University of BRISTOL

Aerospace Controls Lab

Mixed-integer Programming for Control

Arthur Richards

and

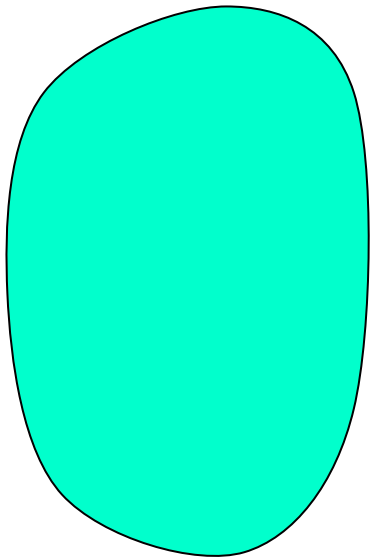
Jonathan How

Motivation

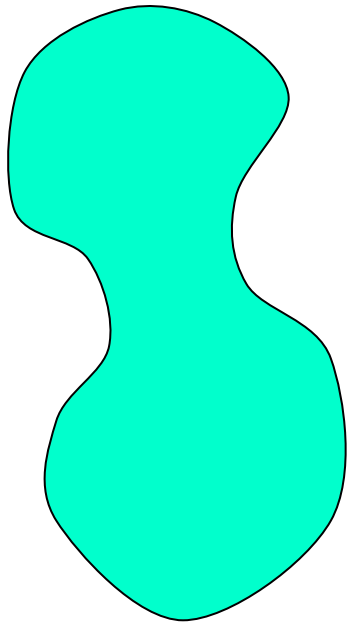
- What is Mixed-integer Programming?
  - MIP is an optimization method that combines continuous and discrete variables
- Why is it useful?
  - MIP can model complex planning and control problems involving both continuous and discrete decisions
- Why now? Is MIP new?
  - MIP is not a new concept, BUT online use has only arrived with fast computers and software

Problem Classes

Nonlinear Convex



Non-convex



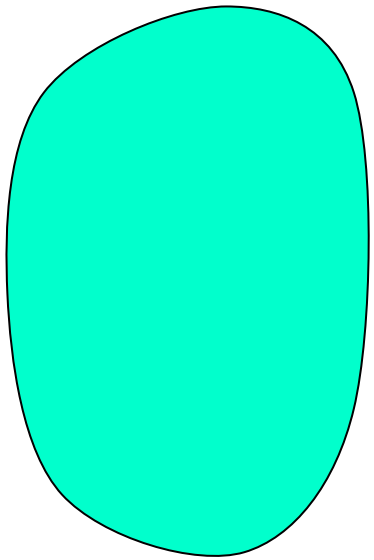
Logical

	T	F
T	✗	✓
F	✓	✓

Problem Classes

Nonlinear

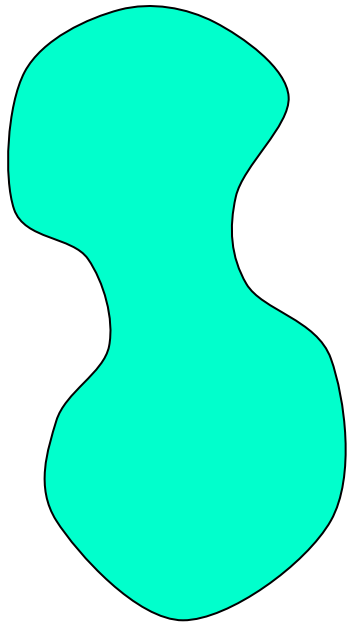
Convex



Nonlinear opt.

No MIP

Non-convex



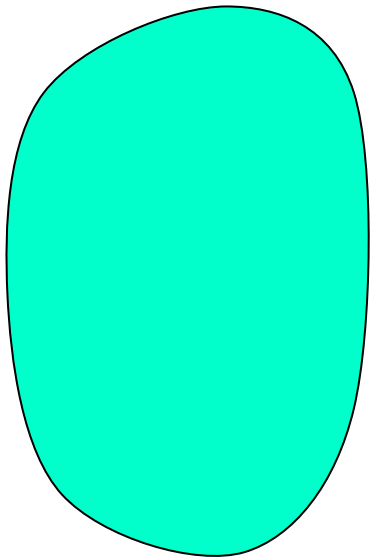
Logical

	T	F
T	✗	✓
F	✓	✓

Problem Classes

Nonlinear

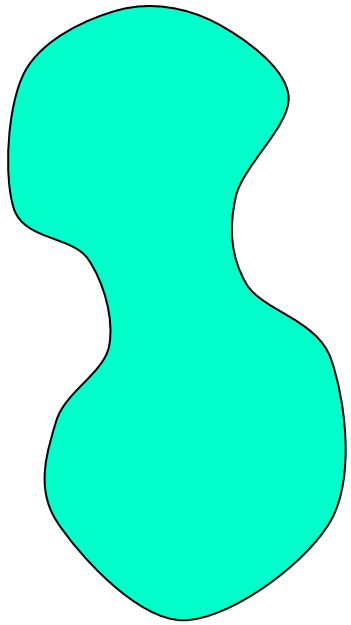
Convex



Nonlinear opt.

No MIP

Non-convex



Logical

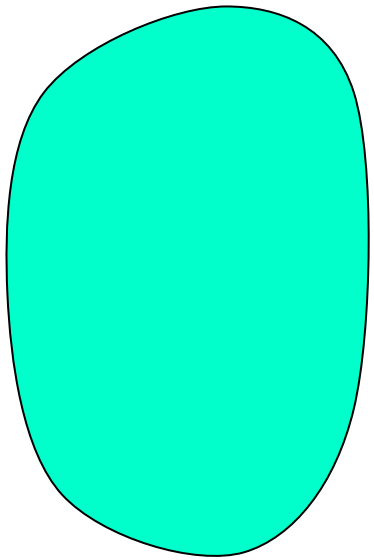
	T	F
T	✗	✓
F	✓	✓

MIP

Problem Classes

Nonlinear

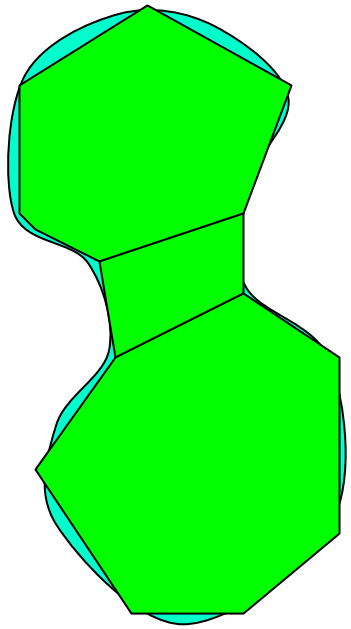
Convex



Nonlinear opt.

No MIP

Non-convex



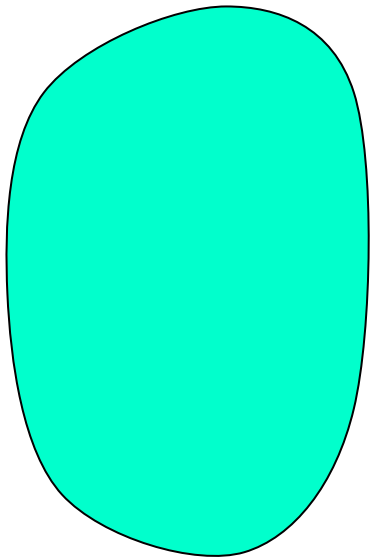
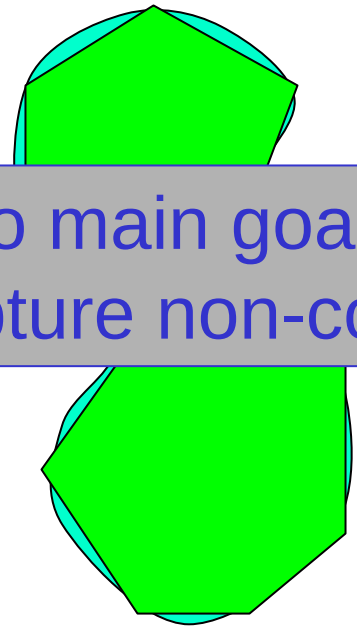
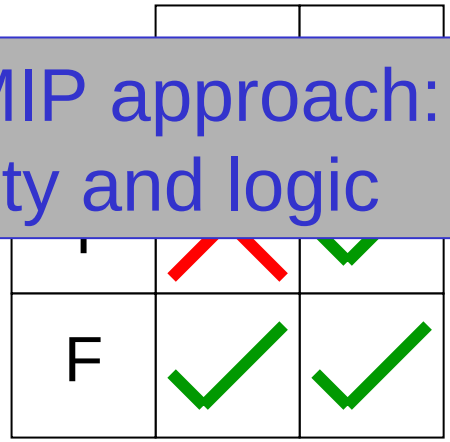
Approximate by MIP

Logical

	T	F
T	✗	✓
F	✓	✓

MIP

Problem Classes

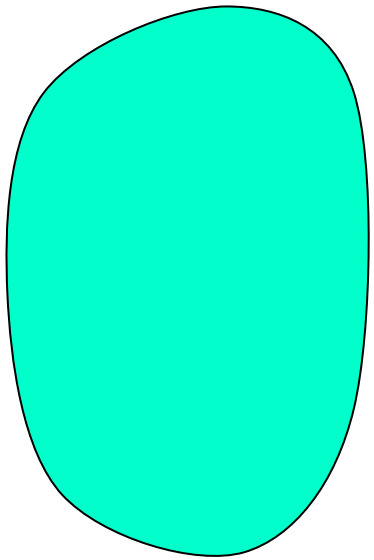
Nonlinear	Non-convex	Logical
Convex		
		
Nonlinear opt. No MIP	Approximate by MIP	MIP

Two main goals of MIP approach: capture non-convexity and logic

Problem Classes

Nonlinear

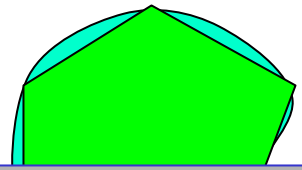
Convex



Nonlinear opt.

No MIP

Non-convex



Logical

Two main goals of MIP approach: capture non-convexity and logic

- Encode non-convexity using integers
- Then apply powerful integer optimization tools to non-convex problems

Session Outline (1/2)

1. Introduction to MIP
   - MIP definition
   - Modeling
     - Assignment
     - Non-convex constraints
     - Piecewise affine systems
   - Solving MIP
   - Using optimization for feedback control
   - Techniques for online solution of MIP
   - Examples

Session Outline (2/2)

2. Projected Variable Metric Algorithm

3. MILP Assignment for Multi-Vehicle Systems

4. Real-time Path-Planning for Tactical UAV

5. Receding Horizon Implementation of MILP for Vehicle Guidance

Web Resources

- Slides and sample codes available online
- Web resources

acl.mit.edu/MILP

seis.bris.ac.uk/~aeagr

- Email

arthur.richards@bristol.ac.uk

jhow@mit.edu

Formal Definition

- Mixed-integer Linear Program (MILP)

$$\begin{aligned} \min_{\mathbf{x},\mathbf{z}} \quad & \mathbf{f}_1^T\mathbf{x} + \mathbf{f}_2^T\mathbf{z} \\ \text{subject to} \quad & \mathbf{A}_1\mathbf{x} + \mathbf{A}_2\mathbf{z} \leq \mathbf{b} \\ & \mathbf{z} \text{ integer} \end{aligned}$$

- Inherently non-convex
- NP-complete

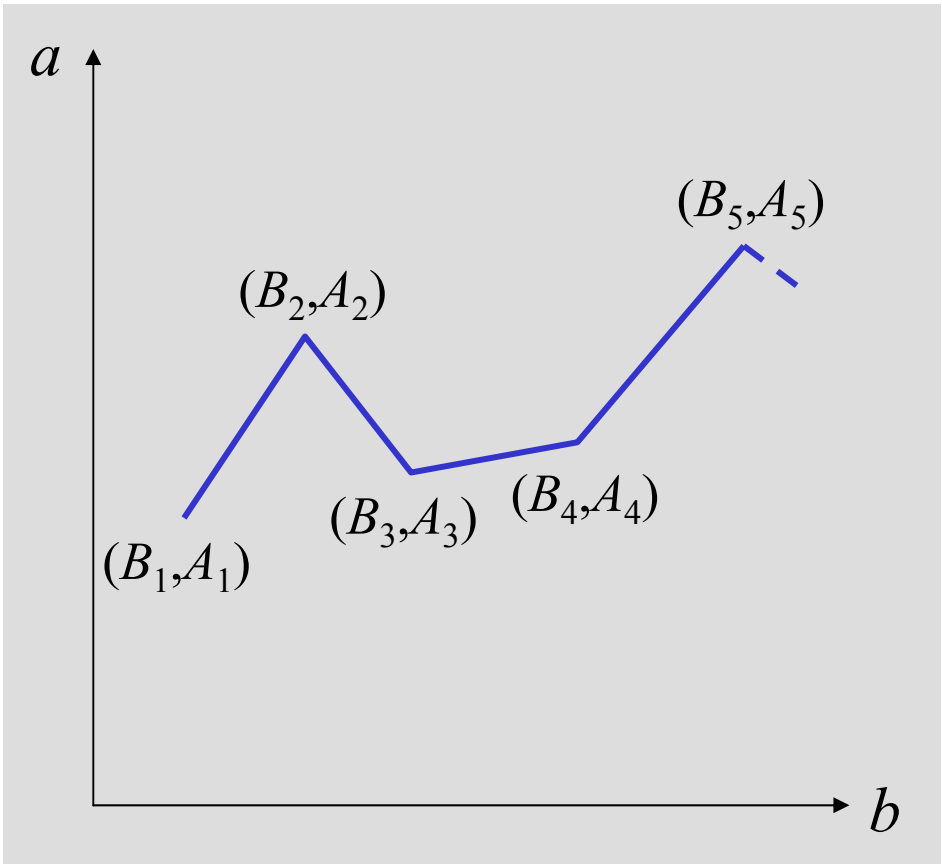
BUT with good software, can find global-optimum in many useful instances

Modelling using MIP

MIP Modelling 1: PWA

[Example from Bersimas and Tsitsiklis]

- Constrain $a = f(b)$ where $f(.)$ is PWA



$$a = \sum_{i=1}^{N} x_i A_i$$

$$b = \sum_{i=1}^{N} x_i B_i$$

$$\sum_{i=1}^{N} x_i = 1$$

$$x_i \geq 0 \ \forall i \in \{1 \dots N\}$$

$$x_1 \leq z_1$$

$$x_i \leq z_{i-1} + z_i \ \forall i \in \{2 \dots N-1\}$$

$$x_N \leq z_{N-1}$$

$$\sum_{i=1}^{N-1} z_i = 1 \,, \ z_i \in \{0, 1\}$$

Binary $z_i = 1$ if $b \in [A_i, A_{(i+1)}]$

MIP Modelling 1: PWA

[Example from Bersimas and Tsitsiklis]

- Constrain $a = f(b)$ where $f(.)$ is PWA

$$a = \sum_{i=1}^{N} x_i A_i$$

$$b = \sum_{i=1}^{N} x_i B_i$$

e.g. choose $z_3 = 1$
all other $z_i = 0$

$$\sum_{i=1}^{N} x_i = 1$$

$$x_i \geq 0 \ \forall i \in \{1 \dots N\}$$

$$x_1 \leq z_1$$

$$x_i \leq z_{i-1} + z_i \ \forall i \in \{2 \dots N-1\}$$

$$x_N \leq z_{N-1}$$

$$\sum_{i=1}^{N-1} z_i = 1 \,, \ z_i \in \{0, 1\}$$

Binary $z_i = 1$ if $b \in [A_i, A_{(i+1)}]$

MIP Modelling 1: PWA

[Example from Bersimas and Tsitsiklis]

- Constrain $a = f(b)$ where $f(.)$ is PWA

$$a = \sum_{i=1}^{N} x_i A_i$$

$$b = \sum_{i=1}^{N} x_i B_i$$

e.g. choose $z_3 = 1$
all other $z_i = 0$

$$\sum_{i=1}^{N} x_i = 1$$

$$x_i \geq 0 \ \forall i \in \{1 \dots N\}$$

$$x_1 \leq z_1$$

$$x_i \leq z_{i-1} + z_i \ \forall i \in \{2 \dots N-1\}$$

$$x_N \leq z_{N-1}$$

$$\sum_{i=1}^{N-1} z_i = 1 \,, \ z_i \in \{0,1\}$$

✓

$$\sum_{i=3}^{4} x_i = 1$$

$$x_i \geq 0 \ \forall i \in \{1 \dots N\}$$

$$x_3 \leq 1$$

$$x_4 \leq 1$$

$$x_i = 0 \ \forall i \neq 3,4$$

Binary $z_i = 1$ if $b \in [A_i, A_{(i+1)}]$

MIP Modelling 1: PWA

[Example from Bersimas and Tsitsiklis]

- Constrain $a = f(b)$ where $f(.)$ is PWA

$$a = \sum_{i=1}^{N} x_i A_i$$
$$b = \sum_{i=1}^{N} x_i B_i$$

e.g. choose $z_3 = 1$
all other $z_i = 0$

$$\sum_{i=1}^{N} x_i = 1$$
$$x_i \geq 0 \ \forall i \in \{1 \dots N\}$$
$$x_1 \leq z_1$$
$$x_i \leq z_{i-1} + z_i \ \forall i \in \{2 \dots N-1\}$$
$$x_N \leq z_{N-1}$$
$$\sum_{i=1}^{N-1} z_i = 1 \,,\ z_i \in \{0,1\}$$

Binary $z_i = 1$ if $b \in [A_i, A_{(i+1)}]$

$$a = \sum_{i=3}^{4} x_i A_i$$
$$b = \sum_{i=3}^{4} x_i B_i$$

$$\sum_{i=3}^{4} x_i = 1$$
$$x_i \geq 0 \ \forall i \in \{1 \dots N\}$$
$$x_3 \leq 1$$
$$x_4 \leq 1$$
$$x_i = 0 \ \forall i \neq 3,4$$

Interpolation over $[A_3, A_4]$

MIP Modelling 2: Disjunction

[Bersimas and Tsitsiklis. Morari. Brailsford and Williams]

- Require either $\mathbf{a}_1^T\mathbf{x} \leq b_1$ or $\mathbf{a}_2^T\mathbf{x} \leq b_2$

- MILP form

$$\begin{aligned} \mathbf{a}_1^T\mathbf{x} &\leq b_1 + Mz_1 \\ \mathbf{a}_2^T\mathbf{x} &\leq b_2 + Mz_2 \\ z_1 + z_2 &\leq 1, \quad z_i \in \{0,1\} \end{aligned}$$

- $M > 0, \quad M >> \mathbf{a}_1\mathbf{x} - b_1, \quad M >> \mathbf{a}_2\mathbf{x} - b_2$
- Known as “big-M” method

MIP Modelling 2: Disjunction

[Bersimas and Tsitsiklis. Morari. Brailsford and Williams]

- Require either $\mathbf{a}_1^T\mathbf{x} \leq b_1$ or $\mathbf{a}_2^T\mathbf{x} \leq b_2$

e.g. choose $z_1 = 1$

- MILP form

$$\begin{aligned} \mathbf{a}_1^T\mathbf{x} &\leq b_1 + Mz_1 \\ \mathbf{a}_2^T\mathbf{x} &\leq b_2 + Mz_2 \\ z_1 + z_2 &\leq 1, \quad z_i \in \{0,1\} \end{aligned}$$

→ requires $z_2 = 0$

- $M > 0, \quad M >> \mathbf{a}_1\mathbf{x} - b_1, \quad M >> \mathbf{a}_2\mathbf{x} - b_2$
- Known as “big-M” method

MIP Modelling 2: Disjunction

[Bersimas and Tsitsiklis. Morari. Brailsford and Williams]

- Require either $\mathbf{a}_1^T\mathbf{x} \leq b_1$ or $\mathbf{a}_2^T\mathbf{x} \leq b_2$

- MILP form

$$\begin{aligned} \mathbf{a}_1^T\mathbf{x} &\leq b_1 + Mz_1 \\ \mathbf{a}_2^T\mathbf{x} &\leq b_2 + Mz_2 \\ z_1 + z_2 &\leq 1, \quad z_i \in \{0,1\} \end{aligned}$$

e.g. choose $z_1 = 1$

$$\begin{aligned} \mathbf{a}_1^T\mathbf{x} &\leq b_1 + M \\ \mathbf{a}_2^T\mathbf{x} &\leq b_2 \end{aligned}$$

requires $z_2 = 0$

- $M > 0, \quad M >> \mathbf{a}_1\mathbf{x} - b_1, \quad M >> \mathbf{a}_2\mathbf{x} - b_2$
- Known as “big-M” method

MIP Modelling 2: Disjunction

[Bersimas and Tsitsiklis. Morari. Brailsford and Williams]

- Require either $\mathbf{a}_1^T\mathbf{x} \leq b_1$ or $\mathbf{a}_2^T\mathbf{x} \leq b_2$

- MILP form

$$\begin{aligned} \mathbf{a}_1^T\mathbf{x} &\leq b_1 + Mz_1 \\ \mathbf{a}_2^T\mathbf{x} &\leq b_2 + Mz_2 \\ z_1 + z_2 &\leq 1, \quad z_i \in \{0,1\} \end{aligned}$$

e.g. choose $z_1 = 1$

$$\mathbf{a}_2^T\mathbf{x} \leq b_2$$

requires $z_2 = 0$

- $M > 0, \quad M >> \mathbf{a}_1\mathbf{x} - b_1, \quad M >> \mathbf{a}_2\mathbf{x} - b_2$
- Known as “big-M” method

MIP Modelling 2: Disjunction

[Bersimas and Tsitsiklis. Morari. Brailsford and Williams]

- Require either $\mathbf{a}_1^T\mathbf{x} \leq b_1$ or $\mathbf{a}_2^T\mathbf{x} \leq b_2$

OR choose $z_2 = 1$

- MILP form

$$\begin{aligned} \mathbf{a}_1^T\mathbf{x} &\leq b_1 + Mz_1 \\ \mathbf{a}_2^T\mathbf{x} &\leq b_2 + Mz_2 \\ z_1 + z_2 &\leq 1, \quad z_i \in \{0,1\} \end{aligned}$$

$$\mathbf{a}_1^T\mathbf{x} \leq b_1$$

requires $z_1 = 0$

- $M > 0, \quad M >> \mathbf{a}_1\mathbf{x} - b_1, \quad M >> \mathbf{a}_2\mathbf{x} - b_2$
- Known as “big-M” method

MIP Modelling 3: Avoidance

[Schouwenaars *et al*, Richards *et al*]

- Similar to disjunction
- Point (x, y) must be outside obstacle

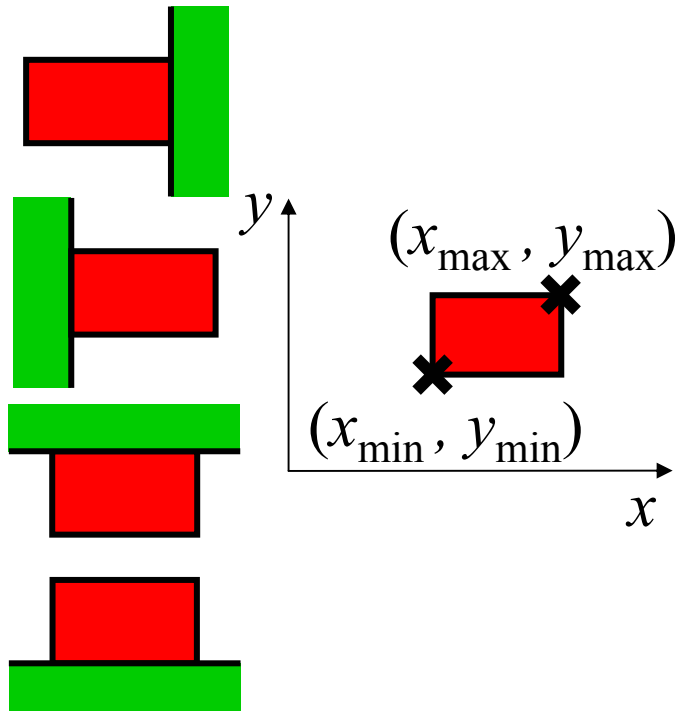
$$
\begin{aligned}
x - x_{\max} &\geq -Mc_1 \\
\text{and } x_{\min} - x &\geq -Mc_2 \\
\text{and } y - y_{\max} &\geq -Mc_3 \\
\text{and } y_{\min} - y &\geq -Mc_4 \\
\text{and } \sum_{k=1}^{4} c_k &\leq 3
\end{aligned}
$$

$c_1=0 \Rightarrow$

$c_2=0 \Rightarrow$

$c_3=0 \Rightarrow$

$c_4=0 \Rightarrow$



MIP Modelling 4: Assignment

- Assign N tasks to N agents.
- Cost of assigning agent i to task j is c_{ij}

$$\begin{aligned} \min_{\mathbf{z}} \quad & \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij} z_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{N} z_{ij} = 1 \forall j \in \{1 \dots N\} \\ & \sum_{j=1}^{N} z_{ij} = 1 \forall i \in \{1 \dots N\} \\ & z_{ij} \in \{0,1\} \forall i,j \end{aligned}$$

Binary $z_{ij} = 1$ if agent i assigned to task j

- Special case – LP works
  - Constrain $0 \leq z_{ij} \leq 1$: all vertices are integer

MIP Modelling 4: Assignment

- Assign N tasks to N agents.
- Cost of assigning agent i to task j is c_{ij}

$$\begin{aligned} \min_{\mathbf{z}} \quad & \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij} z_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{N} z_{ij} = 1 \forall j \in \{1 \dots N\} \\ & \sum_{j=1}^{N} z_{ij} = 1 \forall i \in \{1 \dots N\} \\ & z_{ij} \in \{0,1\} \forall i,j \end{aligned}$$

Binary $z_{ij} = 1$ if agent i assigned to task j

- Add resource constraint – MILP needed

$$\sum_{i=1}^{N}\sum_{j=1}^{N} r_{ij} z_{ij} \leq R$$

MIP Modelling 5: Modes

- System has two modes

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\mathbf{u}(k) \qquad \text{Mode 1}$$

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_2\mathbf{u}(k) \qquad \text{Mode 2}$$

$$\|\mathbf{u}(k)\| \leq 1$$

MIP Modelling 5: Modes

- System has two modes

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\mathbf{u}(k) \qquad \text{Mode 1}$$

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_2\mathbf{u}(k) \qquad \text{Mode 2}$$

$$\|\mathbf{u}(k)\| \leq 1$$

- MIP representation

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\mathbf{u}_1(k) + \mathbf{B}_2\mathbf{u}_2(k)$$

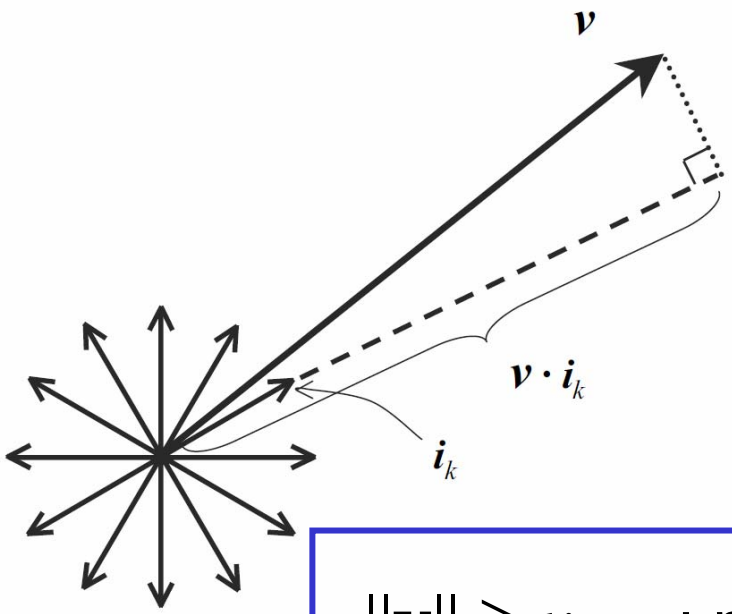
$$\|\mathbf{u}_1(k)\| \leq z_1$$

$$\|\mathbf{u}_1(k)\| \leq z_2$$

$$z_1 + z_2 = 1, \;\; z_1, z_2 = \{0, 1\}$$

MIP Modelling 6 : Speed Limits

- 2-norm approximation



$\|\mathbf{v}\| \leq v_{\max}$: convex, easily handled

$$\boldsymbol{v} \cdot \boldsymbol{i}_k \leq v_{\max}, \qquad k = 1, \ldots, n_{v_{\max}}$$

$$\boldsymbol{i}_k = \begin{bmatrix} \cos\left(\dfrac{2\pi k}{n_{v_{\max}}}\right) \\ \sin\left(\dfrac{2\pi k}{n_{v_{\max}}}\right) \end{bmatrix}$$

$\|\mathbf{v}\| \geq v_{\min}$: non-convex, needs binaries

$$\boldsymbol{v} \cdot \boldsymbol{i}_k \geq v_{\min}, \qquad \exists k,$$

$$\boldsymbol{v} \cdot \boldsymbol{i}_k \geq v_{\min} - M_v(1 - b_{\text{speed},k}),$$

$$\sum_{k=1}^{n_{v_{\min}}} b_{\text{speed},k} \geq 1$$

$$\boldsymbol{i}_k = \begin{bmatrix} \cos\left(\dfrac{2\pi k}{n_{v_{\min}}}\right) \\ \sin\left(\dfrac{2\pi k}{n_{v_{\min}}}\right) \end{bmatrix}$$

MIP Modelling: Remarks

- Examples span many problem classes
  - Combinations and extensions possible
    - Joint assignment/path planning with avoidance
    - PWA systems with disjunction constraints
    - Logical constraints – “if A and B then C”
- There are often multiple ways of expressing a problem using MIP
  - Rule of thumb: big-M is nearly always an option, but look for something better
  - “Most existing MILP formulations that employ big-M constraints do suffer from the poor relaxation (relaxed MILP), which is a notorious feature of big-M.”
    - Improving Mixed Integer Linear Programming Formulations A. Khurana, A. Sundaramoorthy and I. Karimi, AIChE, 2005

Modelling References

- C. Floudas *Nonlinear and Mixed-Integer Programming - Fundamentals and Applications* Oxford University Press, 1995.

- A. Bemporad and M. Morari “Control of systems integrating logic, dynamics, and constraints,” *Automatica*, 35:407-427, 1999

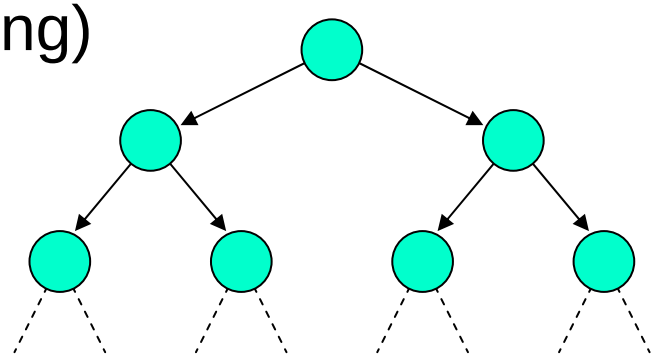
- H. Williams and S. Brailsford, ``Computational Logic and Integer Programming," in *Advances in Linear and Integer Programming*, Editor J. E. Beasley, Clarendon Press, 1996, pp.249-281.

- D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, 1997.

MIP Solution

MIP Solution: Branch & Bound

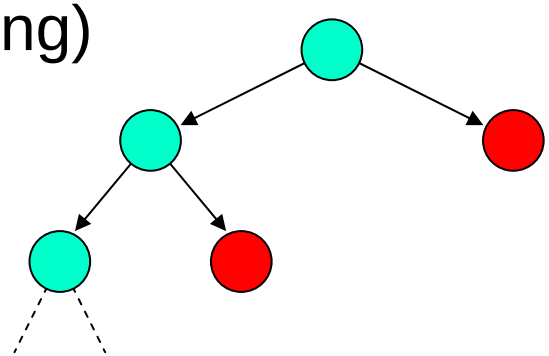
- Finds **global optimum** by tree search
  1. Relax binary constraints $z_i \in \{0,1\} \rightarrow 0 \leq z_i \leq 1$
  2. Solve relaxed problem (bounding)
  3. Choose an i and (branching)
     a) Fix $z_i = 0$; go to 2;
     b) Fix $z_i = 1$; go to 2;



- Recursive tree search of binary options

MIP Solution: Branch & Bound

- Finds **global optimum** by tree search
  1. Relax binary constraints $z_i \in \{0,1\} \rightarrow 0 \leq z_i \leq 1$
  2. Solve relaxed problem (bounding)
  3. Choose an i and (branching)
     a) Fix $z_i = 0$; go to 2;
     b) Fix $z_i = 1$; go to 2;



- Recursive tree search of binary options
- Can stop "early" by fathoming, if relaxation
  - is infeasible, or
  - gives binary result, or
  - has worse cost than best binary so far.

MIP Solution: AMPL

- AMPL easily translates models
  - A Mathematical Programming Language
  - Helps sort out indexing
  - Interfaced to many solver codes

PWA example

$$a = \sum_{i=1}^{N} x_i A_i$$

$$b = \sum_{i=1}^{N} x_i B_i$$

$$\sum_{i=1}^{N} x_i = 1$$

$$x_i \geq 0 \ \forall i \in \{1 \dots N\}$$

$$x_1 \leq z_1$$

$$x_i \leq z_{i-1} + z_i \ \forall i \in \{2 \dots N-1\}$$

$$x_N \leq z_{N-1}$$

$$\sum_{i=1}^{N-1} z_i = 1, \ z_i \in \{0,1\}$$

```
var x{i in 1..N};
var z{i in 1..(N-1)} binary;
subject to acon: a = sum{i in 1..N} A[i]*x[i];
subject to bcon: b = sum{i in 1..N} B[i]*x[i];
subject to xsum: sum{i in 1..N} x[i] = 1;
subject to xpos{i in 1..N}: x[i] >= 0;
subject to x1: x[1] <= z[1];
subject to xi{i in 1..(N-1)}: x[i] <= z[i-1] + z[i];
subject to xN: x[N] <= z[N-1];
subject to zsum: sum{i in 1..(N-1)} = 1;
```

MIP Solution: CPLEX

- Commercial solver code from ILOG
- Implements branch-and-bound in conjunction with tried and tested branching heuristics
- Interfaces
  - AMPL
  - Matlab MEX
  - C API

Other Software

- Decision tree for Optimization with discrete variables Online
- Modeling examples: MILP Model for Short Term Scheduling of Multistage Batch Plants by Grossmann et al. Online
- Good comparison of non-commercial MILP software:
- MINOPT: A Modeling Language and Algorithmic Framework for Linear, Mixed-Integer, Nonlinear, Dynamic, and Mixed-Integer Nonlinear Optimization by Floudas et al. Online
- The Hybrid Systems Group
  - Multi-Parametric Toolbox.
- Interface Software and example (Matlab ←→ AMPL ←→ CPLEX)
- AMPL: R. Fourer, D. M. Gay, and B. W. Kernighar, AMPL, *A modeling language for mathematical programming*, The Scientific Press, 1993.

MIP Solution: Remarks

- AMPL is good for prototyping, but more direct interfaces are faster
- Other discrete optimization tools are available
  - Heuristic methods for general MIP
    - Genetic algorithms
    - Simulated annealing
  - Special cases
    - Dynamic programming for knapsack problem

MIP for Control

Control using Optimization

- Have seen that MIP can find optimal solutions for complex planning problems
- "Control" also considers uncertainty
  - Introduce feedback to compensate
  - Update plans to include new information
- Concept is the same as Model Predictive Control (MPC)
  1. Use numerical optimization to design an open-loop control sequence for the future
  2. Execute some initial portion of that sequence
  3. Go to 1.

MIP and MPC

- MPC optimization has three key features
  - Dynamics model $\mathbf{x}(k+1) = f\left(\mathbf{x}(k), \mathbf{u}(k)\right)$
  - Constraints $\mathbf{x}(k) \in \mathcal{X},\ \mathbf{u}(k) \in \mathcal{U}$
  - Cost $J^*(k) = \min \sum_{k=0}^{N} g\left(\mathbf{x}(k), \mathbf{u}(k)\right)$
- MIP can enter all three
  - PWA dynamics, logic states (e.g. ‘task done’)
  - Non-convex constraints (e.g. avoidance)
  - PWA cost

Properties of MPC

- Start with a recursion
  - Solution is $\{\mathbf{u}(k_0|k_0)\ \mathbf{u}(k_0+1|k_0)\ \ldots\ \mathbf{u}(k_0+N|k_0)\}$ at k_0
  - Then $\{\mathbf{u}(k_0+1|k_0)\ \ldots\ \mathbf{u}(k_0+N|k_0)\ ???\}$ feasible at k_0+1

Properties of MPC

- Start with a recursion
  - Solution is $\{\mathbf{u}(k_0|k_0)\ \mathbf{u}(k_0+1|k_0)\ \ldots\ \mathbf{u}(k_0+N|k_0)\}$ at k_0
  - Then $\{\mathbf{u}(k_0+1|k_0)\ \ldots\ \mathbf{u}(k_0+N|k_0)\ ???\}$ feasible at k_0+1

Properties of MPC

- Start with a recursion
  - Solution is $\{\mathbf{u}(k_0|k_0)\ \mathbf{u}(k_0+1|k_0)\ \ldots\ \mathbf{u}(k_0+N|k_0)\}$ at k_0
  - Then $\{\mathbf{u}(k_0+1|k_0)\ \ldots\ \mathbf{u}(k_0+N|k_0)\ ???\}$ feasible at k_0+1

- Use recursion to prove cost decrease
  - $J(k+1) \leq J(k) - a(\mathbf{x}(k))$
- Use cost J as Lyapunov function
  - Stage cost must be positive definite in $\mathbf{x}$

Properties of MPC

- Start with a recursion
  - Solution is $\{\mathbf{u}(k_0|k_0)\ \mathbf{u}(k_0+1|k_0)\ \ldots\ \mathbf{u}(k_0+N|k_0)\}$ at k_0
  - Then $\{\mathbf{u}(k_0+1|k_0)\ \ldots\ \mathbf{u}(k_0+N|k_0)\ ???\}$ feasible at k_0+1

Very general results, easily handling MIP dynamics, constraints and cost [Bemporad and Morari, 1999]

- Use cost J as Lyapunov function
  - Stage cost must be positive definite in $\mathbf{x}$

MIP/MPC Example: Modes

- Recall earlier example

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\mathbf{u}(k) \qquad \text{Mode 1}$$

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}_2\mathbf{u}(k) \qquad \text{Mode 2}$$

$$\|\mathbf{u}(k)\| \leq 1$$

MIP/MPC Example: Modes

1. Optimize $\min_{\mathbf{u}_1,\mathbf{u}_2,\mathbf{x},\mathbf{z}} \sum_{j=0}^{N} \left(|\mathbf{u}_1(k+j|k)| + |\mathbf{u}_2(k+j|k)| + |\mathbf{x}(k+j|k)|\right)$

$$
\begin{aligned}
&\text{subject to } \forall j \in \{0 \dots N\} \\
\mathbf{x}(k+j+1|k) &= \mathbf{A}\mathbf{x}(k+j|k) + \mathbf{B}_1\mathbf{u}_1(k+j|k) + \mathbf{B}_2\mathbf{u}_2(k+j|k) \\
\mathbf{x}(k|k) &= \mathbf{x}(k) \\
\mathbf{x}(k+N+1|k) &= \mathbf{0} \\
\|\mathbf{u}_1(k+j|k)\| &\leq z_1(k+j|k) \\
\|\mathbf{u}_2(k+j|k)\| &\leq z_2(k+j|k) \\
z_1(k+j|k) + z_2(k+j|k) &= 1 \\
z_i(k+j|k) &\in \{0,1\}
\end{aligned}
$$

2. Apply $\mathbf{u}(k) = \mathbf{u}^*(k|k)$

3. Repeat

MIP/MPC Example: Modes

1. Optimize $\min_{\mathbf{u}_1,\mathbf{u}_2,\mathbf{x},\mathbf{z}} \sum_{j=0}^{N} (|\mathbf{u}_1(k+j|k)| + |\mathbf{u}_2(k+j|k)| + |\mathbf{x}(k+j|k)|)$

Cost

$$\text{subject to } \forall j \in \{0 \dots N\}$$

$$\begin{aligned}
\mathbf{x}(k+j+1|k) &= \mathbf{A}\mathbf{x}(k+j|k) + \mathbf{B}_1\mathbf{u}_1(k+j|k) + \mathbf{B}_2\mathbf{u}_2(k+j|k) \\
\mathbf{x}(k|k) &= \mathbf{x}(k) \\
\mathbf{x}(k+N+1|k) &= \mathbf{0} \\
\|\mathbf{u}_1(k+j|k)\| &\leq z_1(k+j|k) \\
\|\mathbf{u}_2(k+j|k)\| &\leq z_2(k+j|k) \\
z_1(k+j|k) + z_2(k+j|k) &= 1 \\
z_i(k+j|k) &\in \{0,1\}
\end{aligned}$$

Dynamics

Initial condition

Terminal

Control limits

Mode logic

2. Apply $\mathbf{u}(k) = \mathbf{u}^*(k|k)$

3. Repeat

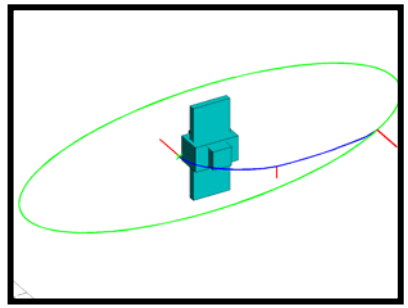
Using MIP Online

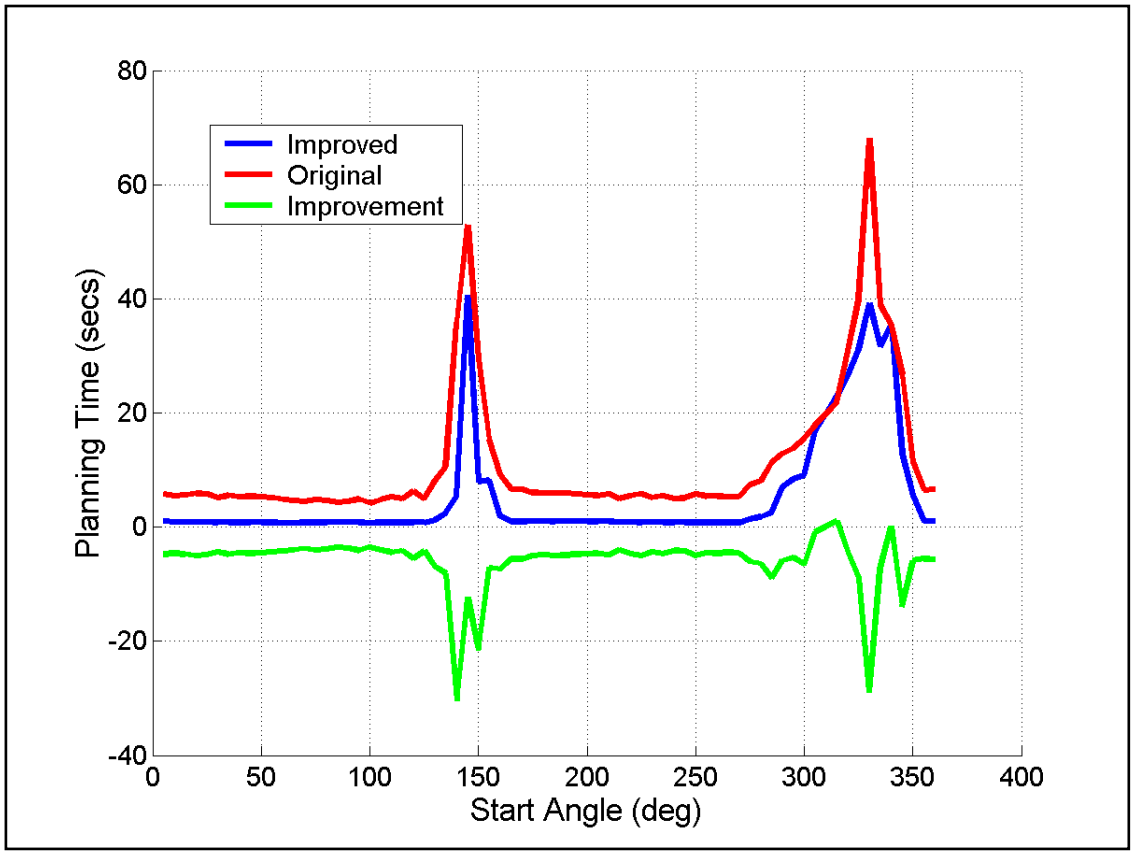
Computation Time

- Need to solve MIP ***online***
  - Good computer and solver handle most cases
  - ***Still NP-hard problem*** – some instances of large problems can be slow
- Tricks to accelerate solution time are often ***problem-specific***
  - More to follow in example talk
- Three general approaches
  - Use prior knowledge
  - Approximate cost-to-go
  - Multi-parametric integer programming

Prior Knowledge I : Iteration

Space Station Rendezvous with Plume Impingement (PI) Constraints

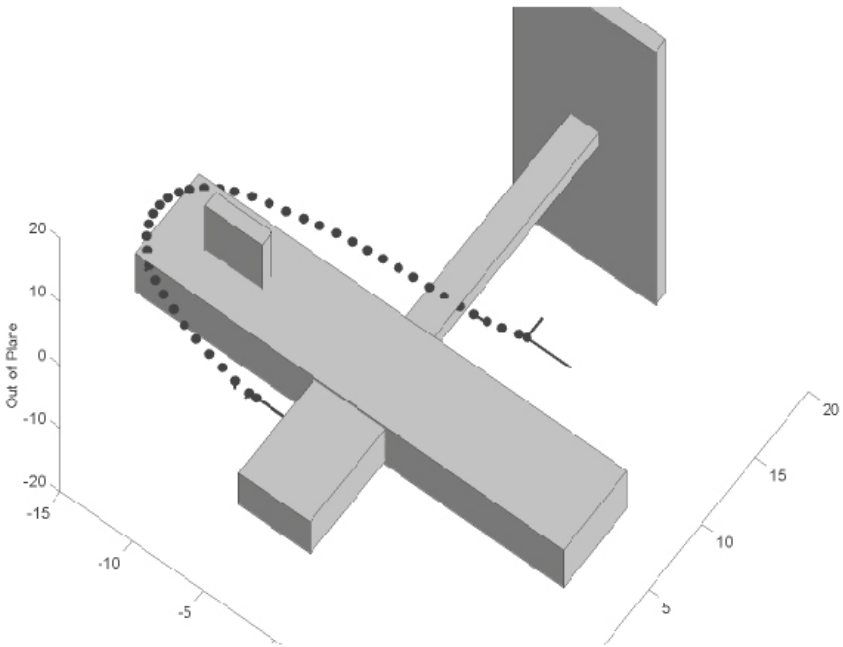


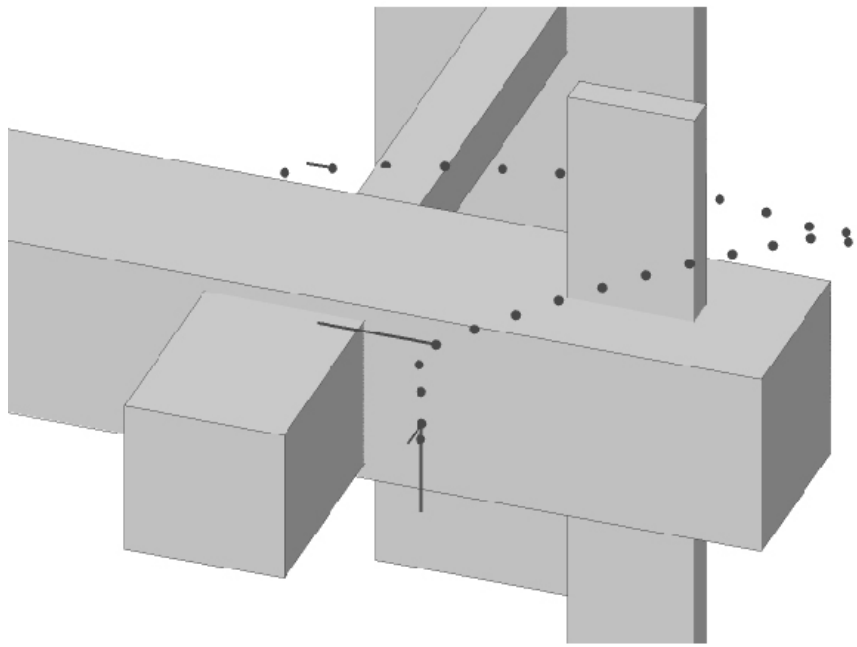


- Firing expected only at start and end: “bang-off-bang”
- Remove constraints on PI around center of maneuver
- Check results for PI
- Iterate, adding constraints back in, until no PI

Prior Knowledge II : Grouping

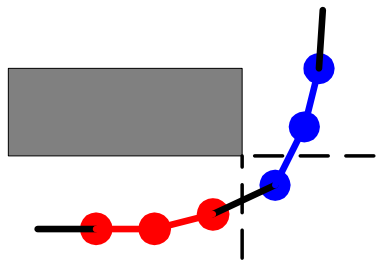
- Minimum fuel Space Station fly-by with PI constraints





Prior Knowledge II : Grouping

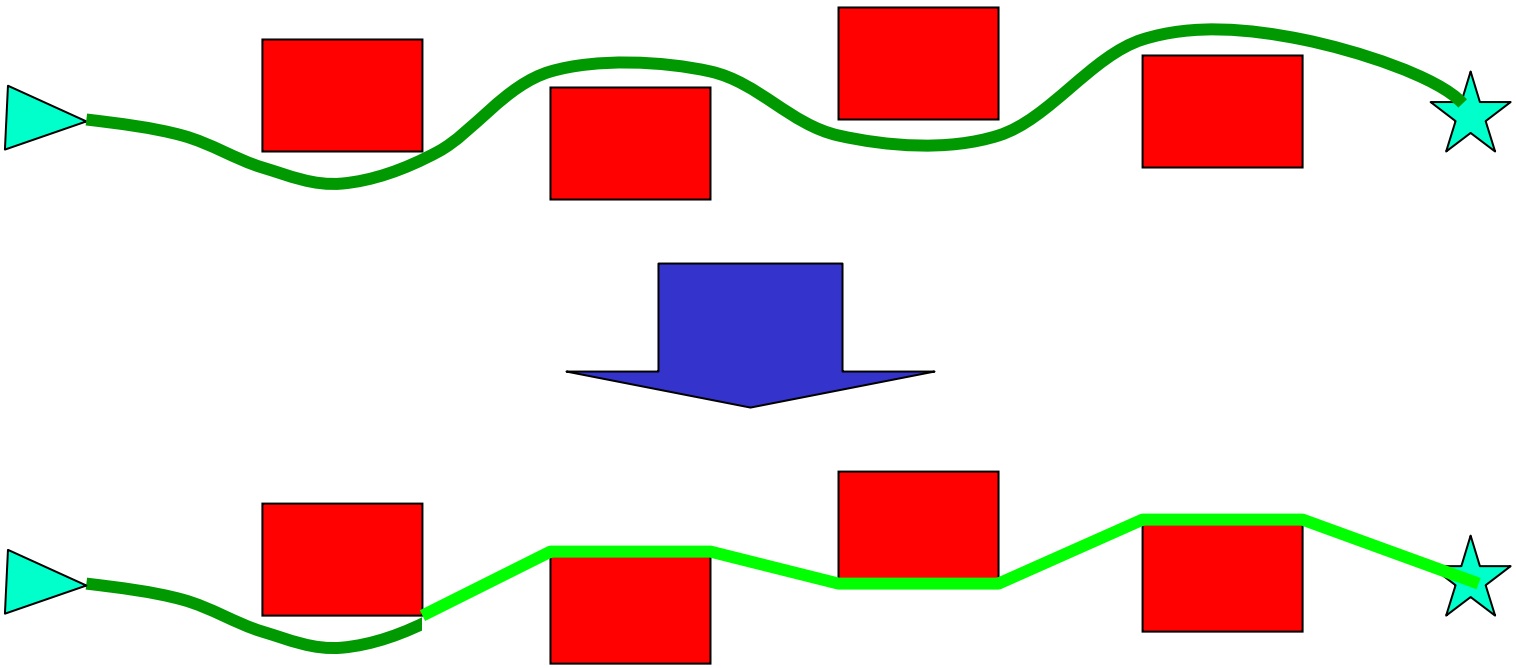
- Minimum fuel Space Station fly-by with PI constraints
  - 9600 binary variables
  - Impractical to find optimal solution
- Strategy:
  - Group time-steps



	Time (secs)
No plume constraints	8
All plume constraints	1800 (limit)
Groups of three	587

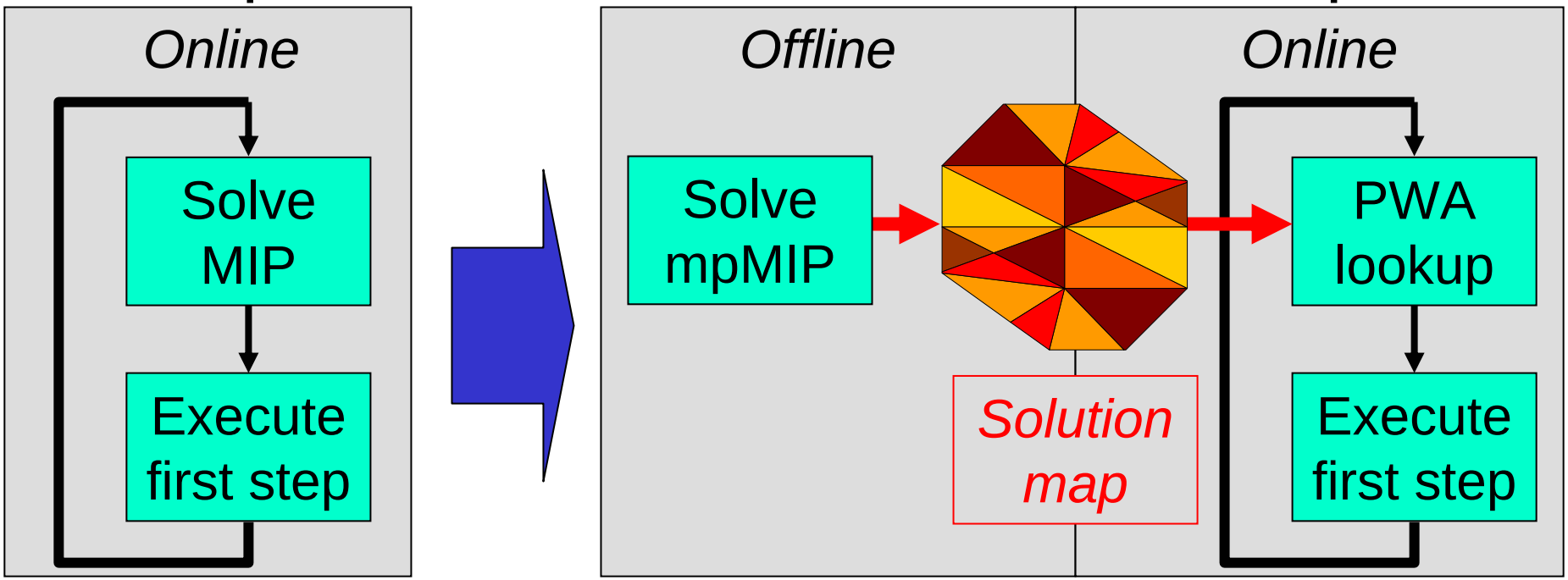
Approximate Cost-to-Go

- Replace tail of long plan with approximate plan
  - Re-plan online as goal approached



Multi-Parametric MIP

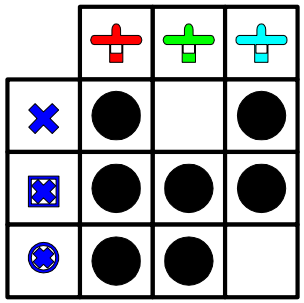
- Calculate solution ***offline*** as a piecewise affine function of the initial condition
  - [Bemporad et al 2004]
- Replace online MIP with PWA look-up

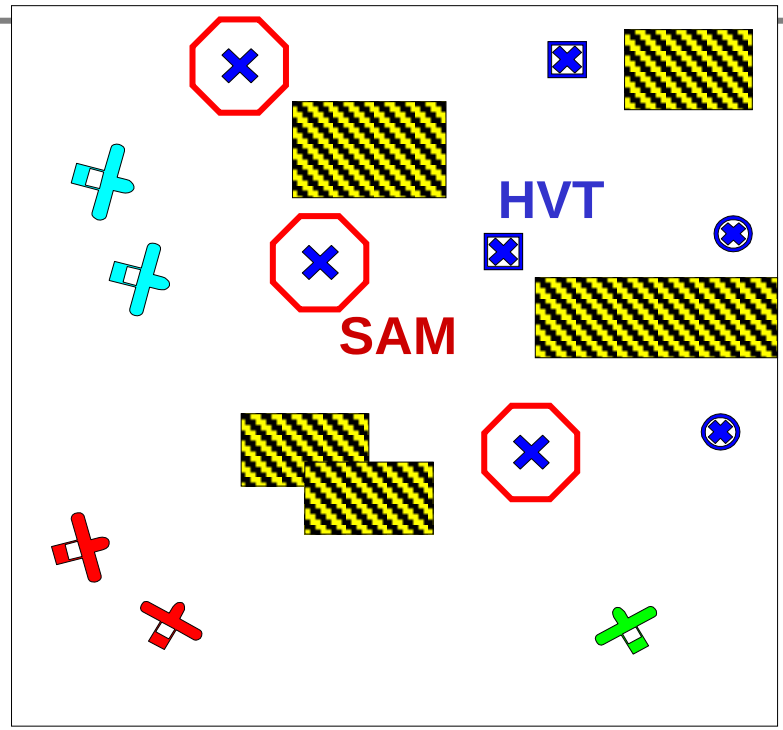


Task Assignment Example

Example: Task Assignment

- Different types of UAV's
- Different types of targets





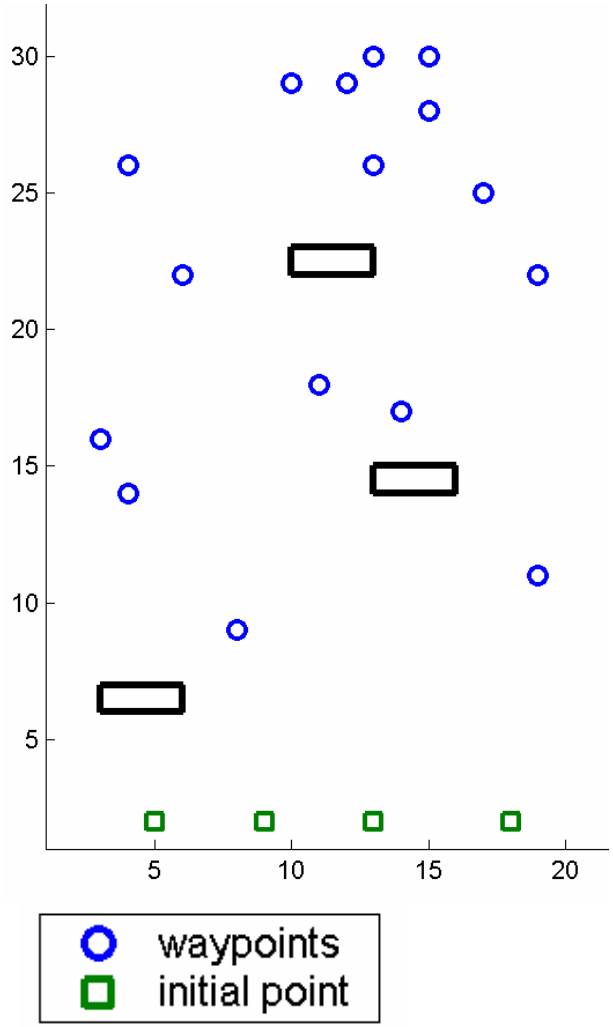
- Example timing constraints
  - BDA after Strike after Recon.
  - SAM site visited before HVT
  - Synchronized strike of HVT

  → All can be expressed in a **canonical form**

- Select assignments & trajectories to optimize desired mission objectives and satisfy dynamic/timing constraints
  - Costs for demonstration based on time, but extends to scores / risks
  - Hard because problems are tightly coupled

Experimental Scenario

- 4 vehicles
- 16 tasks
- Full horizon assignment approach
  not scalable for real-time assignment
- Environment with dynamic changes
  - Original assignment
  - Vehicle death
  - Pop-up target
  - Vehicle death
  - Target location update




RHTA on 4 Rovers
10/09/2003

UAV Example

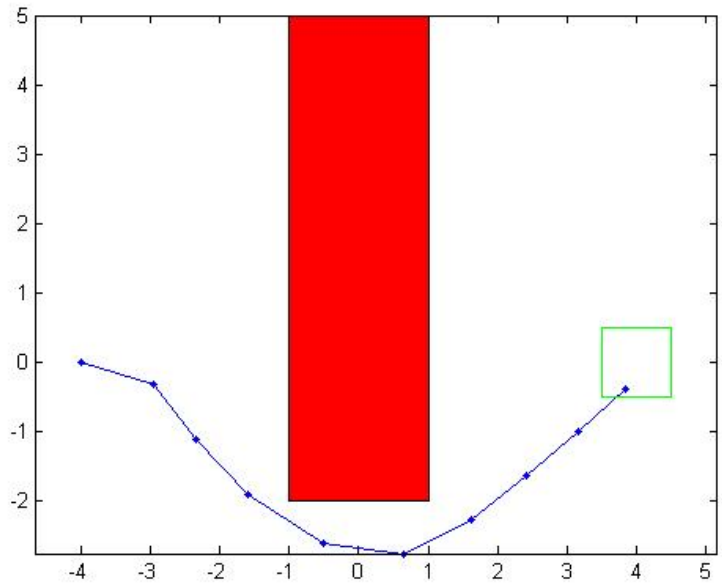
UAV Example

- Very simple trial example
  - One UAV avoiding one obstacle
  - Uses MPC and MIP

- Problem

$$\min_{\mathbf{f},\mathbf{v},\mathbf{r}} T$$

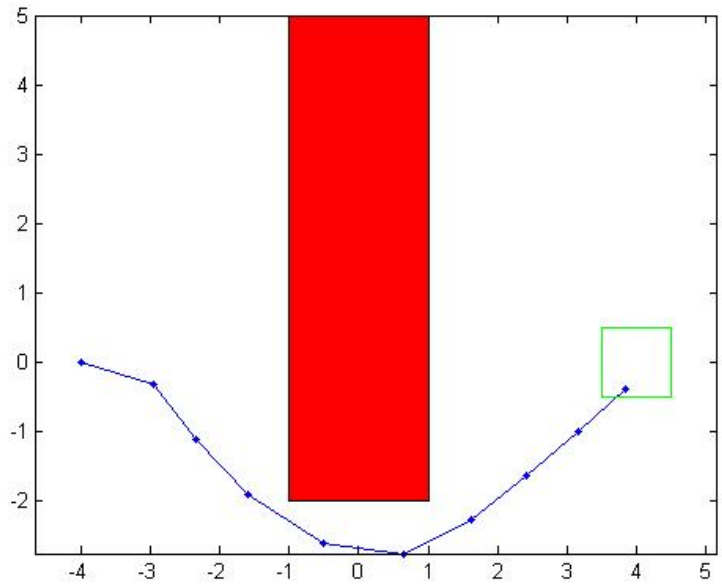
$$\begin{aligned}
\dot{\mathbf{r}} &= \mathbf{v} \\
\dot{\mathbf{v}} &= \mathbf{f} \\
\|\mathbf{v}\|_2 &\leq v_{\max} \\
\|\mathbf{f}\|_2 &\leq f_{\max} \\
\mathbf{x}(T) &\in \mathcal{T} \\
\mathbf{x} &\notin \mathcal{R}
\end{aligned}$$



- Download: http://seis.bris.ac.uk/~aeagr

UAV Example

- Very simple trial example
  - One UAV avoiding one obstacle
  - Uses MPC and MIP



- Implementation
  - Matlab sim script
  - AMPL data file
  - AMPL state file
  - AMPL script
  - AMPL model file

- Download: http://seis.bris.ac.uk/~aeagr

Session Outline

- Projected Variable Metric Algorithm for Mixed Integer Optimization Problem
  - Ali Ahmadzadeh (Univ. of Pennsylvania),
  - Bijan Sayyarrodsari (Pavilion Tech. Inc),
  - Abdollah Homaifar (North Carolina A\&T State Univ.)
- MILP Assignment Problems for Multi-Vehicle Systems
  - Matthew Earl (BAE)
  - Raffaello D'Andrea (Cornell Univ.)
- Real-Time MILP Path-Planning for Tactical UAV Applications
  - Cedric Ma (Northrop Grumman Corp.)
  - Robert Miller (Northrop Grumman Corp.)
- Receding Horizon Implementation of MILP for Vehicle Guidance
  - Yoshiaki Kuwata (MIT)
  - Jonathan P. How (MIT)

Questions?

- Web resources

  acl.mit.edu/MILP

  seis.bris.ac.uk/~aeagr

- Email

  arthur.richards@bristol.ac.uk

  jhow@mit.edu