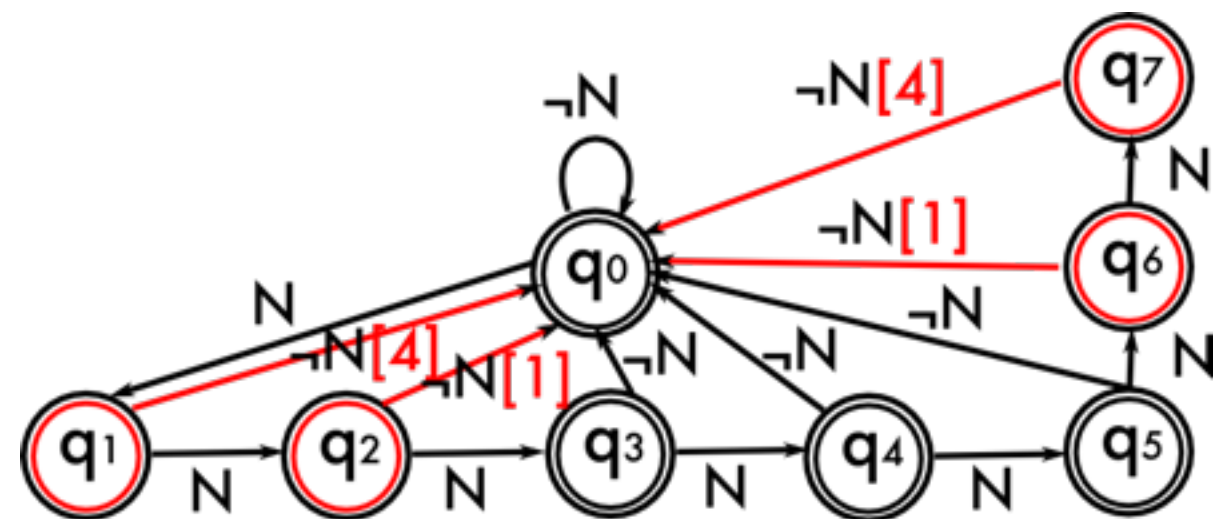
compteurs de valeurs ou de motifs



coûts d'affectation



compteurs de motifs



pénalités de violation

# (MULTI)-COST REGULAR

contraintes d'optimisation: back-propagation

filtrage cost-regular: programmation dynamique SPP graphe en couches

filtrage multicoût-regular: relaxation lagrangienne RCSPP / cost-regular

services pour la composition d'heuristiques de recherche

# 2. AIDE À LA MODÉLISATION

spécialisable à un domaine, ici planification de personnel

génération des automates pondérés à partir des règles de travail

agrégation: intersection des automates + concaténation des poids

Contrat	Analyse	$\sum$ AFD	$\Pi$	Avant	$\Pi_n$
Temps complet	#Nœuds	5782	682	411	230
	# Arcs	40402	4768	1191	400
Temps partiel	#Nœuds	4401	385	791	421
	# Arcs	30729	2689	2280	681



# 3. INTÉGRATION À...

## recherche locale à voisinage large

- voisinage = instantiation partielle
- construction par heuristique de sélection de variables
- exploration par résolution du modèle PPC de planification dans le voisinage

## génération de colonnes

- modèle linéaire de couverture
- modèle PPC de génération des horaires individuels valides de coût réduit négatif
- branch-and-price

# VARIANTES ET DÉRIVÉS

metaheuristiques pour la planification de personnel

CP et musique ou data mining

modèles automates du catalogue de contraintes

linéarisation automatique

contraintes de grammaires hors-contexte et diagrammes de décision

filtrage lagrangien pour atmost-nvalue, tsp,...

# CONTRAINTES GLOBALES



fonctionnel: déclaratif  
réutilisable



explosion du langage



aide à la modélisation  
Global Constraint Catalog  
apprentissage

flexible: composable  
extensible

adaptabilité limitée

implémentation complexe

contraintes polyvalentes:  
schémas de filtrage  
metacontraintes

efficace: spécialisable  
contexte d'hybridation

communication limitée

traitement limité du coût

services composables  
agrégation de contraintes

contraintes d'optimisation  
intégration à des méthodes  
d'optimisation  
contraintes d'optimalité  
relaxation de modèles

# OPTIMISATION pour ÉNERGIE/CLIMAT

# OPTIMISATION ÉNERGIE / CLIMAT

bon timing: maturité des outils - émergence des préoccupations

1. complexité des modèles physiques
2. intrication des échelles temporelles
3. données incertaines

1. contrôle pour l'efficacité énergétique
2. du contrôle au dimensionnement robuste
3. flexibilité des modèles prospectifs

quand la **flexibilité** doit être considérée à hauteur de l'efficacité

DÉCOMPOSITION  
HYBRIDATION  
DÉCLARATIF

pour changer d'échelle

pour affiner les modèles

pour intégrer les incertitudes

# FEATURING

(ordre pseudo-chronologique)

Christian Artigues

Philippe Michelon

Louis-Martin Rousseau

Gilles Pesant

Julien Menana

Narendra Jussien

Nicolas Beldiceanu

Thierry Petit

Mats Carlsson

Fabien Hermenier

Xavier Lorca

Vincent Kherbache

Aurélien Merel

Xavier Gandibleux

Xavier Libeaut

Éric Pinson

Jorge Mendoza

Gilles Chabert

Gratien Bonvin

Nadia Maïzi

Edi Assoumou

Rémy Doudard

Yvann Nzengue

Aurélien Havel

Arnold N’Goran

Dimitra Ignatiadis

Dominique Feillet