COMP9334: Capacity Planning of Computer Systems and Networks

Week 7B: Optimisation (1): Linear programming

Three Weeks of Optimisation

- The lectures for next three weeks will focus on optimization methods for network related design and applications

- You will learn:
  - How to formulate optimization problems
  - Tools to solve optimization problems

- An introduction only, because optimization is a big topic
  - Emphasis is on applying optimization methods rather than the theory behind

Motivation (1)

- A modern approach to managing computer networks is based on the concept of *software-defined networking*

- Two types of nodes:

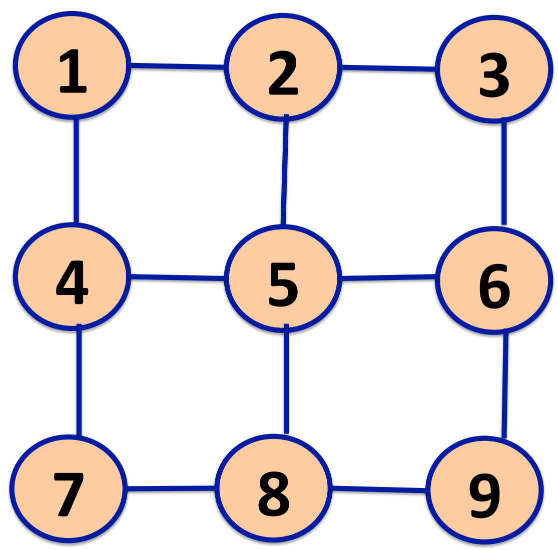
1. Simple packet switches
2. Controllers

- A controller can control a number of simple packet switches but they must be placed in a strategic location in the network

- If the delay between the controller and a packet switch is too long, then it can degrade the network performance

Motivatiation (2)

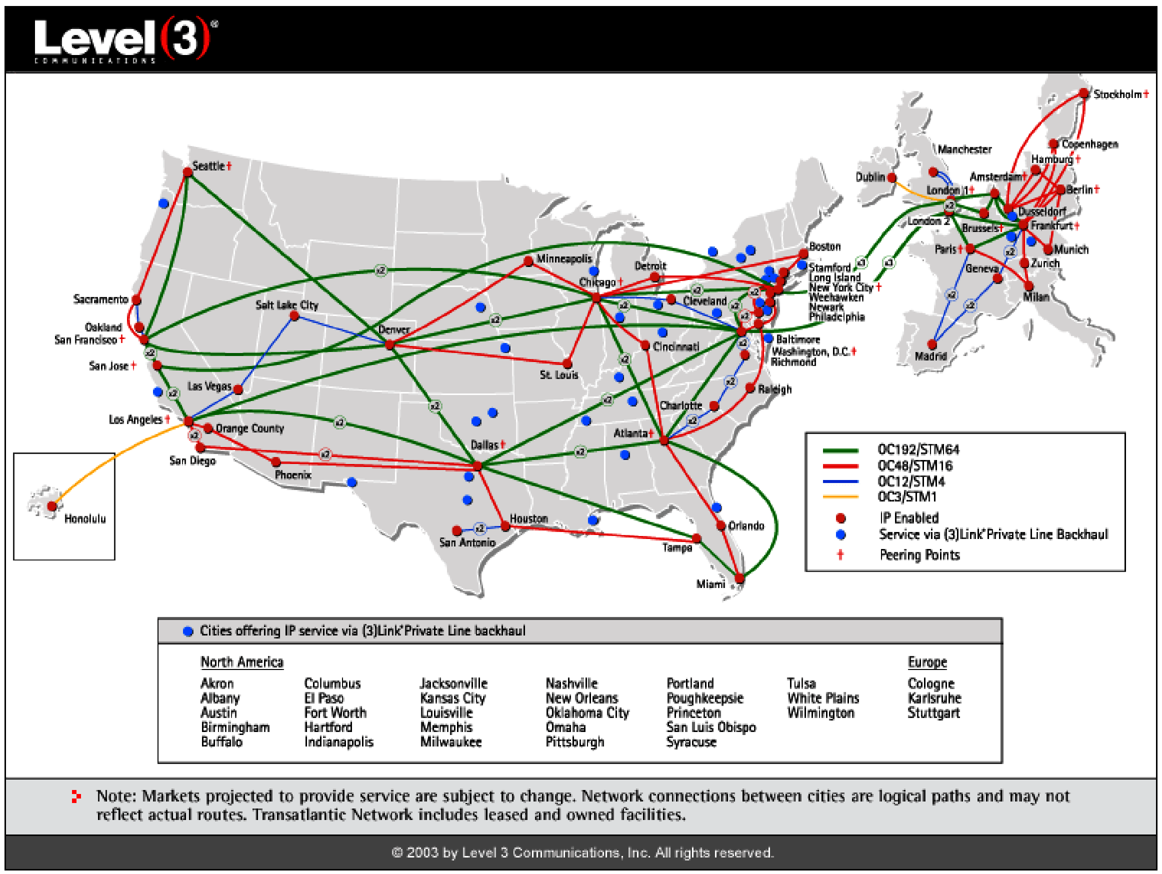
- Consider the following network where there is a packet switch at each node and the delay on each link is 1 time unit.
- Question: Assuming you want to place one controller in the network, where will you place the controller?



- Question: What if you want to place two controllers?

Motivation (3)

- How about solving the same problem for a large heterogeneous network?



- Optimisation provides a systematic method to make decisions

Elements of an optimisation problem

- You want to maximise your WAM and still have a life

Maximise WAM(x_1,x_2,x_3,...)

x_1 hours/week on COMP9334

x_2 hours/week on COMPxxxx

x_3 hours/week on socialising

$x_1 \geq 10$

$x_2 \leq$ maxSocialHours

$x_1 + x_2 + x_3 + \ldots \leq$ totalAwakeHours

- Elements of an optimisation problem
  - Minimise or maximise an objective function
  - Decision variables: x_1,x_2, ... etc.
  - Constraints

What is optimization?

- In mathematics, also known as mathematical programming
  - The term programming refers to planning of activities to obtain an optimal result, not computer programming
  - The amount or level of each activity can be represented as a variable whose value is to be determined
- Optimization means solving problems in which we seek to minimize or maximize the value of an objective function of many decision variables, subject to constraints on the decision variables

Reference books

- Winston, “Operations Research”, 4th edition
  - Examples from this book tend to come from manufacturing, business, finance, etc
  - The abstraction power of mathematics means many optimization problems have similar mathematical formulation
  - Very often an optimization problem in networking may have a similar cousin in other application areas, and their mathematical formulation are identical
- Ahuja, Magnanti and Orlin, “Network Flows”
- Fourer, Gay and Kernighan, “AMPL: A Modeling Language for Mathematical Programming”, 2nd edition

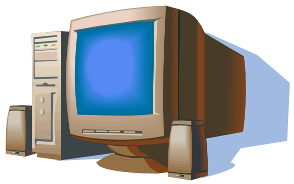
Software

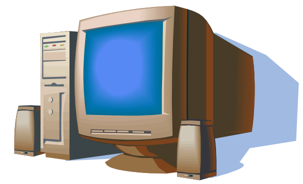
- Modeling language: AMPL and Solver CPLEX
  - High-level programming language for describing optimization problems
  - Syntax similar to mathematical formulation of optimization problems
  - Demo version of the software is available for download from: `http://www.ampl.com`. Click *Try AMPL*, then *Download a Free Demo*
- Note: Demo version of AMPL/CPLEX is full-featured but limited to 500 variables and 500 objectives plus constraints

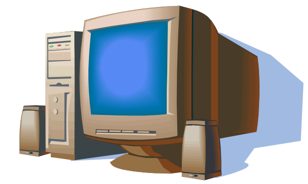
Motivating example 1: Cloud/Grid computing

- Service providers sell computing power as an utility
  - Computing power measured in CPU cycles
- Target customers
  - Financial company, pharmaceutical company, etc.
- Quality of Service in Cloud computing
  - Different service providers might offer the service at different levels for different costs
  - Optimization problem: How to select service providers (allocate resources) to achieve the best level of service without exceeding budget

Cloud computing resource allocation







Resource 1
Speed: 1,000 million cycles/sec
Cost: 0.1 dollars/sec

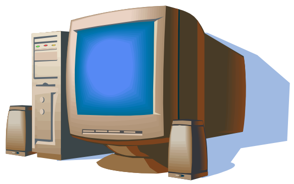
Resource 2
Speed: 2,000 million cycles/sec
Cost: 0.25 dollars/sec

Resource 3
Speed: 3,000 million cycles/sec
Cost: 0.6 dollars/sec

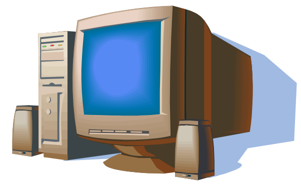
- A computation job:
  - Requires 10^7 million cycles
  - Must be completed in at most 4,800 sec
  - Cost must not exceed 1,500 dollars

- Exercises: For the time being, let us ignore the constraint on the completion time and cost.
  - If you use Resource 1 only, what is the completion time and cost?
  - Repeat for Resources 2 and 3.

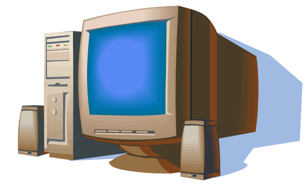
Cloud computing resource allocation (cont.)



Resource 1
Speed: 1,000 million cycles/sec
Cost: 0.1 dollars/sec



Resource 2
Speed: 2,000 million cycles/sec
Cost: 0.25 dollars/sec

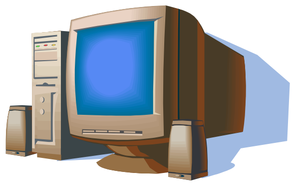


Resource 3
Speed: 3,000 million cycles/sec
Cost: 0.6 dollars/sec

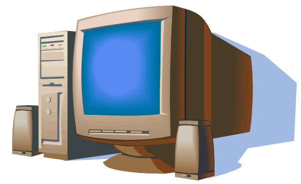
- A computation job:
  - Requires 10^7 million cycles
  - Must be completed in at most 4,800 sec
  - Cost must not exceed 1,500 dollars

- Completion time and cost for each resource:
  - Resource 1: Completion time = 10,000 sec, cost = 1,000 dollars
  - Resource 2: Completion time = 5,000 sec, cost = 1,250 dollars
  - Resource 3: Completion time = 3,333 sec, cost = 2,000 dollars

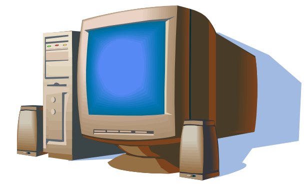
Cloud computing resource allocation (cont.)



Resource 1
Speed: 1,000 million cycles/sec
Cost: 0.1 dollars/sec



Resource 2
Speed: 2,000 million cycles/sec
Cost: 0.25 dollars/sec

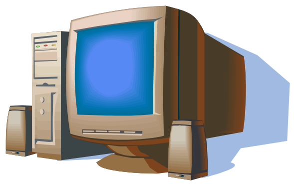


Resource 3
Speed: 3,000 million cycles/sec
Cost: 0.6 dollars/sec

- Assume the computation job can be arbitrarily split into up to three parallel tasks

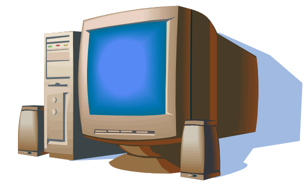
- Question: How should the job be split, so that completion time T is minimized subject to two constraints:
  - Completion time constraint: $T \leq$ 4,800 sec
  - Cost constraint: $C \leq$ 1,500 dollars

Trial and error: Solution 1



Resource 1
Speed: 1,000 million cycles/sec
Cost: 0.1 dollars/sec

Resource 2
Speed: 2,000 million cycles/sec
Cost: 0.25 dollars/sec



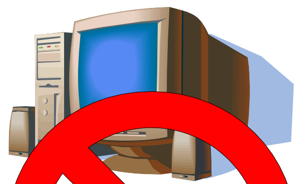
Resource 3
Speed: 3,000 million cycles/sec
Cost: 0.6 dollars/sec

- 48% to Resource 1, 52% to Resource 3
  - Resource 1: Completion time = 4,800 sec, cost = 480 dollars
  - Resource 3: Completion time = 1,733 sec, cost = 1,040 dollars
- Job completion time = 4,800 sec (remember jobs run in parallel)
- cost = 1,520 dollars, Infeasible solution

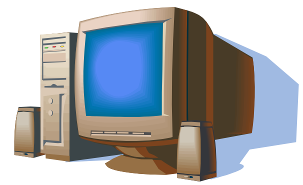
Terminology

- A solution is feasible if all the constraints are satisfied
- A solution is infeasible if not all the constraints are satisfied

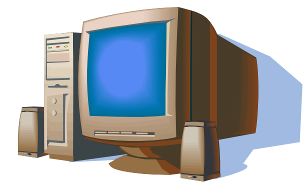
Trial and error: Solution 2



Resource 1
Speed: 1,000 million cycles/sec
Cost: 0.1 dollars/sec



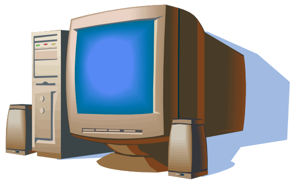
Resource 2
Speed: 2,000 million cycles/sec
Cost: 0.25 dollars/sec

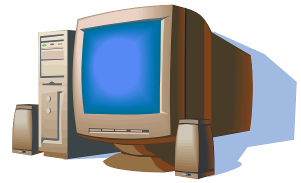


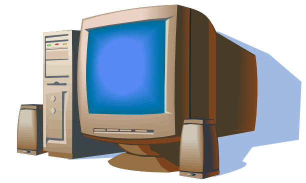
Resource 3
Speed: 3,000 million cycles/sec
Cost: 0.6 dollars/sec

- 70% to Resource 2, 30% to Resource 3
  - Resource 2: Completion time = 3,500 sec, cost = 875 dollars
  - Resource 3: Completion time = 1,000 sec, cost = 600 dollars
- Job completion time = 3,500 sec, cost = 1,475 dollars
- Feasible solution

Trial and error: Solution 3







Resource 1
Speed: 1,000 million cycles/sec
Cost: 0.1 dollars/sec

Resource 2
Speed: 2,000 million cycles/sec
Cost: 0.25 dollars/sec

Resource 3
Speed: 3,000 million cycles/sec
Cost: 0.6 dollars/sec

- 30% to Resource 1, 30% to Resource 2, 40% to Resource 3
  - Resource 1: Completion time = 3,000 sec, cost = 300 dollars
  - Resource 2: Completion time = 1,500 sec, cost = 375 dollars
  - Resource 3: Completion time = 1,333 sec, cost = 800 dollars
- Job completion time = 3,000 sec, cost = 1,475 dollars
- Feasible solution

Optimizing resource allocation

- Given:
  - Job requirement = 10^7 million cycles
  - Completion time $\leq$ 4,800 sec
  - Budget $\leq$ 1,500 dollars
- Let:
  - x_1 = fraction of the job to Resource 1
  - x_2 = fraction of the job to Resource 2
  - x_3 = fraction of the job to Resource 3
- Find x_1, x_2 and x_3 such that
  - All requirements are met
  - Completion time is minimized

Formulating optimization problem

- Completion time:
  - Resource 1 = $\frac{10^7 \times x_1}{1000} = 10000 \times x_1$
  - Resource 2 = $\frac{10^7 \times x_2}{2000} = 5000 \times x_2$
  - Resource 3 = $\frac{10^7 \times x_3}{3000} = \frac{10000}{3} \times x_3$
  - Job completion time $T = \max(10000 \times x_1, 5000 \times x_2, \frac{10000}{3} \times x_3)$
- Cost:
  - Resource 1 = $0.1 \times 10000 \times x_1 = 1000 \times x_1$
  - Resource 2 = $0.25 \times 5000 \times x_2 = 1250 \times x_2$
  - Resource 3 = $0.6 \times \frac{10000}{3} \times x_3 = 2000 \times x_3$
  - Cost $C = 1000 \times x_1 + 1250 \times x_2 + 2000 \times x_3$

Formulating optimization problem (cont.)

- Mathematically, the optimization problem can be formulated as

$$\min \quad T$$

subject to

$$\begin{aligned}
T &\geq 10000 \times x_1 \\
T &\geq 5000 \times x_2 \\
T &\geq \frac{10000}{3} \times x_3 \\
T &\leq 4800 \\
1000 \times x_1 + 1250 \times x_2 + 2000 \times x_3 &\leq 1500 \\
x_1 + x_2 + x_3 &= 1 \\
x_1, x_2, x_3 &\geq 0
\end{aligned}$$

Components of an optimization problem

- Given parameters
- Decision variables
  - In this example, they are x_1, x_2, x_3 and T
- Objective function
  - Can be minimization or maximization
  - Can be single objective or multi-objective
- Constraints

Exercise

- Consider the following optimization problem where x is the decision variable:

$$\min_{x} \quad 2x - 1$$

subject to

$$\begin{aligned} x &\leq 20 \\ x &\geq 8 \end{aligned}$$

- What are the feasible solutions?

- What is the optimal solution?

LP solvers

- Many commercial and free software are available for solving LP problems
- Commercial software
  - Capable of solving large LP problems, e.g. millions of variables
  - A 50,000-variable LP problem takes about 5 seconds on a standard linux PC
  - You can try out many commercial solvers at the NEOS web site
    - `https://neos-server.org/neos/`
- Free software / demo version
  - `http://www.ampl.com`
  - `http://ampl.com/try-ampl/download-a-demo-version/`

LP solvers (cont.)

- LP solvers require the user to write the problem in fixed format
- Can be embedded in C, C++ or Java, e.g.

```
model.add(IloMinimize(env,-9*x[0]+x[1]+4*x[2]));
model.add(-x[0]+x[2] == -3);
model.add( x[0]-x[1] <= 1);
```

- Can be used with some modeling languages
  - AMPL
  - MPS
  - GAMS

AMPL/CPLEX for solving example 1

- In AMPL, the grid computing problem formulated earlier becomes

```
var T;
var x_1 >= 0;
var x_2 >= 0;
var x_3 >= 0;
minimize time: T;
subject to T_1: T >= 10000*x_1;
subject to T_2: T >= 5000*x_2;
subject to T_3: T >= 10000/3*x_3;
subject to T_max: T <= 4800;
subject to C_max: 1000*x_1+1250*x_2+2000*x_3 <= 1500;
subject to x_sum: x_1+x_2+x_3 = 1;
```

- This is saved in the file `grid_lp.mod`

AMPL/CPLEX for solving example 1 (cont.)

- The problem can be solved by CPLEX with the batch file `grid_lp_batch`

```
model grid_lp.mod;
option solver cplex;
solve;
display x_1;
display x_2;
display x_3;
display T;
```

AMPL/CPLEX for solving example 1 (Command line version)

- At the ampl command prompt, type `commands grid_lp_batch;`, it returns

```
commands grid_lp_batch;
CPLEX 12.6.0.0: optimal solution; objective 2000
4 dual simplex iterations (1 in phase I)
x_1 = 0.2

x_2 = 0.4

x_3 = 0.4

T = 2000

1000*x_1 + 1250*x_2 + 2000*x_3 = 1500
```

- Note: All these files can be downloaded from the course web site

AMPL/CPLEX for solving example 1 (IDE version)

- At the AMPL prompt, type `commands grid_lp_batch;`
- Need to `reset;` before working on a new problem

Console

AMPL

```
ampl: commands grid_lp_batch;
CPLEX 12.6.1.0: optimal solution; objective 2000
4 dual simplex iterations (1 in phase I)
x_1 = 0.2

x_2 = 0.4

x_3 = 0.4

T = 2000

ampl:
```

Acknowledgment

- Grid computing example based on Menascé and Casalicchio, “QoS in computing”, IEEE Internet Computing, pp. 85–87, Jul./Aug. 2004.