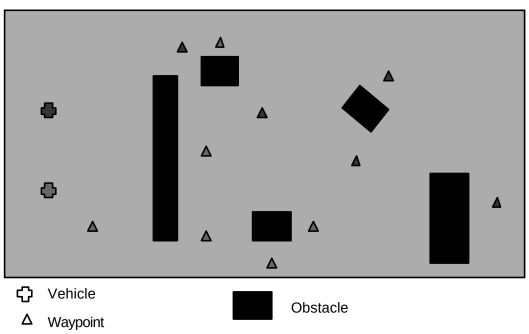
Integer Programming and Branch and Bound

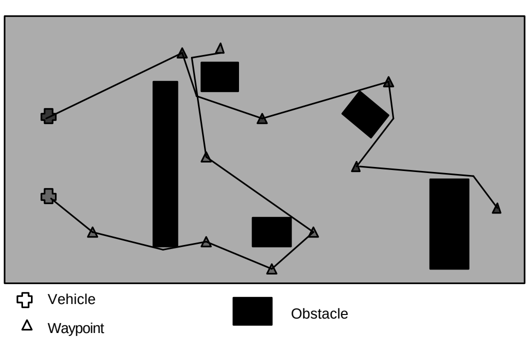
Brian C. Williams
16.410-13
Session 26

Adapted from slides
by Eric Feron, 16.410, 2002.

Cooperative Vehicle Path Planning



Cooperative Vehicle Path Planning



Cooperative Vehicle Path Planning

Objective: Find most fuel-efficient 2-D paths for all vehicles.

Constraints:

- Operate within vehicle dynamics
- Avoid static and moving obstacles
- Avoid other vehicles
- Visit waypoints in specified order
- Satisfy timing constraints

Cooperative Path Planning MILP Encoding: Constraints

- Min J_T — Receding Horizon Fuel Cost Fn
- $s_{ij} = w_{ij}$, etc. — State Space Constraints
- $\mathbf{s}_i+1 = \mathbf{As}_i + \mathbf{Bu}_i$ — State Evolution Equation
- Obstacle Avoidance:
  - $x_i = x_{min} + My_{i1}$
  - $-x_i = -x_{max} + My_{i2}$
  - $y_i = y_{min} + My_{i3}$
  - $-y_i = -y_{max} + My_{i4}$ — At least one enabled
  - $\Sigma y_{ik} = 3$ — At least one enabled
- Similar constraints for Collision Avoidance (for all pairs of vehicles)

Integer Programs

LP: Maximize $3x_1 + 4x_2$

Subject to:

$x_1 = 4$

$2x_2 = 12$

$3x_1 + 2x_2 = 18$

$x_{1}, x_2 = 0$

IP: Maximize $3x_1 + 4x_2$

Subject to:

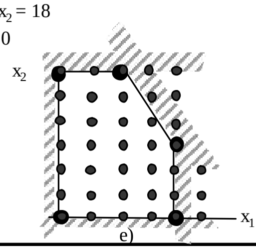
$x_1 = 4$

$2x_2 = 12$

$3x_1 + 2x_2 = 18$

$x_{1}, x_2 = 0$

x_1, x_2 integers



Integer Programming

Integer programs are LPs where some variables are integers

<u>Why Integer programs?</u>

1. Some variables are not real-valued:
   - Boeing only sells complete planes, not fractions.

2. Fractional LP solutions poorly approximate integer solutions:
   - For Boeing Aircraft Co., producing 4 versus 4.5 airplanes results in radically different profits.

<u>Often a mix is desired</u> of integer and non-integer variables

- Mixed Integer Linear Programs (MILP).

Outline

- Review of Integer Programming (IP)
- How do we solve using Branch and Bound?
  - Solving Binary IPs

Appendices:

- How do we encode decisions using IP?
  - Exclusion between choices
  - Exclusion between constraints
- Solving Mixed IPs and LPs

Solving Integer Programs: Characteristics

- Fewer feasible solutions than LPs.
- Worst-case exponential in # of variables.
- Solution time tends to:
  - Increase with increased # of variables.
  - Decrease with increased # of constraints.
- Commercial software:
  - Cplex

Methods To Solve Integer Programs

- Branch and Bound
  - Binary Integer Programs
  - Integer Programs
  - Mixed Integer (Real) Programs
- Cutting Planes

Branch and Bound

Problem: Optimize f(x) subject to A(x) = 0, $x \in D$

B & B - an instance of Divide & Conquer:

I. Bound D's solution and compare to best alternative.
   1) Bound solution to D quickly.
      - Perform quick check by relaxing hard part of problem and solve.
      - ➔ Relax integer constraints. Relaxation is LP.
   2) Use bound to " fathom" (finish) D if possible.
      - **a.** **If** relaxed solution is integer, **Then** keep soln if best found to date ("incumbent"), delete D_i
      - **b.** **If** relaxed solution is worse than incumbent, **Then** delete D_i.
      - **c.** **If** no feasible solution, **Then** delete D_i.

II. Otherwise Branch to smaller subproblems
   1) Partition D into subproblems $D_1 \ldots D_n$
   2) Apply B&B to all subproblems, typically Depth First.

B&B for Binary Integer Programs (BIPs)

Problem i: Optimize f(x) st A(x) = 0, $x_k \in \{0,1\}$, $x \in D_i$

Domain D_i encoding (for subproblem):

- partial assignment to x,
  - $\{x_1 = 1, x_2 = 0, \ldots\}$

Branch Step:

1. Find variable x_j that is unassigned in D_i
2. Create two subproblems by splitting D_i:
   - $D_{i1} \equiv D_i \cup \{x_j = 1\}$
   - $D_{i0} \equiv D_i \cup \{x_j = 0\}$
3. Place on search Queue

Example: B&B for BIPs

{}

Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-x_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1$, $x_i = 0$, x_i **integer**

- Initialize

Queue: {}

Incumbent: none

Best cost Z*: - inf

Example: B&B for BIPs

{}

Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-x_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1$, $x_i = 0$, x_i **integer**

- Dequeue {}

Queue: ~~{}~~

Incumbent: none

Best cost Z*: - inf

Example: B&B for BIPs

{}

Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-x_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1$, $x_i = 0$, ~~x_i **integer**~~

Z = 16.5, x = <0.8333,1,0,1>

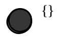
- Bound {}
  1. Constrain x_i by {}
  2. Relax to LP
  3. Solve LP

Queue:

Incumbent: none

Best cost Z*: - inf

Example: B&B for BIPs



Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-x_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1$, $x_i = 0$, ~~x_i **integer**~~

Z = 16.5, x = <0.8333,1,0,1>

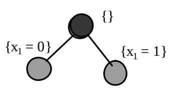
- Try to fathom:
  1. infeasible?
  2. worse than incumbent?
  3. integer solution?

Queue:

Incumbent: none

Best cost Z*: - inf

Example: B&B for BIPs



Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-x_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1$, $x_i = 0$, ~~x_i **integer**~~

Z = 16.5, x = <0.8333,1,0,1>

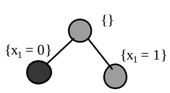
- Branch:
  1. select unassigned x_i
     - pick non-integer (x_1)
  2. Split on x_i

Queue: {x_1 = 0}{x_1 = 1}

Incumbent: none

Best cost Z*: - inf

Example: B&B for BIPs



Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-x_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1$, $x_i = 0$, x_i **integer**

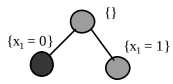
- Dequeue:
  - <u>depth first</u> or
  - best first

Queue: ~~{x_1 = 0}~~{x_1 = 1}

Incumbent: none

Best cost Z*: - inf

Example: B&B for BIPs



Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-x_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1, x_i = 0, x_i$ **integer**

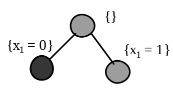
Queue: $\{x_1 = 1\}$

Incumbent: none

Best cost Z*: - inf

- Bound $\{x_1 = 0\}$
  - constrain x by $\{x_1 = 0\}$
  - relax to LP
  - solve

Example: B&B for BIPs



Solve:

Max $Z = 9\,\mathbf{0} + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6\,\mathbf{0} + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $\mathbf{-0} + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1, x_i = 0, x_i$ **integer**

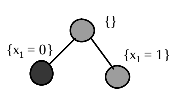
Queue: $\{x_1 = 1\}$

Incumbent: none

Best cost Z*: - inf

- Bound $\{x_1 = 0\}$
  - constrain x by $\{x_1 = 0\}$
  - relax to LP
  - solve

Example: B&B for BIPs



Solve:

Max $Z = 9\,0 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6\,0 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-0 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1, x_i = 0,$ ~~x_i **integer**~~

Z = 19,5, x = <0,1,0,1>

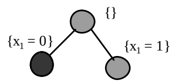
Queue: $\{x_1 = 1\}$

Incumbent: none

Best cost Z*: - inf

- Bound $\{x_1 = 0\}$
  - constrain x by $\{x_1 = 0\}$
  - relax to LP
  - solve LP

Example: B&B for BIPs



Solve:

Max Z = 9 0 + $5x_2$ + $6x_3$ + $4x_4$

Subject to:

- 6 0 + $3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-0 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1$, $x_j = 0$, ~~x_j **integer**~~

Z = 19,5, x = <0,1,0,1>

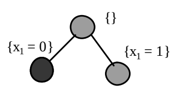
- Try to fathom:
  1. infeasible?
  2. worse than incumbent?
  3. <u>integer solution?</u>

Queue: {x_1 = 1}

Incumbent: none

Best cost Z*: - inf

Example: B&B for BIPs



Solve:

Max Z = 9 0 + $5x_2$ + $6x_3$ + $4x_4$

Subject to:

- 6 0 + $3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-0 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1$, $x_j = 0$, ~~x_j **integer**~~

Z = 19,5, x = <0,1,0,1>

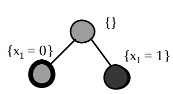
- Try to fathom:
  1. infeasible?
  2. worse than incumbent?
  3. <u>integer solution?</u>

Queue: {x_1 = 1}

Incumbent: **x = <0,1,0,1>**

Best cost Z*: 9

Example: B&B for BIPs



Solve:

Max Z = $9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-x_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1$, $x_j = 0$, x_j **integer**

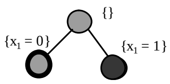
- Dequeue

Queue: ~~{x_1 = 1}~~

Incumbent: **x = <0,1,0,1>**

Best cost Z*: 9

Example: B&B for BIPs



Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-x_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1$, $x_i = 0$, x_i **integer**

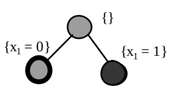
• Bound $\{x_1 = 1\}$

Queue:

Incumbent: **x = <0,1,0,1>**

Best cost Z*: 9

Example: B&B for BIPs



Solve:

Max $Z = \mathbf{9}x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $\mathbf{6}x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $\mathbf{-1}_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1$, $x_i = 0$, ~~x_i **integer**~~

Z = 16.~~2~~, x = <1,.8,0,.8>

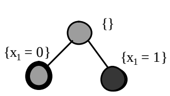
• Bound $\{x_1 = 1\}$

Queue:

Incumbent: **x = <0,1,0,1>**

Best cost Z*: 9

Example: B&B for BIPs



Solve:

Max $Z = \mathbf{9}x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $\mathbf{6}x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $\mathbf{-1}_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1$, $x_i = 0$, ~~x_i **integer**~~

Z = 16.~~2~~, x = <1,.8,0,.8>

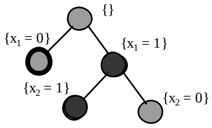
- Try to fathom:
  - infeasible?
  - worse than incumbent?
  - integer solution?

Queue: $\{x_1 = 1\}$

Incumbent: **x = <0,1,0,1>**

Best cost Z*: 9

Example: B&B for BIPs



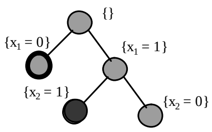
Solve:
Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$
Subject to:
- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-1_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1, x_i = 0, x_i$ **integer**

Z = 16.2, x = <1,.8,0,.8>

Queue: {x_1=1, x_2=1}{x_1=1, x_2=0}
Incumbent: **x = <0,1,0,1>**
Best cost Z*: 9

- Branch
- Dequeue

Example: B&B for BIPs

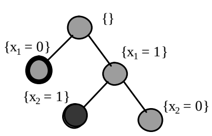


Solve:
Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$
Subject to:
- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-x_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1, x_i = 0, x_i$ **integer**

Queue: {x_1=1, x_2=0}
Incumbent: **x = <0,1,0,1>**
Best cost Z*: 9

- Bound {x_1 = 1, x_2 = 1}

Example: B&B for BIPs



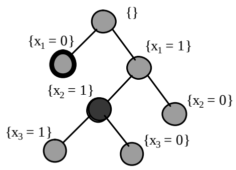
Solve:
Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$
Subject to:
- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-1_1 + x_3 = 0$
- $-1_2 + x_4 = 0$
- $x_i = 1, x_i = 0,$ ~~x_i **integer**~~

Z = 16, x = <1,1,0,.5>

Queue: {x_1=1, x_2=0}
Incumbent: **x = <0,1,0,1>**
Best cost Z*: 9

- Try to fathom:
  - infeasible?
  - worse than incumbent?
  - integer solution?

Example: B&B for BIPs



Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-x_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1$, $x_i = 0$, ~~x_i **integer**~~

Z = 16, x = <1,1,0,.5>

- Branch

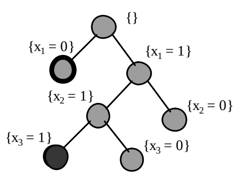
Queue: {…,x_3=1}{…,x_3=0}{…,x_2=0}

Incumbent: **x = <0,1,0,1>**

Best cost Z*: 9

Correction: Should branch on x_4

Example: B&B for BIPs



Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-x_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1$, $x_i = 0$, x_i **integer**

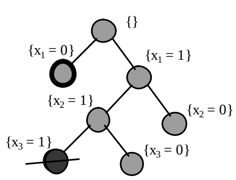
Queue: ~~{…,x_3=1}~~ {…,x_3=0}{…,x_2=0}

Incumbent: **x = <0,1,0,1>**

Best cost Z*: 9

- Dequeue
- Bound {x_1=1, x_2=1, x_3=1}

Example: B&B for BIPs



Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $1_3 + x_4 = 1$
- $-1_1 + 1_3 = 0$
- $-1_2 + x_4 = 0$
- $x_i = 1$, $x_i = 0$, ~~x_i **integer**~~

No Solution

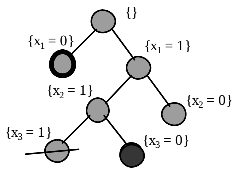
- Try to fathom:
  - infeasible?

Queue: {…,x_3 = 0}{…,x_2 = 0}

Incumbent: **x = <0,1,0,1>**

Best cost Z*: 9

Example: B&B for BIPs



Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-x_1 + x_3 = 0$
- $-x_2 + x_4 = 0$
- $x_i = 1, x_i = 0$, x_i **integer**

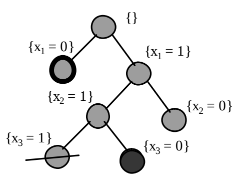
Queue: ~~{…,x_3 = 0}~~{…,x_2 = 0}

Incumbent: **x = <0,1,0,1>**

Best cost Z*: 9

- Dequeue
- Bound {x_1=1, x_2=1, x_3=0}

Example: B&B for BIPs



Solve:

Max Z = **9**x_1 + **5**x_2 + $6x_3 + 4x_4$

Subject to:

- **6**x_1 + **3**x_2 + $5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- **-1_1** + $x_3 = 0$
- **-1_2** + $x_4 = 0$
- $x_i = 1, x_i = 0$, ~~x_i **integer**~~

Z = 16, x = <1,1,0,.5>

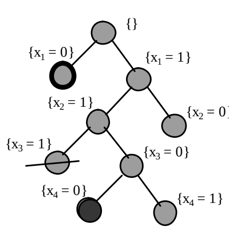
Queue: {…,x_2 = 0}

Incumbent: **x = <0,1,0,1>**

Best cost Z*: 9

- Try to fathom:
  - infeasible?
  - worse than incumbent?
  - integer solution?

Example: B&B for BIPs



Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-1_1 + x_3 = 0$
- $-1_2 + x_4 = 0$
- $x_i = 1, x_i = 0$, ~~x_i **integer**~~

Z = 14, x = <1,1,0,0>

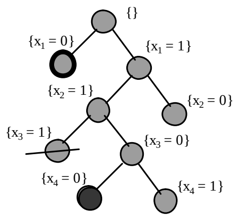
Queue: ~~{…,x_4**=0}**~~{…,x_4=1}{…,x_2=0}

Incumbent: **x = <0,1,0,1>**

Best cost Z*: 9

- Branch
- Dequeue
- Bound

Example: B&B for BIPs



Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-1_1 + x_3 = 0$
- $-1_2 + x_4 = 0$
- $x_i = 1, x_i = 0,$ ~~x_i **integer**~~

Z = 14, x = <1,1,0,0>

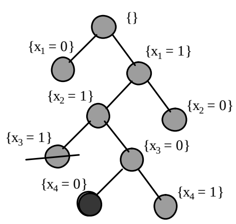
- Try to fathom:
  - infeasible?
  - worse than incumbent?
  - integer solution?

Queue: $\{\ldots, x_4=1\}\{\ldots, x_2=0\}$

Incumbent: **x = <0,1,0,1>**

Best cost Z*: 9

Example: B&B for BIPs



Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-1_1 + x_3 = 0$
- $-1_2 + x_4 = 0$
- $x_i = 1, x_i = 0,$ ~~x_i **integer**~~

Z = 14, x = <1,1,0,0>

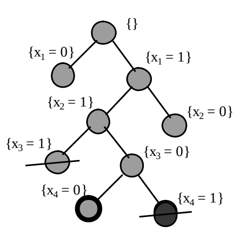
- Try to fathom:
  - infeasible?
  - worse than incumbent?
  - integer solution?

Queue: $\{\ldots, x_4=1\}\{\ldots, x_2=0\}$

Incumbent: **x = <1,1,0,0>**

Best cost Z*: 14

Example: B&B for BIPs



Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + 1_4 = 1$
- $-1_1 + x_3 = 0$
- $-1_2 + 1_4 = 0$
- $x_i = 1, x_i = 0,$ ~~x_i **integer**~~

No Solution, x = <1,1,0,1>

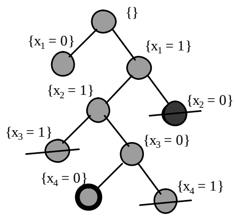
- Try to fathom:
  - infeasible?
  - worse than incumbent?
  - integer solution?

Queue: $\{\ldots, x_4=1\}\{\ldots, x_2=0\}$

Incumbent: **x = <1,1,0,0>**

Best cost Z*: 14

Example: B&B for BIPs



Solve:

Max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to:

- $6x_1 + 3x_2 + 5x_3 + 2x_4 = 10$
- $x_3 + x_4 = 1$
- $-1_1 + x_3 = 0$
- $-1_2 + x_4 = 0$
- $x_i = 1$, $x_i = 0$, ~~x_i **integer**~~

Z = 13.8, x = <1,0,.8,0>

Queue: {…, x_2=0}

Incumbent: **x = <1,1,0,0>**

Best cost Z*: 14

- Try to fathom:
  - infeasible?
  - worse than incumbent?
  - integer solution?

Outline

- What is Integer Programming (IP)?
- How do we solve using Branch and Bound?
  - Characteristics
  - Solving Binary IPs

Appendices:

- How do we encode decisions using IP?
  - Exclusion between choices
  - Exclusion between constraints
- Solving Mixed IPs and LPs

Integer Programming for Decision Making

Encode "Yes or no" decisions with binary variables:

x_j — 1 if decision is yes; 0 if decision is no.

<u>Binary Integer Programming (BIP):</u>

- Binary variables + linear constraints.

- How is this different from propositional logic?

Binary Integer Programming Example: Cal Aircraft Manufacturing Company

Problem:

1. Cal wants to expand:
   - Build new factory in either Los Angeles, San Francisco, both or neither.
   - Build new warehouse (at most one).
   - Warehouse must be built close to the city of a new factory.
2. Available capital: $10,000,000
3. Cal wants to maximize "total net present value" (profitability vs. time value of money)

		NPV	Price
1	Build a factory in L.A.?	$9m	$6m
2	Build a factory in S.F.?	$5m	$3m
3	Build a warehouse in L.A.?	$6m	$5m
4	Build a warehouse in S.F.?	$4m	$2m

Binary Integer Programming Example: Cal Aircraft Manufacturing Company

Cal wants to expand:
Build new factory in Los Angeles, San Francisco, both or neither.
Build new warehouse (at most one).
Warehouse must be built close to the city of a new factory .

What decisions are to be made?

1.Build factory in LA
2.Build factory in SFO
3.Build warehouse in LA
4.Build warehouse in SFO

Introduce 4 binary variables x_i = 1 if decision i is yes; 0 if decision i is no

Binary Integer Programming Example: Cal Aircraft Manufacturing Company

1. Cal wants to expand
2. Available capital: $10,000,000
3. Cal wants to maximize "total net present value" (profitability vs. time value of money)

		NPV	Price
1	Build a factory in L.A.?	$9m	$6m
2	Build a factory in S.F.?	$5m	$3m
3	Build a warehouse in L.A.?	$6m	$5m
4	Build a warehouse in S.F.?	$4m	$2m

What is the objective?
- Maximize NPV:

$$Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$$

What are the constraints on capital?
- Don't go beyond means:

$$6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10$$

Binary Integer Programming Example: Cal Aircraft Manufacturing Company

LA factory(x_1), SFO factory(x_2), LA warehouse(x_3),SFO warehouse (x_4)

- Build new factory in Los Angeles, San Francisco, both or neither.
- Build new warehouse (at most one).
- Warehouse <u>must</u> be built close to city of a new factory.

What are the constraints between decisions?

1. No more than one warehouse:
   Most 1 of $\{x_3, x_4\}$
   $x_3 + x_4 \leq 1$
2. Warehouse in LA only if Factory is in LA:
   x_3 implies x_1
   $x_3 - x_1 \leq 0$
3. Warehouse in SFO only if Factory is in SFO:
   x_4 implies x_2
   $x_4 - x_2 \leq 0$

Encoding Decision Constraints:

- <u>Exclusive choices</u>
  - Example: at most 2 decisions in a group can be yes:

  LP Encoding:
  $x_1 + \ldots + x_k \leq 2.$

- <u>Logical implications</u>
  - x_1 implies x_2: (x_1 requires x_2)

  LP Encoding:
  $x_1 - x_2 \leq 0.$

Binary Integer Programming Example: Cal Aircraft Manufacturing Company

LA factory(x1), SFO factory(x2), LA warehouse(x3),SFO warehouse (x4)

- Build new factory in Los Angeles, San Francisco, or both.
- Build new warehouse (only one).
- Warehouse <u>must</u> be built close to city of a new factory.

What are the constraints between decisions?

1. No more than one warehouse:
   Most 1 of $\{x_3, x_4\}$
   $x_3 + x_4 \leq 1$
2. Warehouse in LA only if Factory is in LA:
   x_3 implies x_1
   $x_3 - x_1 \leq 0$
3. Warehouse in SFO only if Factory is in SFO:
   x_4 implies x_2
   $x_4 - x_2 \leq 0$

Binary Integer Programming Example: Cal Aircraft Manufacturing Company

Complete binary integer program:

Maximize $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Subject to: $6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10$

$x_3 + x_4 \leq 1$
$x_3 - x_1 \leq 0$
$x_4 - x_2 \leq 0$
$x_j \leq 1$
$x_j \geq 0$
$x_j = \{0,1\}, j=1,2,3,4$

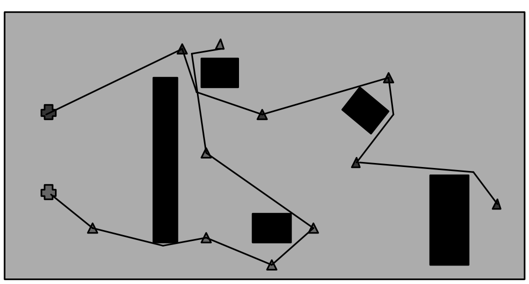
Outline

- What is Integer Programming (IP)?
- How do we solve using Branch and Bound?
  - Characteristics
  - Solving Binary IPs

Appendices:

- How do we encode decisions using IP?
  - Exclusion between choices
  - Exclusion between constraints
- Solving Mixed IPs and LPs

Cooperative Vehicle Path Planning



Cooperative Path Planning MILP Encoding: Constraints

- Min J_T — Receding Horizon Fuel Cost Fn
- $s_{ij} = w_{ij}$, etc. — State Space Constraints
- $\mathbf{s}_i+1 = \mathbf{A}\mathbf{s}_i + \mathbf{B}\mathbf{u}_i$ — State Evolution Equation
- Obstacle Avoidance:
  $x_i = x_{min} + My_{i1}$
  $-x_i = -x_{max} + My_{i2}$
  $y_i = y_{min} + My_{i3}$
  $-y_i = -y_{max} + My_{i4}$
  $\Sigma\, y_{ik} = 3$
- Similar constraints for (for all pairs of vehicles) — Collision Avoidance

Cooperative path planning MILP Encoding: Fuel Equation

total fuel calculated over all time instants i

past-horizon terminal cost term

$$\min_{w_i, v_i} = J_T = \min_{w_i, v_i} \sum_{i=1}^{N-1} q'w_i + \sum_{i=1}^{N-1} r'v_i + p'w_N$$

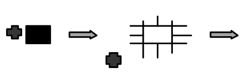
slack control vector

weighting vectors

slack state vector

How Do We Encode Obstacles?

- Each obstacle-vehicle pair represents a disjunctive constraint:



Red Vehicle is above obstacle OR
Red Vehicle is below obstacle OR
Red Vehicle is left of obstacle OR
Red Vehicle is right of obstacle

- Each disjunct is an inequality
  - let xR, yR be red vehicle's co-ordinates then:
  - Left: xR < 3
  - Above: R > 4, . . .
- Constraints are not limited to rectangular obstacles
  - (inequalities might include both co-ordinates)
- **May be any polygon**
  - (convex or concave)

Encoding Exclusion Constraints

Example (x1 ,x2 real)

Either $3x_1 + 2x_2 \leq 18$

Or x + 4x $\leq$ 16

BIP Encoding:

- **Use Big M to turn-off constraint:**

  Either:

  $3x_1 + 2x_2 \leq 18$

  and $x_1 + 4x_2 \leq 16 + M$ (and M is very BIG)

  Or:

  $3x_1 + 2x_2 \leq 18 + M$

  and $x_1 + 6x_2 \leq 16$

- **Use binary y to decide which constraint to turn off:**

  3x1 + 2x2 $\leq$ 18 + y M

  x1 + 2x2 $\leq$ 16 + (1-y)M

  $y \in \{0,1\}$

Cooperative Path Planning MILP Encoding: Constraints

- Min J_T — Receding Horizon Fuel Cost Fn
- $s_{ij} = w_{ij}$, etc. — State Space Constraints
- $\mathbf{s}_i+1 = \mathbf{A}\mathbf{s}_i + \mathbf{B}\mathbf{u}_i$ — State Evolution Equation
- $x_i = x_{min} + My_{i1}$

  $-x_i = -x_{max} + My_{i2}$

  $y_i = y_{min} + My_{i3}$ — Obstacle Avoidance

  $-y_i = -y_{max} + My_{i4}$ — At least one enabled

  $\Sigma\, y_{ik} = 3$ — At least one enabled
- Similar constraints for Collision Avoidance (for all pairs of vehicles)

Encoding General Exclusion Constraints

- <u>K out of N constraints hold:</u>

  $f_1(x_1, x_2, \ldots x_n) \leq d_1$ OR

  :

  $f_N(x_1, x_2, \ldots, x_n) \leq d_N$

  where f_i are linear expressions

- <u>At least K of N hold:</u>

• LP Encoding:

- Introduce y_i to turn off each constraint i:
- Use Big M to turn-off constraint:

  f1(x1, ... , xn) $\leq$ d1 + My1

  :

  fN(x1, ... , xn) $\leq$ dN + MyN

- Constrain K of the y_i to select constraints:

$$\sum_{i=1}^{N} y_i = N - K \qquad\qquad \sum_{i=1}^{N} y_i \leq N - K$$

Encoding Mappings to Finite Domains

- Function takes on one out of n possible values:

  $a_1x_1 + \ldots a_n x_n = [d_1 \text{ or } d_2 \ldots \text{ or } d_p]$

  - LP Encoding:

    $y_i \in \{0,1\} \quad i=1,2,\ldots p$

    $\Sigma\, y_i = 1$

    $a_1x_1 + \ldots a_n x_n = \Sigma_t\, d_i\, y_i$

Encoding Constraints

- Fixed – charge problem:

  $f_i(x_j) = \quad | \; k_j + c_j x_j \;$ if $x_j > 0$

  $\qquad\qquad | \; 0$ if $x_j = 0$

Minimizing costs:

Minimizing $z = f_1(x_1) + \text{---} + f_n(x_n)$

Yes or no decisions: should each of the activities be undertaken?

Introduce auxiliary variables:

$y_1, \ldots, y_n = 0,1$

$y = 1$ if $x > 0$

0 if $x = 0$

$$Z = \sum_{i=1}^{n} c_i x_i + k_i y_i$$

Which can be written as a linear constraint using big M:

$x \leq yM$

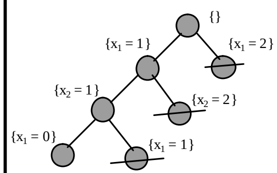
Outline

- What is Integer Programming (IP)?
- How do we solve using Branch and Bound?
  - Characteristics
  - Solving Binary IPs

Appendices:

- How do we encode decisions using IP?
  - Exclusion between choices
  - Exclusion between constraints
- Solving Mixed IPs and LPs

Example: B&B for MIPs



Max $Z = 4x_1 - 2x_2 + 7x_3 - x_4$

Subject to:

- $x_1 + 5x_3 = 10$
- $x_1 + x_2 - x_3 = 1$
- $6x_1 + 5x_2 = 0$
- $-x_1 + 2x_3 - 2x_4 = 3$
- $x_i = 0$, x_i **integer** x_1, x_2, x_3,

Z = 14.25, x = <1.25,1.5,1.75,0>

Z = 14.2, x = <1,1.2,1.8,0>

Z = 14 1/6, x = <5/6,1,11/6,0>

Z = 13.5, x = <0,0,2,.5>

Infeasible, x = <1,=1,?,?>

Z = 12 1/6, x = <5/6,2,11/6,0>

Infeasible, x = < =2,?,?,?>

Incumbent	**x = <0,0,2,.5>**
Best cost Z*	13.5