MIP is a Flexible Tool

- Logistics: Traveling Salesman problem, Vehicle Routing.
- Inventory (and Production) Planning.
- Facilities Location.
- Capacity planning: Matching, Assignment Problem.
- Data Mining: Classification, Regression.
- Airline Industry: Schedule Planning, Fleet Assignment, Aircraft Rotation, Crew-pairing.
- Mining and Forestry Industry: Covering Models, partitioning models.
- National Security Planning.
- VLSI Chip Design.
- Computational Biology: Sequence Alignment, Genome Rearrangement.
- Health care: IMRT, Scheduling.
- Sports Scheduling, Timetabling.

Some "Slightly Dated" Applications of Integer Programming

Papers from Interfaces.

Company	Year	Type of Model	Revenue
Air New Zealand	2001	Crew Scheduling	NZ $15.6 million
AT&T	2000	Network Restoration	'Hundreds of millions of dollars'
NBC	2002	Product Mix/ Commercials/Schedule	$200 million
Procter & Gamble	2006	Expressive Bidding	$298.4 million
Schindler Elevator	2003	Routing planning	$1 million
UPS	2004	Network Design	$87 million
Ford	2001	Set-covering/ Product optimization	$250 million
Hewlett-Packard	2004	Inventory Optimization	$130 million
Merrill-Lynch	2002	Integrated Choice Strategy	$80 million
Motorola	2005	Bidding/ Supplier Negotiation	$200 million

A Natural Way to Solve MIPs

1. Enumerate all possible integer solution vectors.
2. For each potential solution vectors, check if it is feasible.
3. If feasible, then compute objective function value.
4. Pick the best solution.

Some Calculations

Suppose we can evaluate 10^6 potential solution vectors in a second.

Number of binary variables	Number of vectors 2^n	Time
10	1024	0.001 seconds
20	1048576	1 second
30	$\sim 10^9$	16 minutes
40	$\sim 10^{12}$	11 days
50	$\sim 10^{15}$	31 years
60	$\sim 10^{18}$	31,709 years
70	$\sim 10^{21}$	31,709,791 years

This has got “nothing" to do with speed of computers. Every time the speed is doubled $\Rightarrow$ It allows me to solve a problem with **1** extra variable.

However, we routinely solve problems with thousands of integer variables.