Multicommodity Flows

Alyssa Newman and Zane Showalter-Castorena

What is a Multicommodity Flow?

- Different commodities are sent over a shared network where these commodities interact in some way
- Each commodity can have its own
  - Flow on an arc x_{ij}^k
  - Cost per unit of flow c_{ij}^k
  - Capacities along a particular arc u_{ij}^k
  - Supply at nodes b_i^k
  - Source and sink $\mathrm{s}^{\mathrm{k}}, t^k$

Why Do we Care?

- Jobs
- Research
- Applications – Trains, Scheduling, Networking, dynamic traffic assignment, distribution systems, bandwidth packing problem, etc

Tiny bit of formal math

- Optimality
  - Minimize total cost of all commodities

$$\sum_{1 \le k \le K} c^k x^k$$

- Constraints
  - Total arc capacities are met for all arcs

$$\sum_{1 \le k \le K} x_{ij}^k \le u_{ij}, \qquad \forall (i,j) \in A$$

  - Individual commodity capacities are met on all arcs

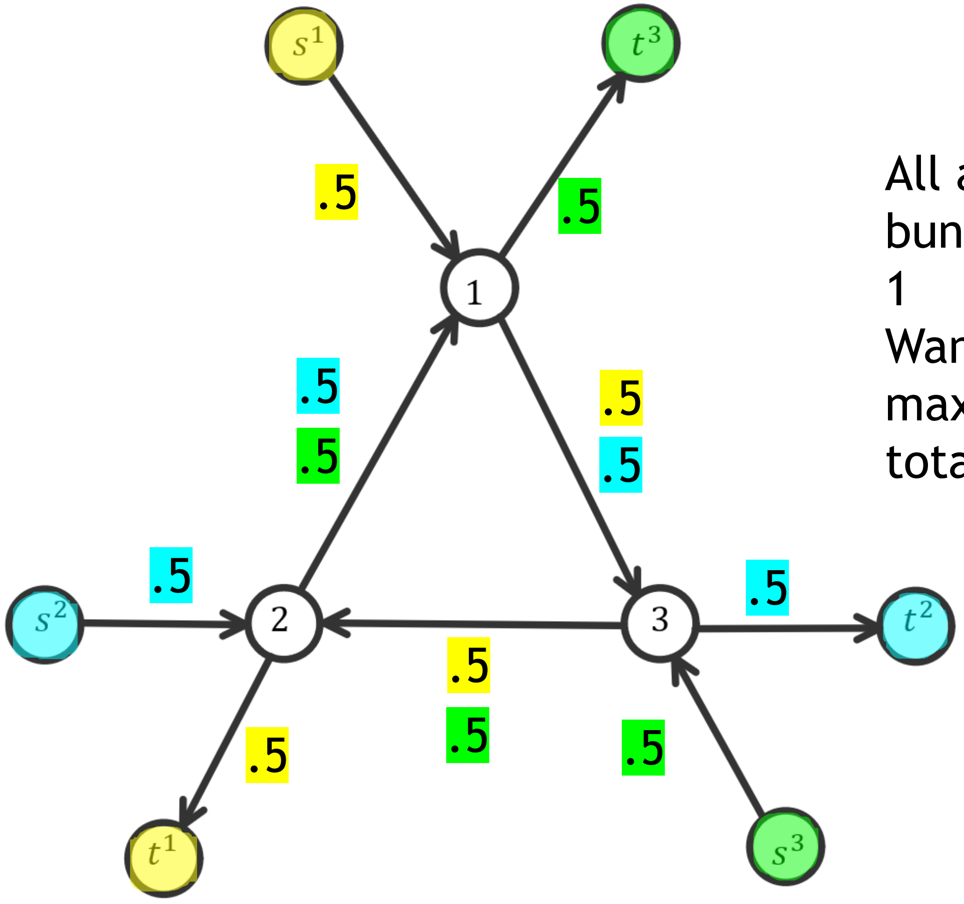
$$0 \le x_{ij}^k \le u_{ij}^k, \forall (i,j) \in A \; and \; 1 \le k \le K$$

  - Supply/demand is met for all commodities

$$\sum_{(j,i) \in A} x_{ji}^k - \sum_{(i,j) \in A} x_{ij}^k = b_i^k, \forall i \in A \; and \; 1 \le k \le K$$

Why these are hard

Having integer valued parameters no longer means we have an integer valued max flow



All arcs have bundle capacity 1

Want to maximize the total flow

Why these are hard

One ‘unit’ of two commodities may not seem to use the same amount of an arc’s total capacity

Inconsistency with what our ‘simplifying assumptions’ should be

General Complexity of the problem

- If want to produce **integer** flow satisfying all demands is NP-complete, even for 2 commodities
  - NP-complete means the problem can **only** be solved in polynomial time if we ‘break’ the ‘normal’ rules of computing.
- If fractions are allowed they can be solved in polynomial time using LP techniques
- What is typically done is answers are approximated with each iteration getting a better approximate.

Methods of solving multicommodity flows

PRICE-DIRECTIVE DECOMPOSITION

RESOURCE-DIRECTIVE DECOMPOSITION

PARTITIONING METHODS

Price Directive Decomposition

- Decompose problem into k subproblems
- Solve by running a minimum cost algorithm on each of the k subproblems

Resource Driven Decomposition

- Decompose the problem into k subproblems
- Sequentially solve each subproblem, then use LP methods to ‘tie’ the solutions back together

Partitioning Methods

- Exploit the fact that multicommodity flow problem is a specially structured LP
- Using the network simplex method, solves a linear program with embedded network flow problems

Lagrangian Relaxation

- Price directive decomposition method
- Split the multicommodity flow into k min cost flow problems
- Associate non-negative weights, called lagrangian multipliers, with every flow
  - There are k number of lagrangian multipliers
  - These multipliers get updated algorithmically m times
    - solve k subproblems m times
- Through LP properties, the optimal solution of the algorithm is the optimal solution of the kth min cost flow
- In the end, the k subproblems all tie back together to solve the original multicommodity flow.

Column Generation

- Special case of multicommodity flow
- Assumptions
  - Each commodity has a single source & sink node, and a single flow requirement between them
  - The only bound on a flow of a commodity through an arc is the maximum capacity of all commodities through that path
  - For every commodity, the cost of every cycle is non-negative
- Decompose the problem into k subproblems and transform the network into a path and cycle flows network.
- Use linear programming to solve each problem, then tie back together.

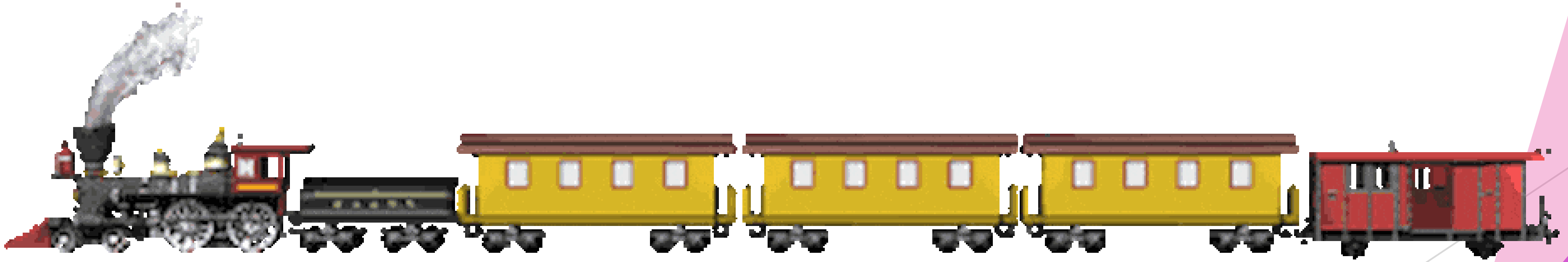
Resource directive decomposition

- All commodities are competing for fixed capacity of an arc
- Decompose the problem into k subproblem.
  - Allocate total capacity across all commodities to individual commodities,
- Rather than solve the original problem directly, it is transformed into a problem that is easier to solve.
- Solution methods include
  - heuristic methods “arc at a time” approach
  - subgradiant optimization methods - LP

Application

Railroad Transportation

- Nodes represent yards and junction points
- Arcs represent tracks between yards
- Demand represented as number of cars per train
- Different goods can be loaded, each with their own demand and cost



Applications – Railroad Transportation

- Nodes represent yards and junction points
- Arcs represent tracks between yards
- Demand represented as number of cars per train
- Different goods can be loaded, each with their own demand and cost

Relaxation in action - Transportation

- *Solving large scale linear multicommodity flow problems with an active set strategy and Proximal-ACCPM*
- Formulated as a linear program combined with optimization criteria
- Result is a numerical solution
- N = Nodes, A = Arcs, K = Commodities, Z* = Optimal Solution

Problem ID	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{K}\rvert$	z^*
Transportation problems				
Sioux-Falls	24	76	528	3.20184×10^5
Winnipeg	1067	2975	4345	2.94065×10^7
Barcelona	1020	2522	7922	3.89400×10^7
Chicago-sketch	933	2950	93513	5.49053×10^6
Chicago-region*	12982	39018	2297945	3.05199×10^6
Philadelphia	13389	40003	1151166	1.65428×10^7

* This problem is solved with a 10^{-4} optimality gap.

Another application

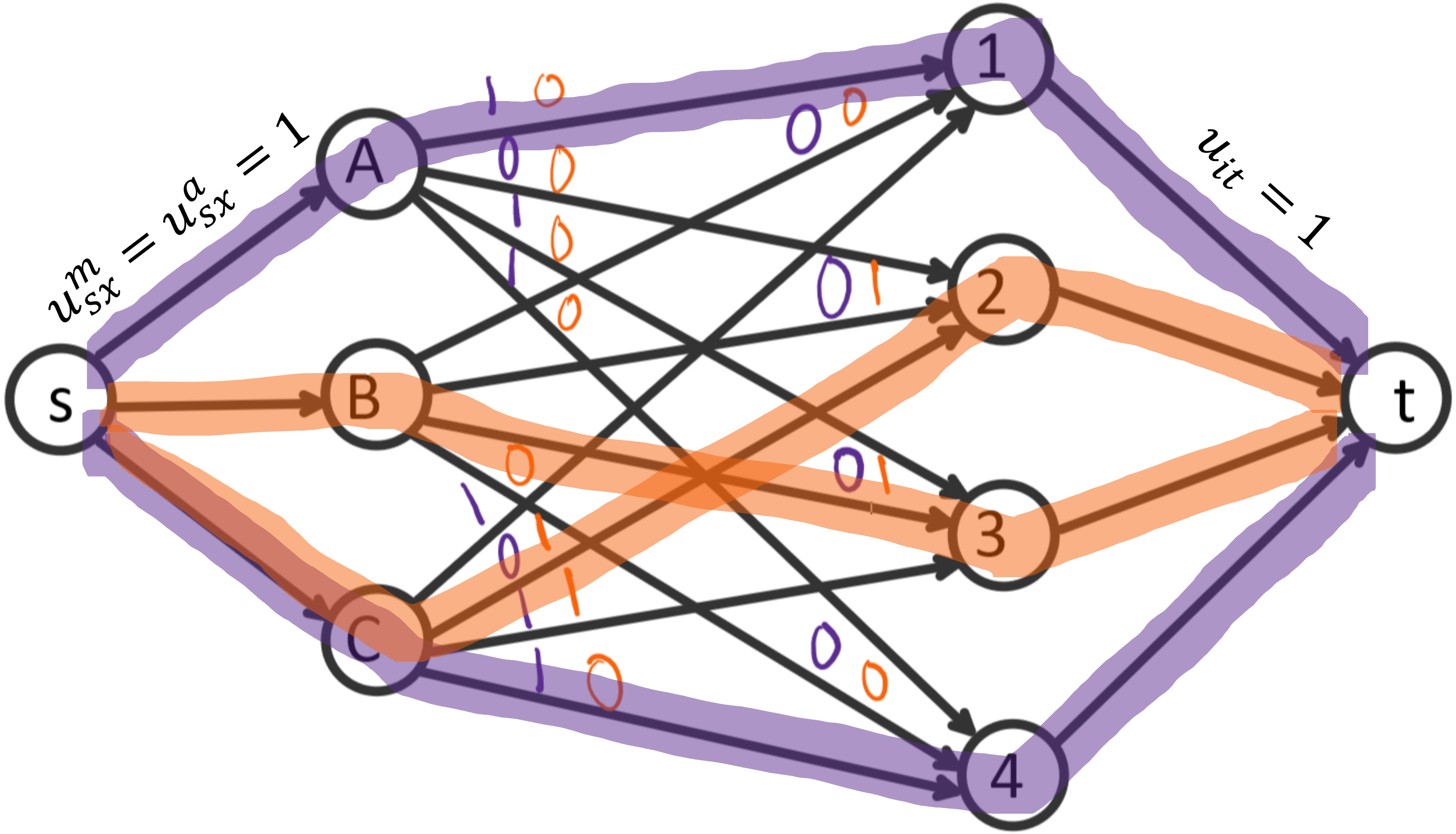
Scheduling:

- Minimize cost to schedule all positions
- People and/or assignments can be nodes
- Feasible assignments are arcs
- Demand represents number of workers needed
- Different types of workers or different shifts are commodities

Teaching allocations example problem

- Simplified example:
  - Two time slots: morning and afternoon
  - 3 teachers: Alice, Bob, Carol
    - Alice only available in the morning, Bob only available In the afternoon, Carol is always available
  - 4 classes: class 1, class 2, class 3, and class 4
    - Classes 1 and 4 can only be offered in the morning, class 2 can only be offered in the afternoon, class 3 can be offered in either time slot

u^{m}_{xi} u^{a}_{xi}



References

- Ahuja, Ravindra K., et al. *Network Flows: Theory, Algorithms, and Applications*. Pearson, 2014.
- Cormen, T. H. (2009). *Introduction to algorithms* (Third ed.). Cambridge, Mass: MIT Press
- Floudas, C. A., & Pardalos, P. M. (2009). *Encyclopedia of optimization* (2nd ed.). New York: Springer.
- Pierce, Rod. (3 Jan 2018). "NP-Complete - A Rough Guide". Math Is Fun. Retrieved 4 May 2020 from http://www.mathsisfun.com/sets/np-complete.html
- “Rail Network Maps: BNSF.” BNSF Railway, 2018, www.bnsf.com/ship-with-bnsf/maps-and-shipping-locations/rail-network-maps.html.
- Solving large scale linear multicommodity flow problems with a n active set strategy and Proximal-ACCPM. (2004, June 1). Retrieved April 29, 2020, from https://pdfs.semanticscholar.org/fba0/1b54d12f415c011907ffdaf439a4f6740f3d.pdf
- *S. Even and A. Itai and A. Shamir (1976). "On the Complexity of Timetable and Multicommodity Flow Problems". SIAM Journal on Computing. SIAM.* ***5*** *(4): 691-703. doi:10.1137/0205048.Even, S.; Itai, A.; Shamir, A. (1975).*

It really do be like that, Questions?