

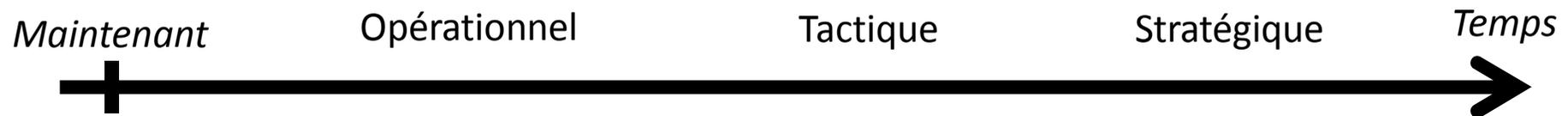
Deux petites histoires en optimisation combinatoire

Philippe Morignot
pmorignot@yahoo.fr

Sommaire

- Introduction : conseils
- Domaine aéroportuaire
- Domaine militaire

Taxonomie des problématiques d'optimisation combinatoire



- Ordonnancer une production industrielle de voitures
 - Construire des plannings de personnel
 - Attribuer des places de parking à des avions
 - Optimiser des tournées logistiques
 - Planifier l'utilisation optimale de ressources rares
- Déterminer la localisation d'entrepôts, les ouvertures et fermetures d'exploitations minières, etc.
 - Dimensionner une flotte aérienne en fonction des prévisions de trafic.

Détecter un problème d'optimisation combinatoire

- Des combinaisons entre objets à trouver ...
 - Des variables dont il faut trouver la valeur
- ... sous contraintes
 - Dures : légales, réglementaires, accords entreprises, ...
 - Molles : préférences
- ... avec un coût / qualité à minimiser / maximiser.
 - Coût : financier, ressources, temporel, etc.
 - Qualité : sociale, économique, etc.

Questions à poser

- En phase de pré-étude :
 - Quelles sont les ressources critiques ? (par ex., grues, personnel, trains, voitures)
 - Où sont les gros flux financiers ? (qu'est ce qui coûte le plus cher, qu'est qui rapporte le plus)
- En phase d'étude :
 - Se ramener à un problème connu (par ex., ordonnancement, plus court chemin, affectation, flôt).
 - Décomposer le problème en éléments simples.
 - Faut il trouver la meilleure solution ou une bonne solution suffit elle ?
 - Quel est le temps de calcul permis ?

Pourquoi vous êtes consulté ?

- Le calcul à la main prend trop de temps ou est fastidieux.
- Faire mieux que l'expert, sans nécessairement reprendre sa méthode.
- Capitaliser sur les connaissances d'un expert.

L'optimisation combinatoire

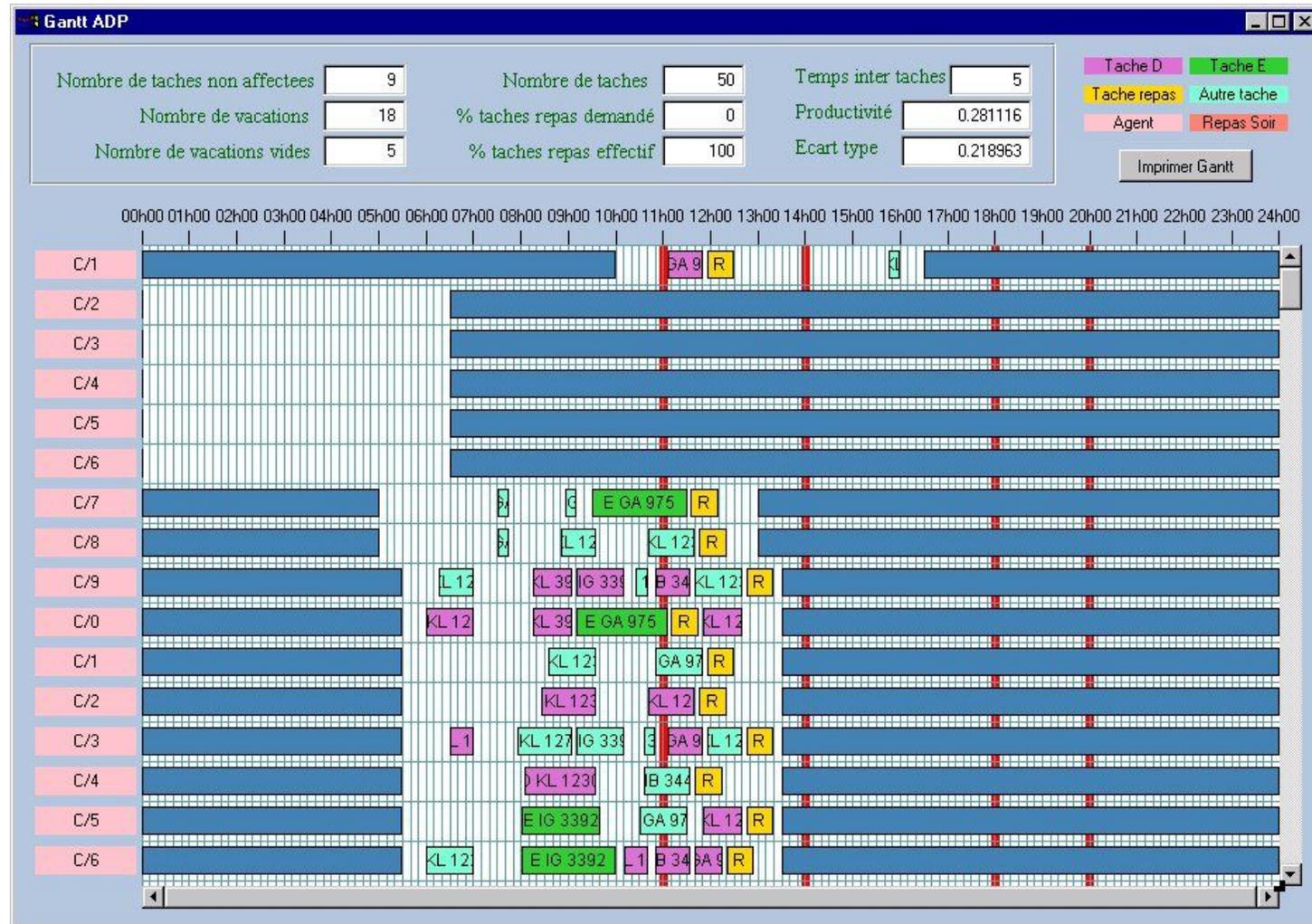
- Recherche opérationnelle : programmation linéaire / mixte, théorie du flût, ...
- Intelligence artificielle : programmation par contraintes, algorithmes de recherche dans un espace d'états, algorithme tabou, algorithme du recuit simulé, algorithmes évolutionnaires, algorithmes de fourmis, ...

Domaine aéroportuaire

Contexte

- Le trafic aérien croît de 5% par an en moyenne.
 - Chaque compagnie aérienne dépose un plan de vol, résultant en tâches devant être assurées à heures fixes par le personnel des aéroports.
- Le personnel au sol dans un aéroport est principalement en nombre fixe.
- Le superviseur a de plus en plus de mal à construire un planning quotidien.
 - Il souhaite construire les plannings six mois à l'avance et pouvoir les rectifier le jour J.

Exemple de planning



Problème

1. Lisser et écrêter la courbe de charge.
2. Affecter les tâches à des vacations, en minimisant les tâches non affectées et en maximisant les tâches repas.
3. Augmenter l'équité sociale (productivité, diversité, etc).
4. Intégrer de nouvelles tâches le jour J.

NB : 1 à 3 en moins d'une minute.

Modèle en Minizinc pour l'affectation

```
int T; // Nombre de tâches
int V; // Nombre de vacations
int T_deb[1..T] = ...; // Date de début d'une tâche
int T_fin[1..T] = ...; // Date de fin d'une tâche
int V_deb[1..V] = ...; // Date de début d'une vacation
int V_fin[1..V] = ...; // Date de fin d'une vacation
int vac[1..T] of var 0..V; // 0 = non affectée
constraint forall(i in 1..N) ( forall(j in 1..V)
  T_fin[i] > V_fin[j] || T_deb[i] < V_deb[j] => vac[i] != j );
constraint forall(i,j in 1..N where i != j)
  ( T_deb[j] < T_fin[i] && T_deb[j] > T_deb[j] ) ||
  ( T_fin[j] > T_deb[i] && T_fin[j] < T_fin[j] ) => vac[i] != vac[j] ;
maximize sum(i in 1..T) ( vac[i] != 0 );
```

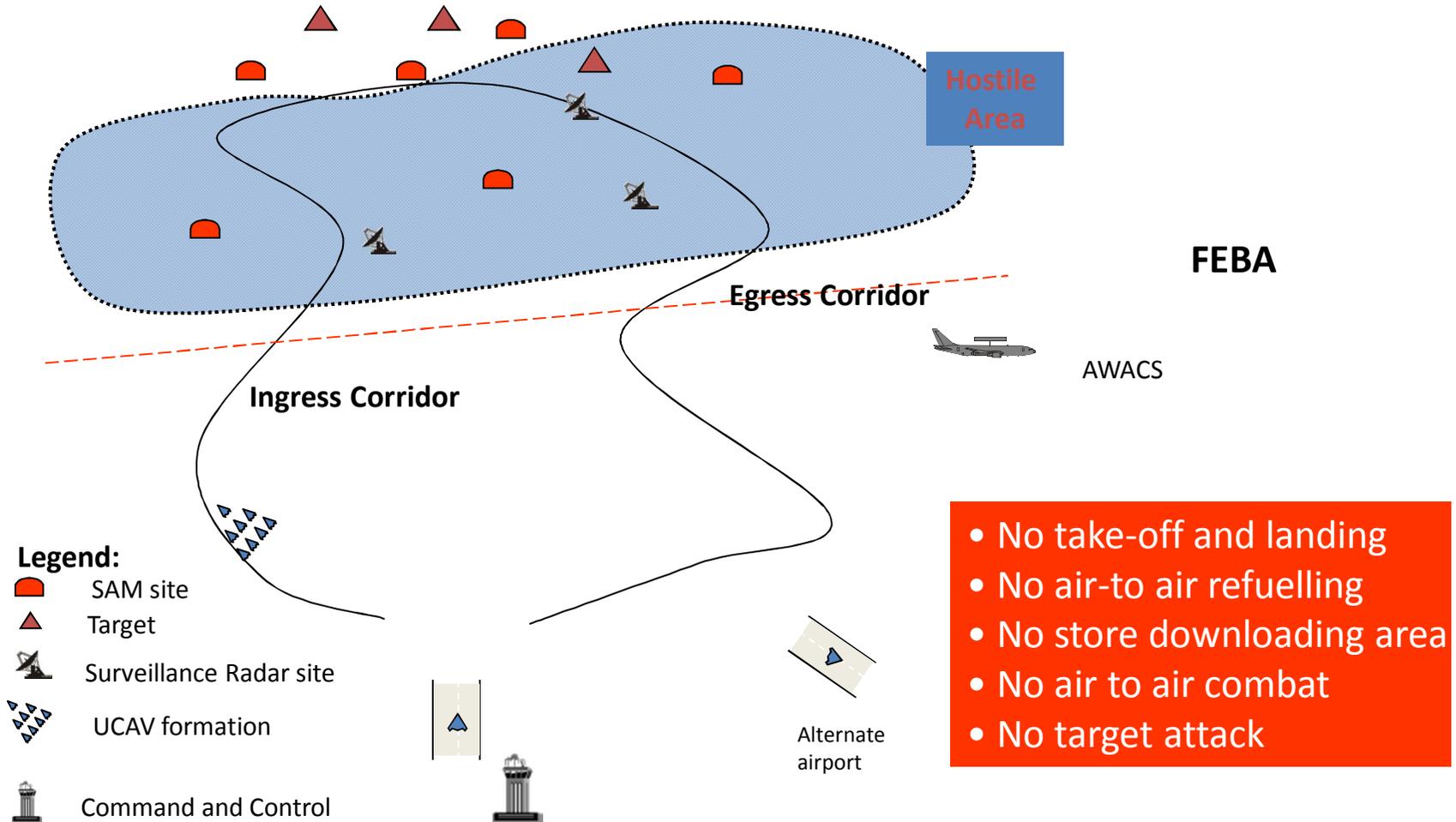
Domaine militaire

Contexte (1 / 2)



- Détruire des cibles peut s'avérer stratégique pour la survie d'un pays.
- Un pilote d'avion de chasse est précieux (des années d'apprentissage et d'entraînement) et faillible (« trou noir »).
- Faire des avions de chasse pilotés par ordinateur.

Contexte (2 / 2)



Problème

- Couloirs de vol.
- Vol en formation, sauf pendant l'attaque.
- Simulation :
 - Des objets au sol : cible, radars, missiles, base, etc.
 - Des organes de chaque avion : communication, capteurs, actionneurs, loi de vol, etc.
 - Des règles de vol en formation: anticollision, empreinte électromagnétique, etc.
- Calculer à l'avance un chemin parmi les couloirs possibles ; le modifier en cours de mission.
- Calculer en cours de route la trajectoire dans chaque couloir.

Analyse

- Recherche dynamique d'un chemin dans un graphe de 60 à 100 nœuds sous contraintes :
 - Le rayon de courbure d'un virage dépend de la vitesse de l'avion.
 - Ne pas larguer toutes ses bombes sur une cible intermédiaire, pour en garder pour la cible principale.
 - Avoir suffisamment de kérosène.
 - Maximiser la probabilité de destruction de la cible et la probabilité de survie de chaque avion.
- Agents autonomes : délibération et réaction.
 - Programmation par contraintes dans une architecture logicielle à deux niveaux.

Résultats

- Domaine aéroportuaire : 600 tâches peuvent être affectées sur 150 vacations en une minute de calcul.
- Domaine militaire : Film résumant la simulation des avions de chasses autonomes (confidentiel défense).

Références

- Domaine aéroportuaire : P. Morignot, L. Somek, C. Miller, B. Gaudinat. *Construire et Optimiser les Emplois du Temps d'Employés d'Aéroports*. In Actes des Journées Nationales sur la Résolution Pratique de Problèmes NP-complets. ONERA, Toulouse, juin 2001, pages 209-219.
- Domaine militaire : [Johan Baltié](#), [Eric Bensana](#), [Patrick Fabiani](#), [Jean- Loup Farges](#), Stéphane Millet, Philippe Morignot, [Bruno Patin](#), [Gérald Petitjean](#), [Gauthier Pitois](#), [Jean- Clair Poncet](#). *Multi-Vehicle Missions: Architecture and Algorithms for Distributed On Line Planning*. Chapter 1. In Dimitris Vrakas and Ioannis Vlahavas (eds.), [Artificial Intelligence for Advanced Problem Solving Techniques](#), Information Science Reference. December 2007.

Merci pour votre attention !