ECE532 Project: Linear Programming Solver

Adham Ragab, Martin Staadecker, Ahmed Hamoda, Abnash Bassi

Group 1

Agenda

Topics Covered

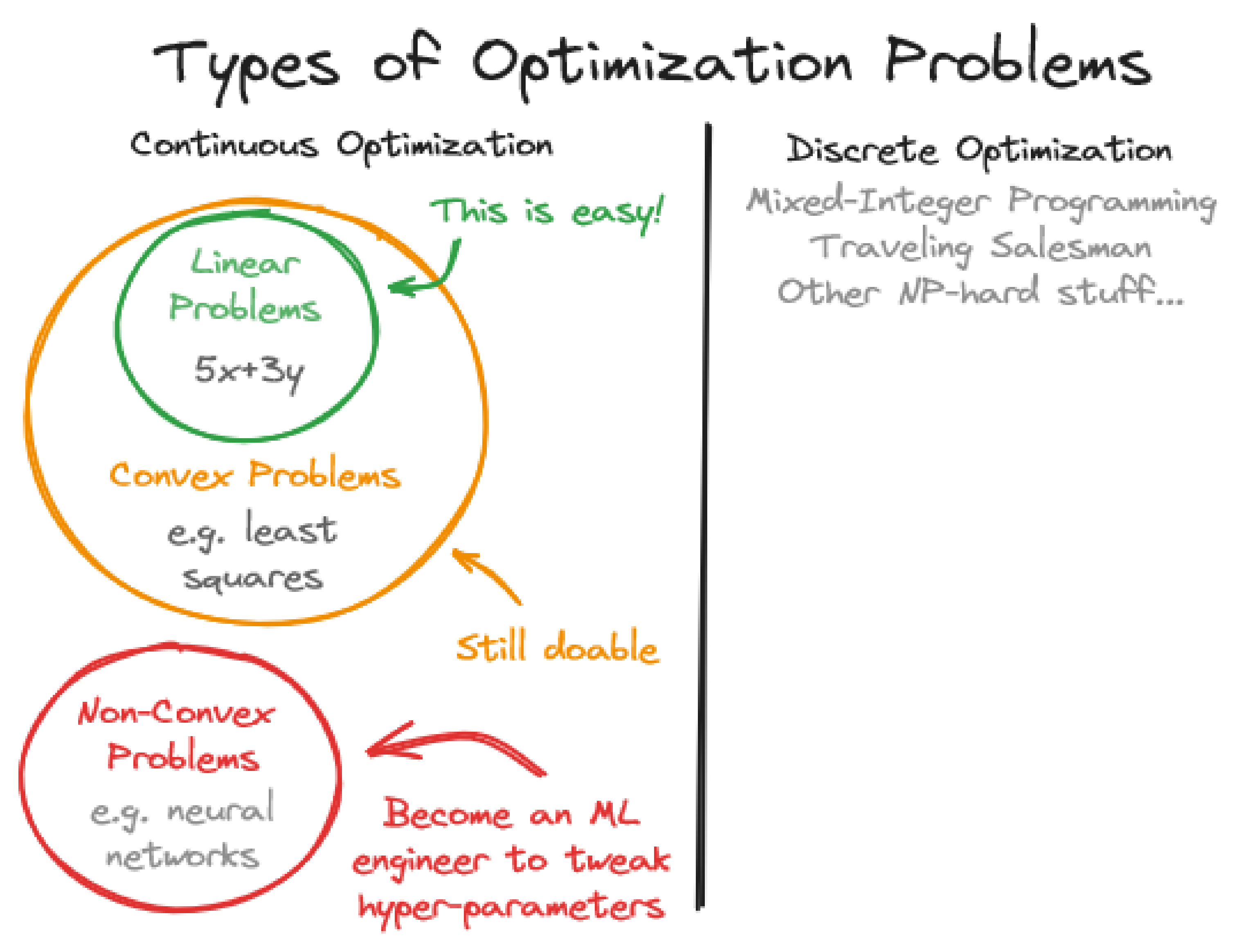
- What even is this about?
- What we are doing and How
- Risks and Uncertainties
- What we are not doing

Start

What even is this about?

Back to Agenda Page

What is Linear Programming?



Minimize (or maximize) the objective function:

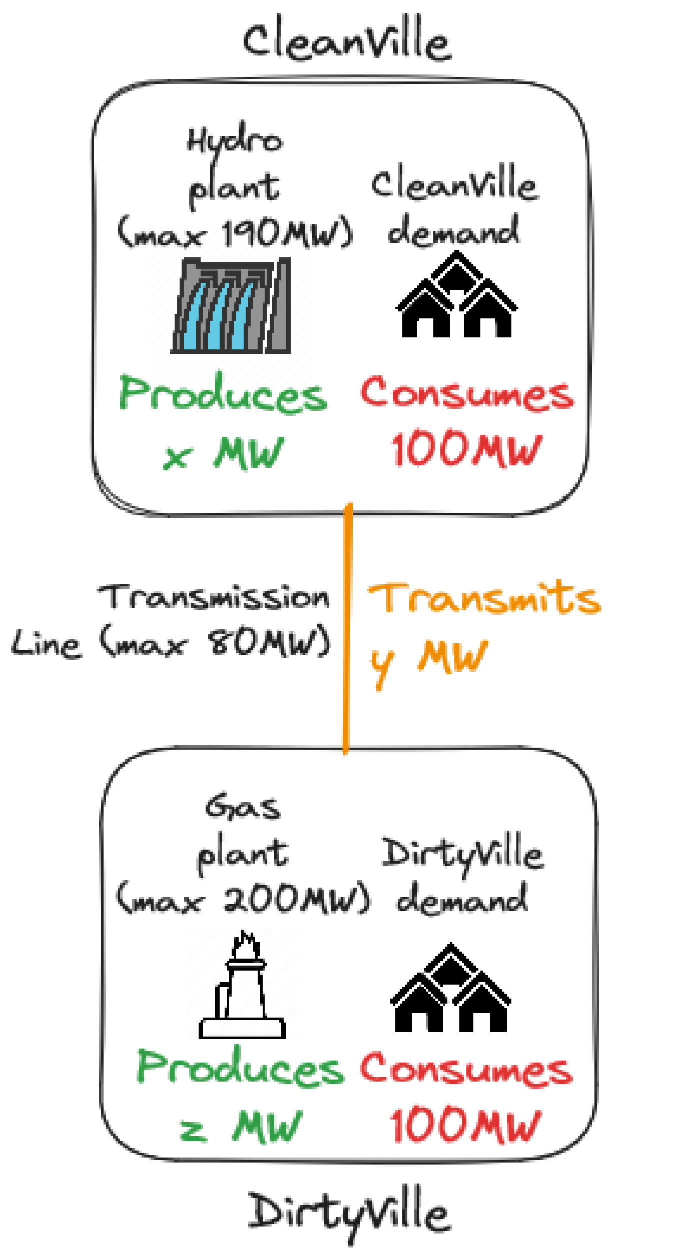
$$c_1x_1 + c_2x_2 + c_3x_3 + \ldots + c_nx_n$$

$$\text{subject to constraint:} \begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq b_2 \\ \ldots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n \leq b_n \end{cases}$$

Back to Agenda Page

Minimize Costs of Electricity Grids

The Electricity Grid



The Equivalent Linear Program

```
Minimize cost:
    30 * x + 50 * z
Such that:
CleanVille's power is balanced
    x - y = 100
DirtyVille's power is balanced
    y + z = 100
The hydro plant operates within its limits
    0 ≤ x ≤ 190
The gas plant operates within its limits
    0 ≤ z ≤ 200
The transmission line is within its limits
    -80 ≤ y ≤ 80
```

*This is one of many possible models! Depending on your application you might want to consider transmission losses, intracity distribution networks, etc.

Back to Agenda Page

The Linear Programmnig Approaches

SIMPLEX

From Dantzig, 1947. Do lots of small matrix 'pivot' operations until the exact solution is found. Very numerically stable but slower for massive problems.

STANDARD SIMPLEX

Simplest to understand but uses too much memory. Not used in practice.

REVISED SIMPLEX

Represents the matrix in a memory-efficient way. Used in practice.

(REVISED) DUAL SIMPLEX

Same as the previous but operates on the equivalent 'dual' problem.

INTERIOR-POINT

From Dikin, 1967. Repeatedly takes 'steps' to approach the solution. Faster for massive problems but I've found it is prone to numerically unstability.

BARRIER

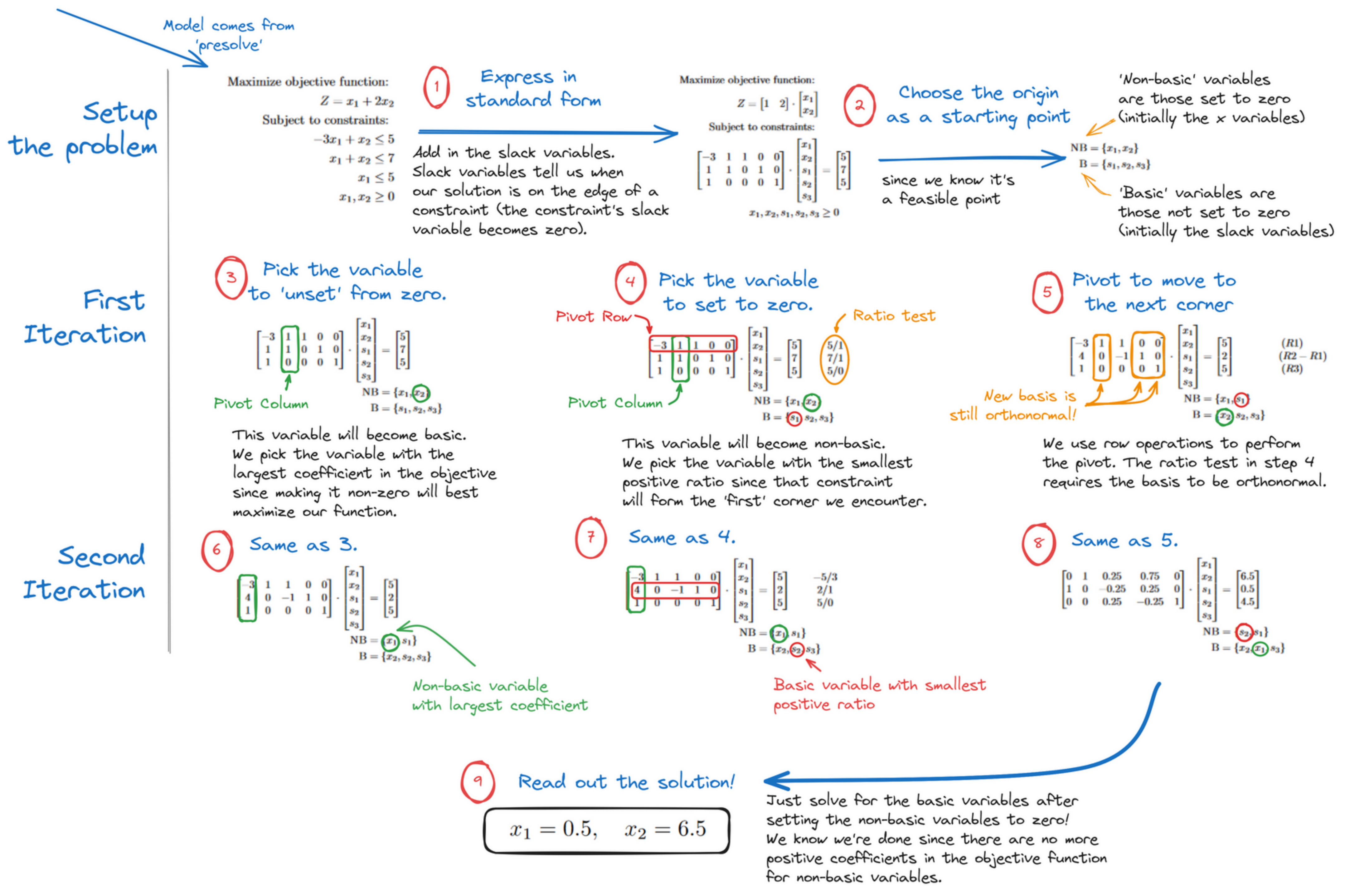
Uses a 'barrier' term to move the constraints into the objective.

PRIMAL-DUAL

Similar to barrier but solves the primal & dual problem simultaneously.

Back to Agenda Page

The Simplex Algorithm



Back to Agenda Page

What are we going to do?

Back to Agenda Page

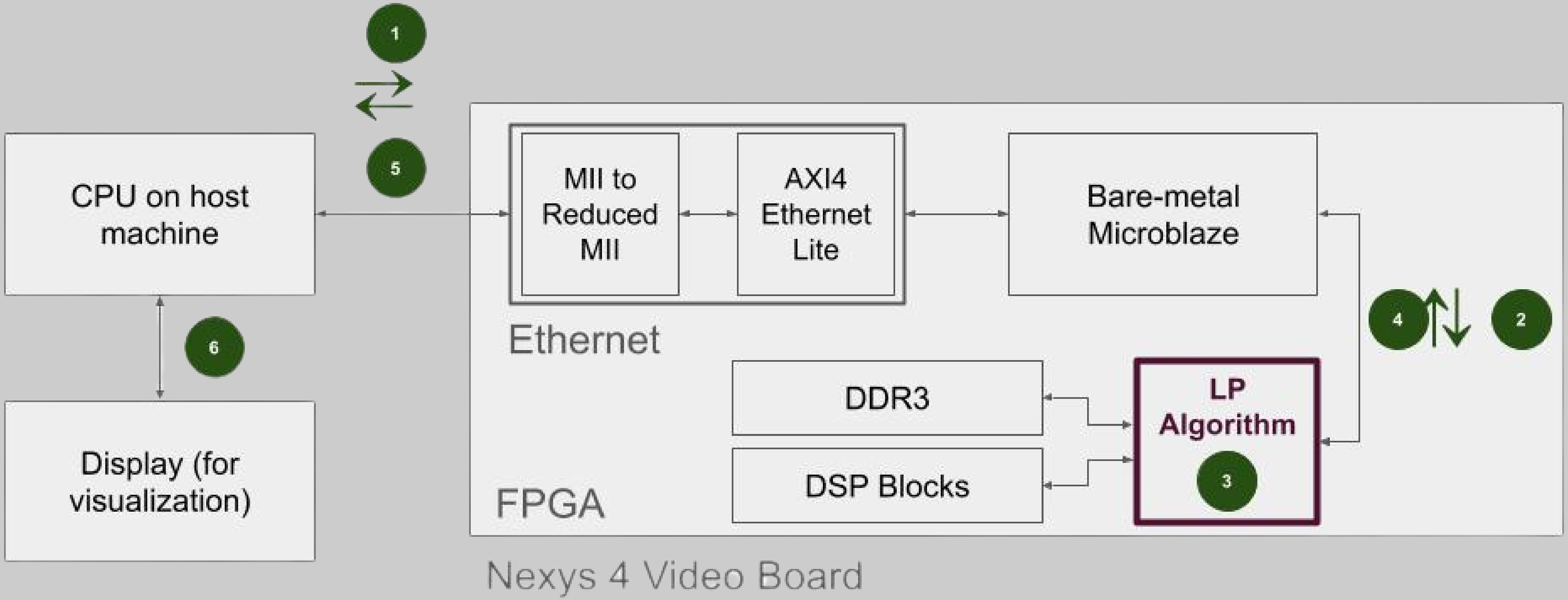
to do • What to do • What to do • What to do • What to do • What to do • What to do • What to do • What to do • What to do • What

Implement a linear programming solver on an FPGA

Solve such problems using an implementation of the Simplex Method

Back to Agenda Page

What does it look like



Implementation Plan

Milestone	Objective	Deliberables
1	Improve familiarity with Linear Programming	Have members attempt to research, investigate, and implement different algorithms in Python to solve a sample problem
2	**Pick specific approach**	**Team members pick a specific algorithm and implement a version in C**
3	Core Development 1	Work on LP core and memory access
4	Core Development 2	Get “80%” of functionality in-place - components should be working individually, with minor bug-fixes
5	System Integration	Get all cores working and integrated together
6	GUI Development	Develop a GUI to visualize solver progress
	Performance Benchmarking + Buffer	Some added buffer for project + optional performance benchmarking

Back to Agenda Page

2 Dimensions of Testing

Functional - Does it Work?

- Ongoing functional validation
- Behavioural Simulation in Vivado of HDL testbenches for Custom IPs
- Software testbenches for software-based parts of the solver (running on the embedded Microblaze processor)

Performance - How Well?

- Later in the development process
- Benchmarking against commercial solvers to observe performance, efficiency, and scalability
- Results may influence design decisions if have time for modifications

Back to Agenda Page

Risks and Uncertainties

Back to Agenda Page

What are we not sure about?

Model Sizes

- Existing FPGA solvers run small-sized models (~80 variables)
- Goal is >1000 variables
- Scaling input size & constraints

Complexity of Algorithms

- Different solving algorithms
- Runtime complexity --> Performance
- Simplex vs Revised Simplex
- Feasability of implementation

Back to Agenda Page

Major Risks

Time

- Only 7 milestones total
- Need to keep implementation reasonable
- Prioritizing functional goals
- Scaling & performance goals can become auxiliary if needed

Resources

- Limited by course board options & on-board resources
- Any damage = major hindrance for project progress
- Onboard FPGA testing should only be done once designs have been synthesized, verified, etc.

Memory

- 128MB DDR2 on Nexys DDR vs 512MB DDR3 on Nexys Video
- Some existing implementations already run on the Virtex 7
- Risk involved with potential scaling to large models

Back to Agenda Page

What are we NOT going to do?

Back to Agenda Page

NOT to do • What NOT to do • What NOT to do • What NOT to do • What NOT to do • What NOT to do • What NOT to do • What NOT to

Though it is cool to do...

Running Commercial Models

- Cannot run large, commercial models due to memory contraints - too large for Nexys boards
- However, study of such models can uncover different opportunities for additional performance

Multiple Algorithms

- Some solvers can run multiple different algorithms at the same time and terminate upon the first exit
- Interesting from a performance perspective, but we are resource-constrained

Back to Agenda Page

Thank you!

All Questions are Welcome!