Mathematical Programming-based Heuristics: Telecommunication Network Design meets Species Distribution Planning

David Shmoys
Cornell University

Part of this talk represent joint work with
Bistra Dilkina, Adam Elmachtoub, Ryan Finseth, Dan Sheldon, Jon Conrad, Carla Gomes, Ole Amundsen and Will Allen

What is Mathematical Programming? or really, what is linear programming (LP), and integer (linear) programming (IP)?

Want to

- minimize linear objective function
- subject to linear equality/inequality constraints
- (possibly) requiring the variables to take integer values;

that is, given an n-dimension vector c, an m-dimensional vector b, and an m§ n matrix *A*

minimize cx

subject to *A*x = b, x ¿ 0, (x integer)

Standard assumption:

LP is easy to solve but IP is hard

What is Mathematical Programming? or really, what is linear programming (LP), and integer (linear) programming (IP)?

Want to

- minimize linear objective function
- subject to linear equality/inequality constraints
- (possibly) requiring the variables to take integer values;

that is, given an n-dimension vector c, an m-dimensional vector b, and an m§ n matrix *A*

minimize cx

subject to *A*x = b, x ¿ 0, (x integer)

Standard assumption:

LP is easy to solve but IP is hard (mostly, but not as hard as they used to be) eg 160,000 0-1 vars

The survivable network design problem

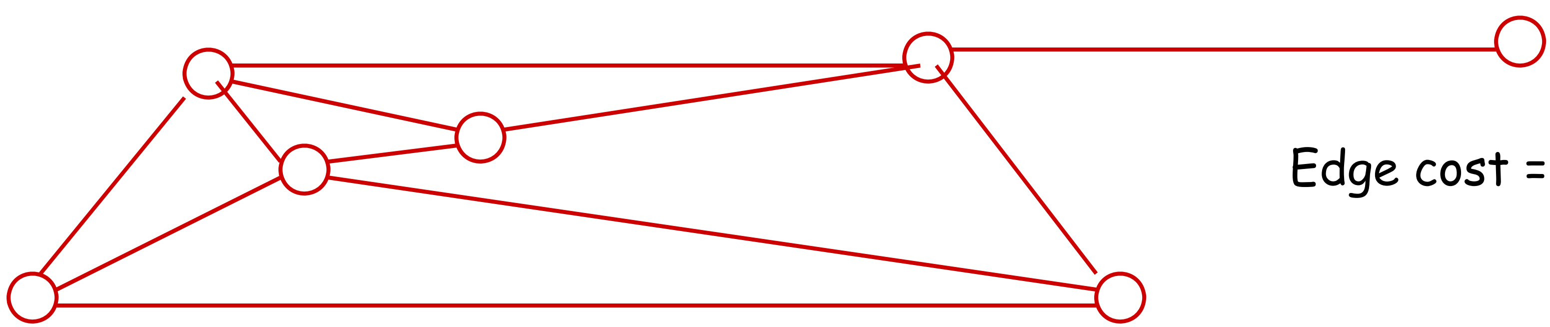
Given: an n-node undirected graph with edge costs and a connectivity requirement r_{ij} for each pair of nodes i,j

Find: a subgraph of minimum cost with the required number of edge-disjoint paths between each i,j

A special case: r_{ij} =1 for each pair of nodes i,j

This is the so-called minimum spanning tree problem

but this is the only "easy" special case.



Edge cost = length

The survivable network design problem

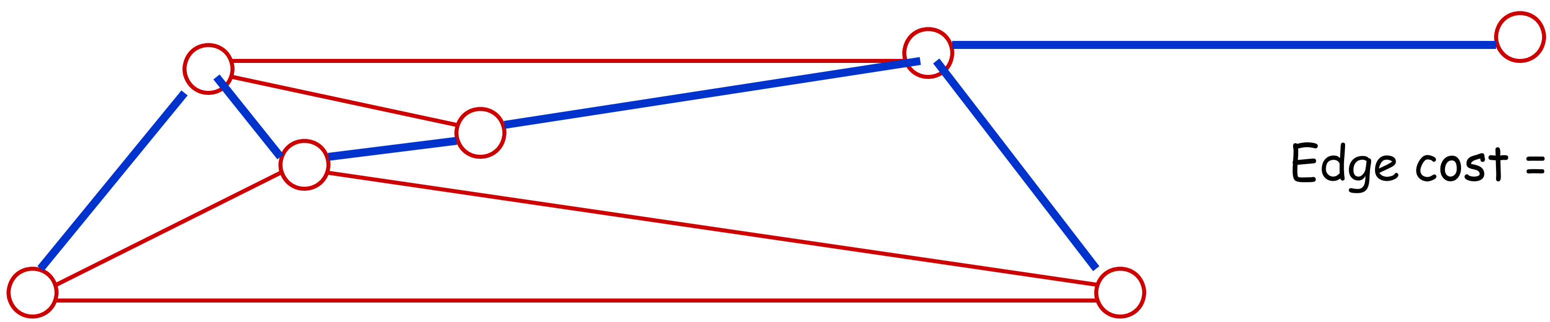
Given: an n-node undirected graph with edge costs and a connectivity requirement r_{ij} for each pair of nodes i,j

Find: a subgraph of minimum cost with the required number of edge-disjoint paths between each i,j

A special case: r_{ij} =1 for each pair of nodes i,j

This is the so-called minimum spanning tree problem

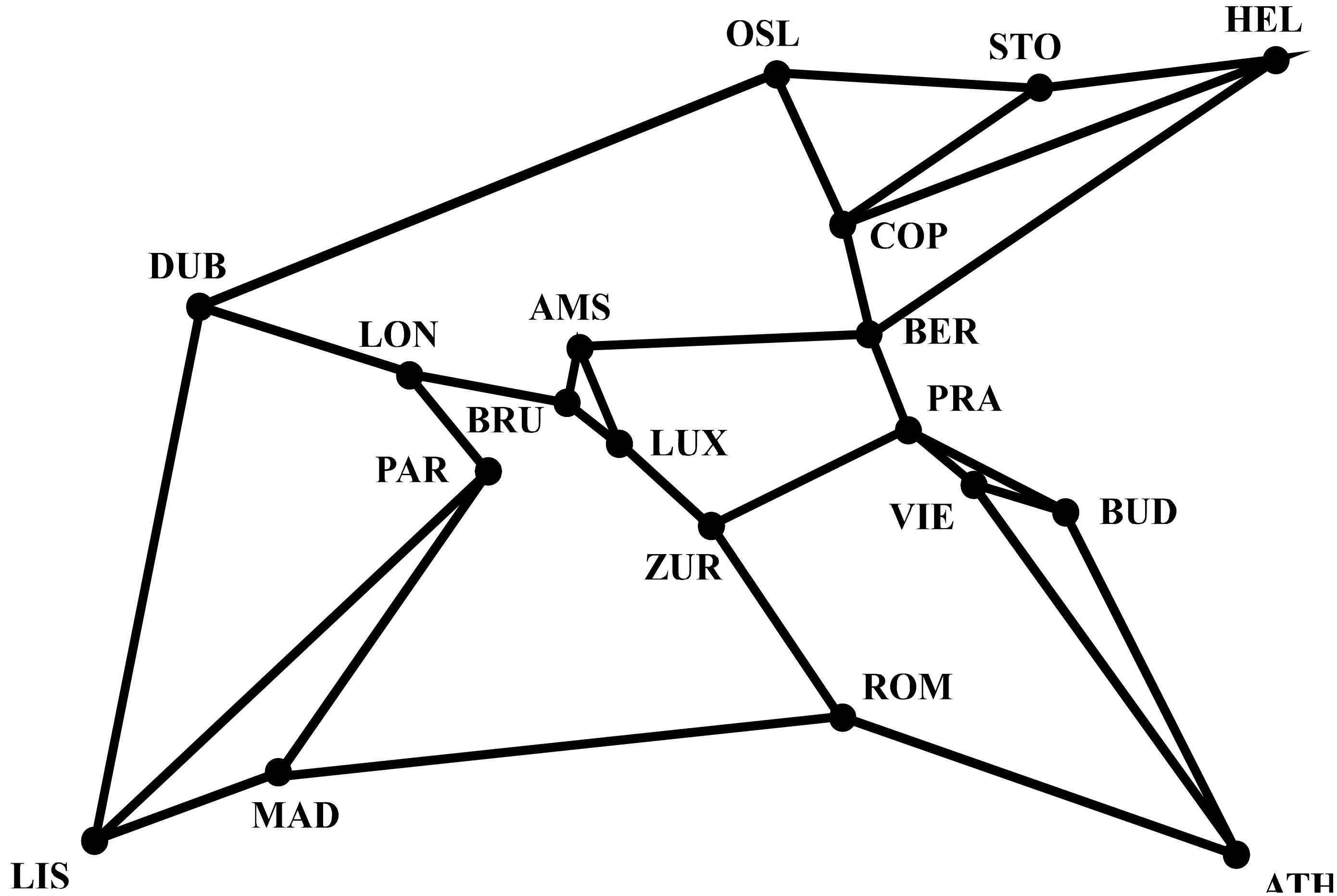
but this is the only "easy" special case.



Edge cost = length

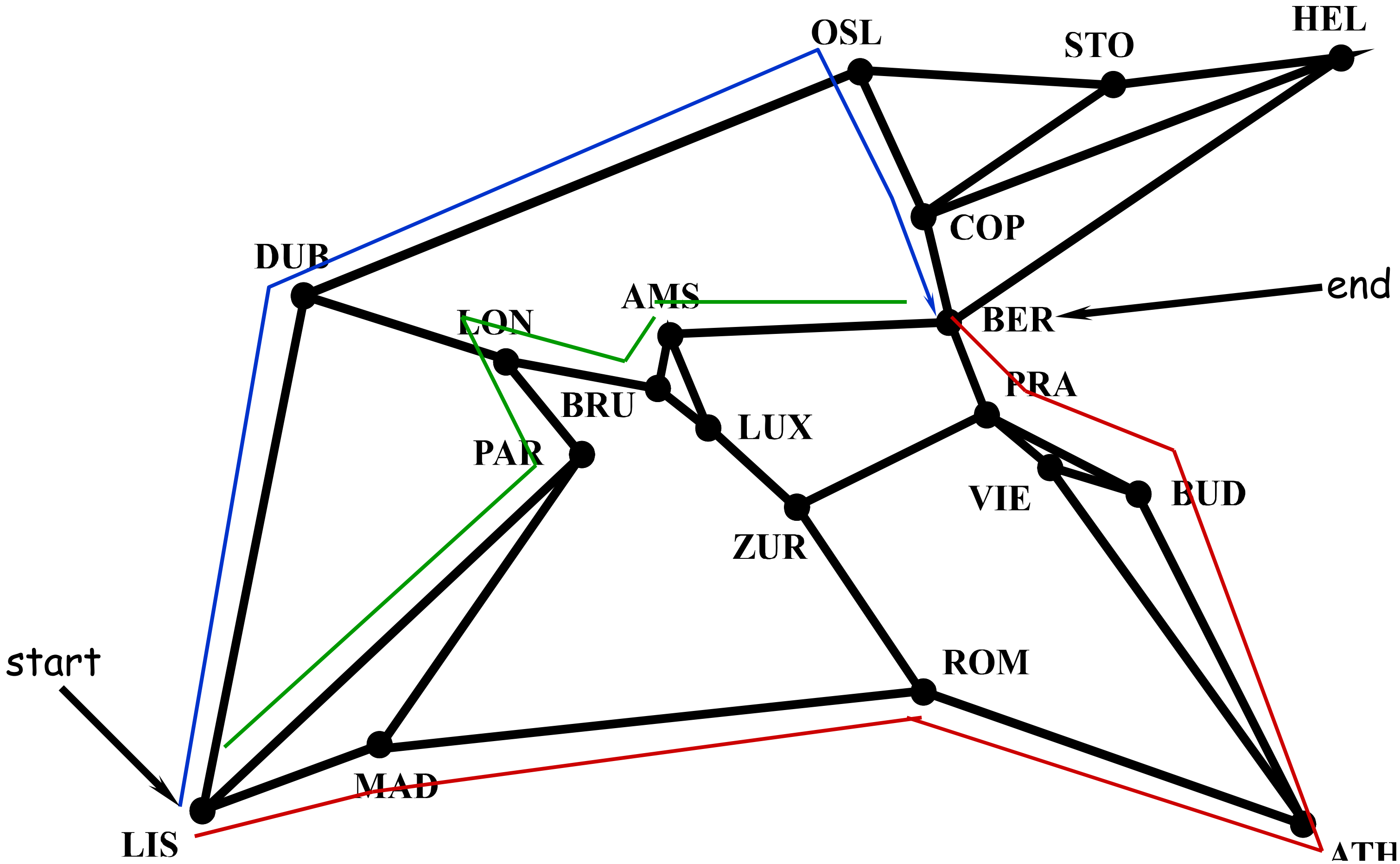
An Example of a 3-connected Solution

Here the input requires for each pair of cities (nodes) that $r_{i,j} = 3$ for all i,j and output is:



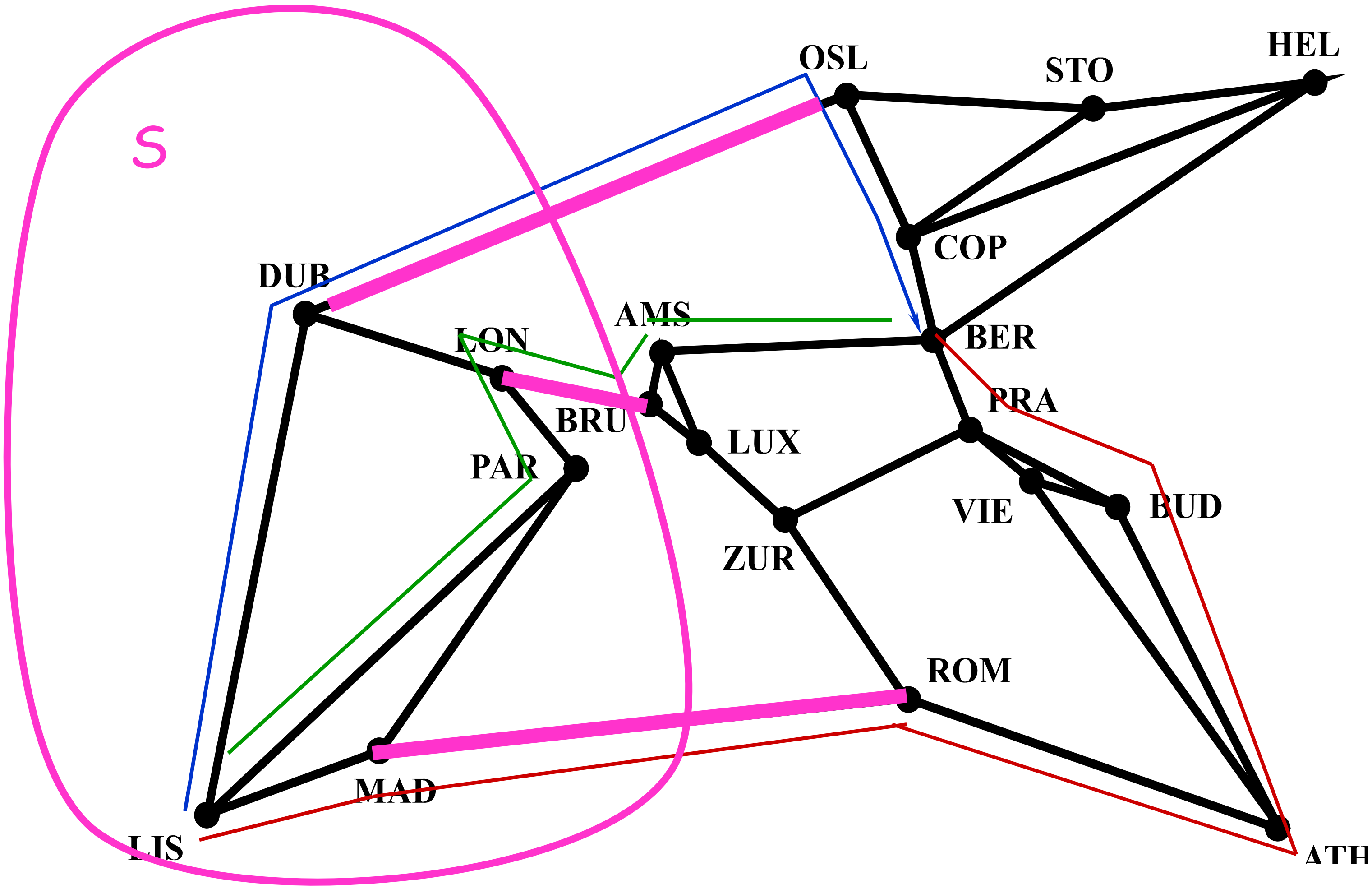
An Example of a 3-connected Solution

Here the input contains a direct connection between each pair of cities (nodes)



An Example of a 3-connected Solution

Here the input contains a direct connection between each pair of cities (nodes)



Using Integer Programming for Network Design

Let $x_e = 1$ denote "include edge e in subgraph"

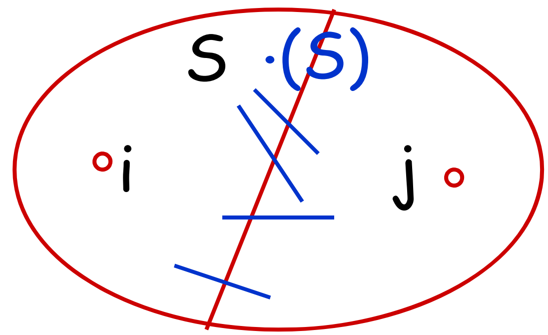
$= 0$ denote "don't include edge e in subgraph"

Objective: minimize $\sum_e c_e x_e$

Subject to: $\sum_{e \in \delta(V)} x_e \geq r_{ij}$

for all i,j and all S s.t. $i \in S$ & $j \in S$

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



Seems hopeless: IP with $O(n^2\, 2^n)$ constraints

But it isn't!! LP gives very strong bound + cutting planes!!

LP-based Heuristics

1. Solve LP "relaxation" (ignore integrality)
2. Find variables that are "nearly 1" (say > .9)
3. Set those variables to 1
4. Resolve to satisfy remaining requirements

Folklore Theorem: (Magnanti et al.) This procedure works (well) in practice, even with side-constraints

Theorem: (Jain) The optimal LP solution always has at least one variable that is at least .5

Corollary: can always find a solution of cost at most twice optimal

LP-based Heuristics

1. Solve LP "relaxation" (ignore integrality) \$ x_e
2. For each edge e, independently set corresponding variable to 1 with probability x_e
3. Resolve to satisfy remaining requirements

Folklore Theorem: (Magnanti et al.) This procedure works (well) in practice, even with side-constraints

Theorem: (Jain) The optimal LP solution always has at least one variable that is at least .5

Corollary: can always find a solution of cost at most twice optimal

An equivalent linear program

Introduce a "commodity" for each pair of nodes i,j

View design decisions x_e as "flow capacities"

Require, for each pair i,j, that a flow of value r_{ij} is possible given these capacities

We shall let $f^{ij}(e)$ denote the flow of the commodity for pair i,j on edge e

Minimize $\sum_e c_e x_e$

Subject to

0 ‰ $f^{ij}(e)$ ‰ x_e ‰ 1 for each e, i, j

$$\sum_{e \text{ entering } k} f^{ij}(e) - \sum_{e \text{ leaving } k} f^{ij}(k) = \begin{cases} 0 & \text{if } k \neq i, j \\ +r_{ij} & \text{if } k = j \\ -r_{ij} & \text{if } k = i \end{cases}$$

Comparing the two LPs

- Flow LP has a lot more variables
- But only a polynomial number of constraints
- Flow LP is suitable for what are called "decomposition methods" that view different flow problem for each commodity separately, linked together by a few additional constraints
- Cut constraints can be efficiently generated on the fly (simple-minded heuristics make this even faster)
- Not always clear which is easier to solve!
- But optimal x is identical for two LPs!

BOTTOM LINE: IP/LP methods solve large-scale inputs

Optimization Models for Red-Cockaded Woodpecker Management

Degradation of and loss of longleaf pine ecosystem has led to the decline of the Red-Cockaded Woodpecker (RCW)

Goal: develop methods to prioritize land aquisition adjacent to current RCW populations

Some (naïve) simplifying assumptions:

- decide now a long-term plan for land acquisition
- assume a simple diffusion model for the population of regions (information cascade eg [Kempe, Kleinberg, Tardos])
- incorporate stochastic model via a sample average approximation approach

Sample Average Approximation

"True" Stochastic Optimization Model

Maximize $E_P(F(x,y))$
subject to y5 Y
where P is a probability distribution over possible inputs x

Sample Average Approximation

Draw m samples $x_1, x_2, \ldots, x_m$ independently from P
and instead

Maximize $(1/m) \sum_i F(x_i,y)$
subject to y 5 Y

Sample Average Approximation

"True" Stochastic Optimization Model

Maximize $E_P(F(x,y))$
subject to y5 Y
where P is a probability distribution over possible inputs x

Strong convergence results (Shapiro) even approximation schemes in some cases (Swamy&S)

Sample Average Approximation

Draw m samples $x_1, x_2, \ldots, x_m$ independently from P
and instead

Maximize $(1/m) \sum_i F(x_i,y)$
subject to y 5 Y

Simple Patch-based Diffusion Model

There is a set $\mathcal{R}$ of regions and a time horizon of T periods

For each region i 5 $\mathcal{R}$ and for each t=1,…,T,

- if the region is occupied at that time, then the territory becomes unoccupied with probability †

For each pair of regions i,j 5 $\mathcal{R}$ and for each t=1,…,T,

- there is a *given* probablity p_{ij}, that, conditioned on the event that region i is occupied at time t-1, that region j is occupied at time t

The transition probabilities were drawn based on the RCW DSS code provided to us by Jeff Walters.

PROBLEM: When do we buy territories and/or make them suitable?

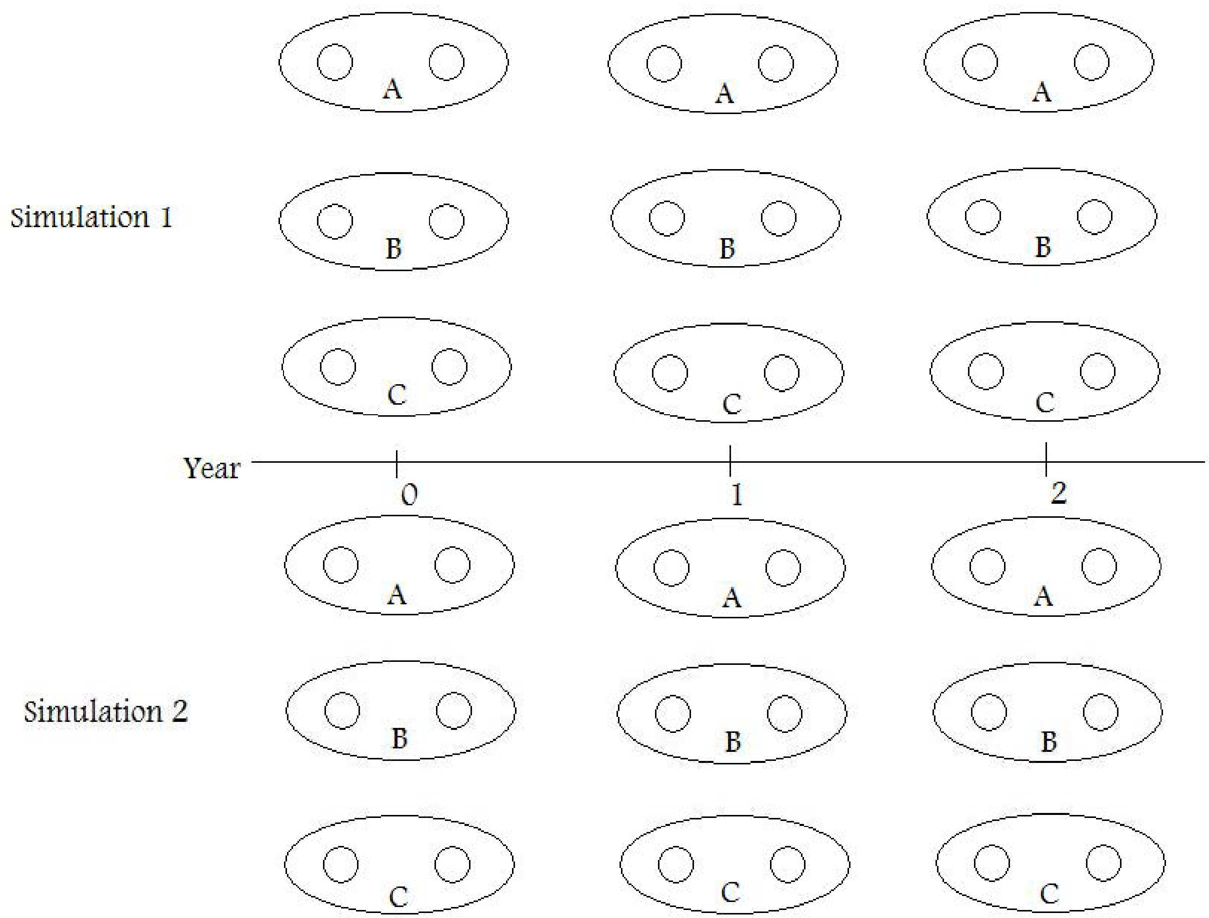
Want to maximize the expected total number of occupied regions at the end of time horizon

Decide to buy/improve certain territories in order to increase the potential number of future occupied territories

Decision effects propagate across the space-time domain

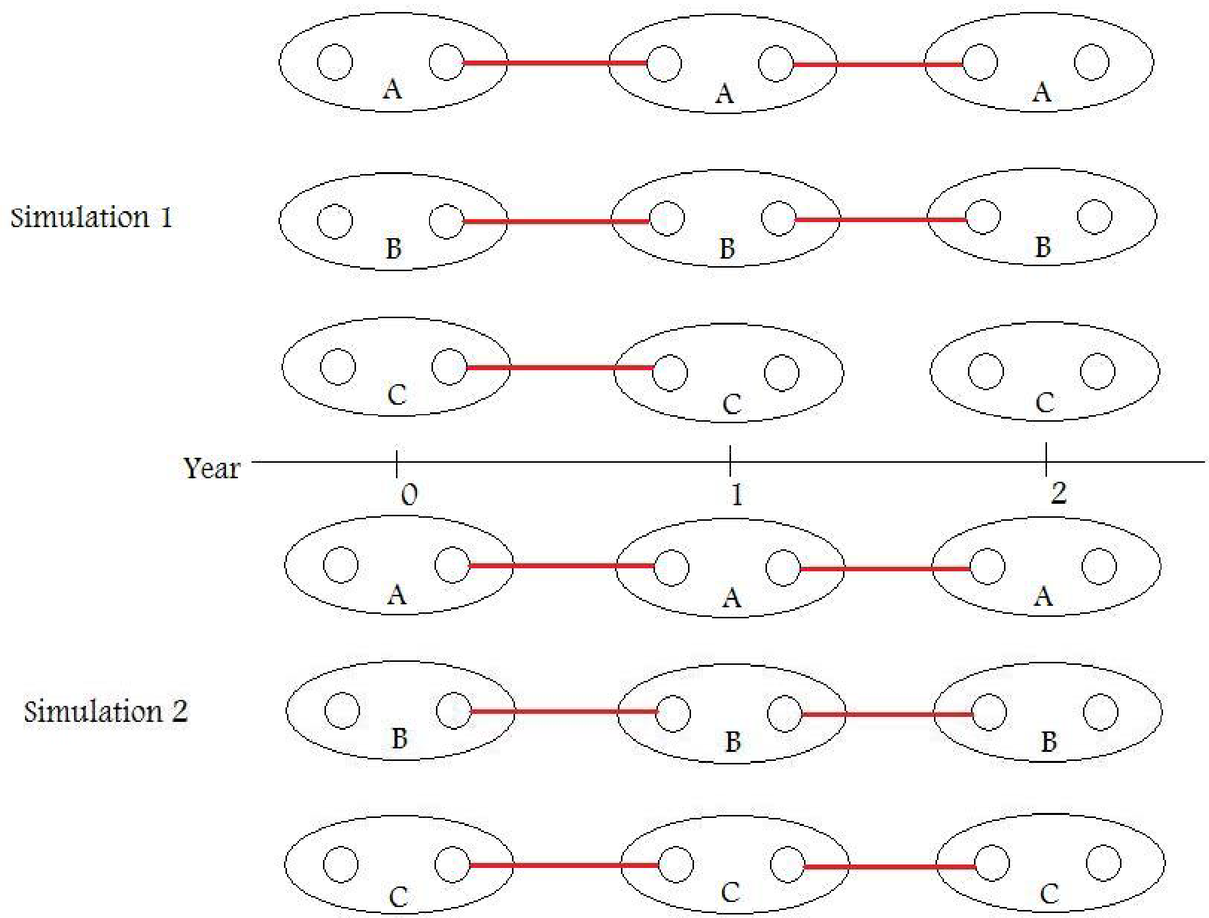
There is a budget constraint that limits the total spent on acquisition/improvement

This can be modeled as a network connectivity problem



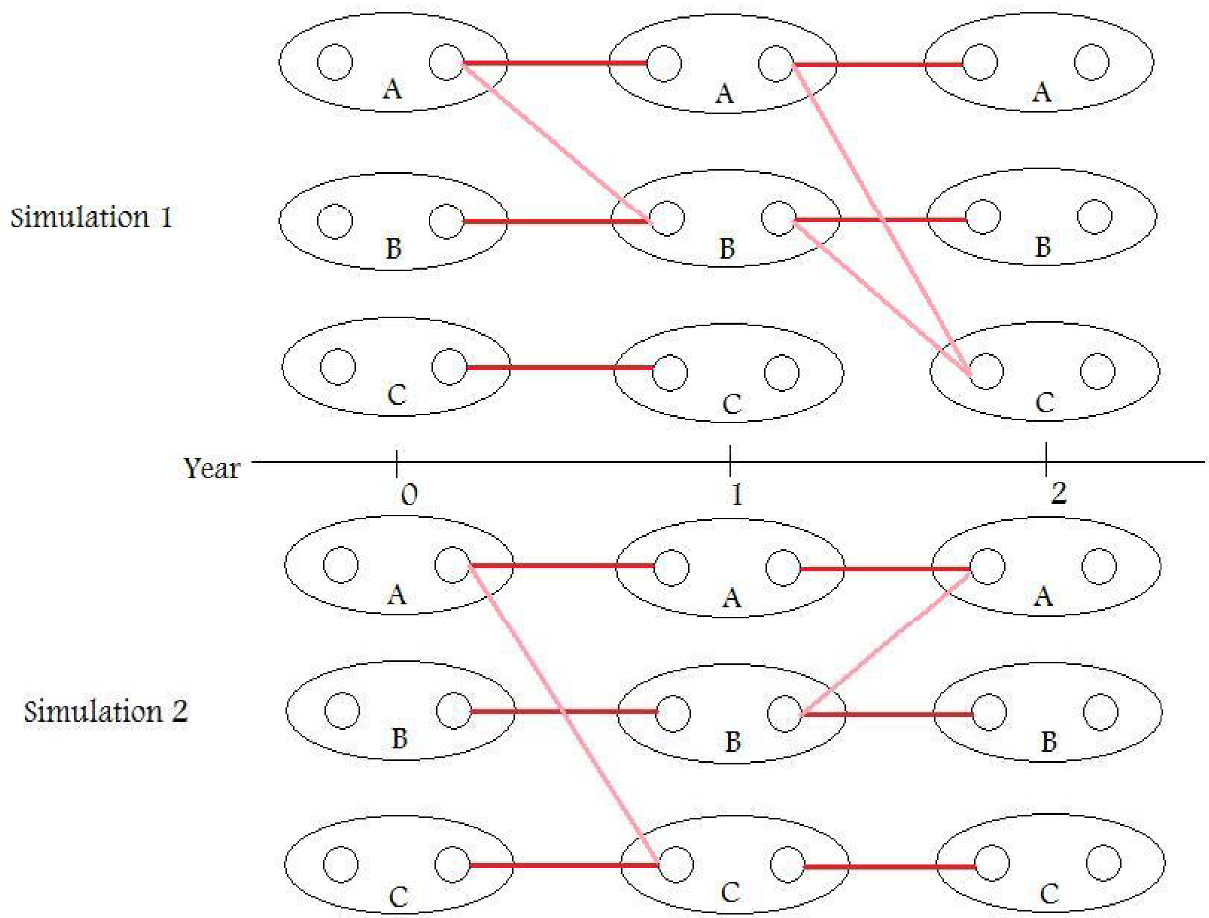
- Territories A,B,C
- 2 Years
- 2 "Trials"

Red lines indicate the chance of a territory remaining occupied in 1 year



- A line from one oval to another represents the ability for a bird from the first territory to colonize the second
- Red lines indicate that if birds occupy a territory, then they will continue occupying it in the next time step
- Birds at C in year 1 in simulation 1 won't make it…

Pink lines indicate the chance that one territory will occupy another

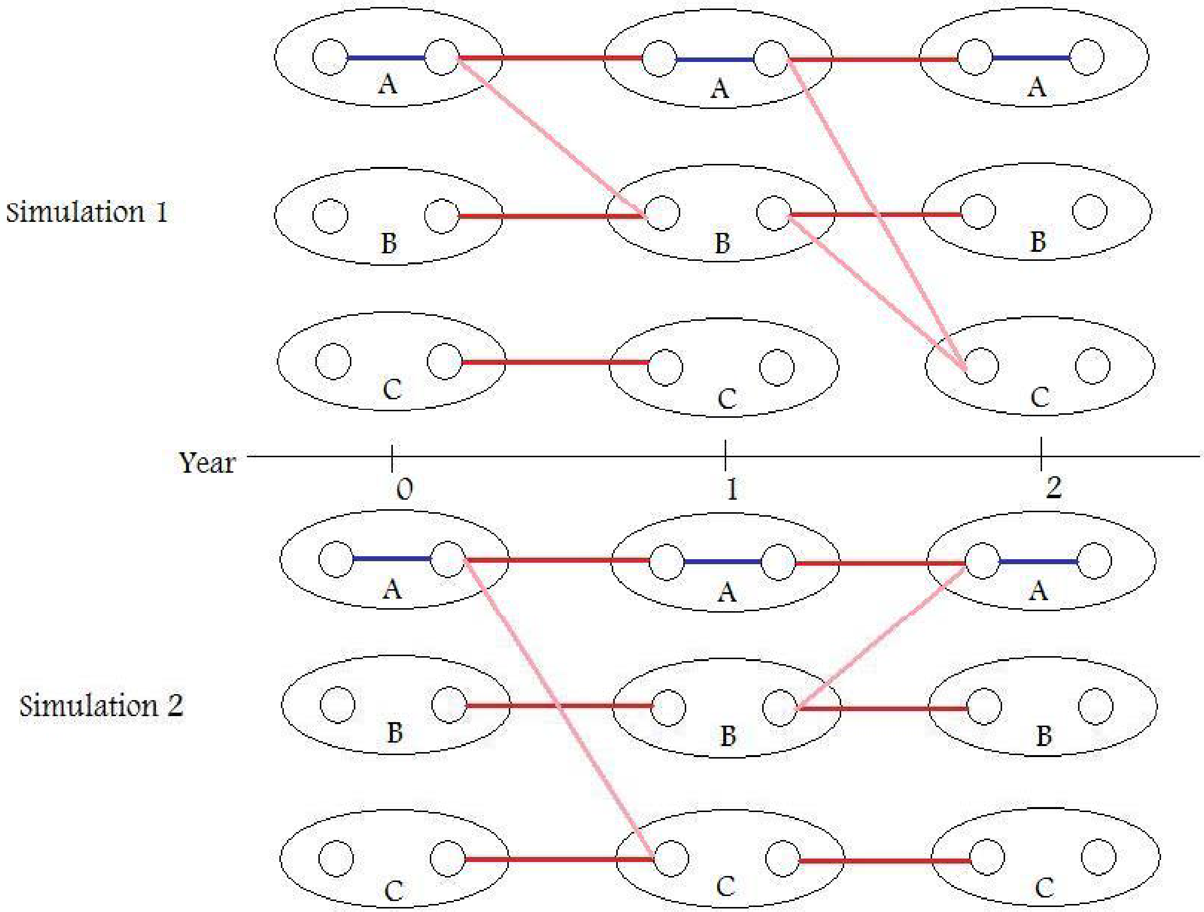


- Using data we can estimate the probability of a bird in one territory occupying another territory in one time step

- The pink lines represent the outcomes of the simulation using these probabilities

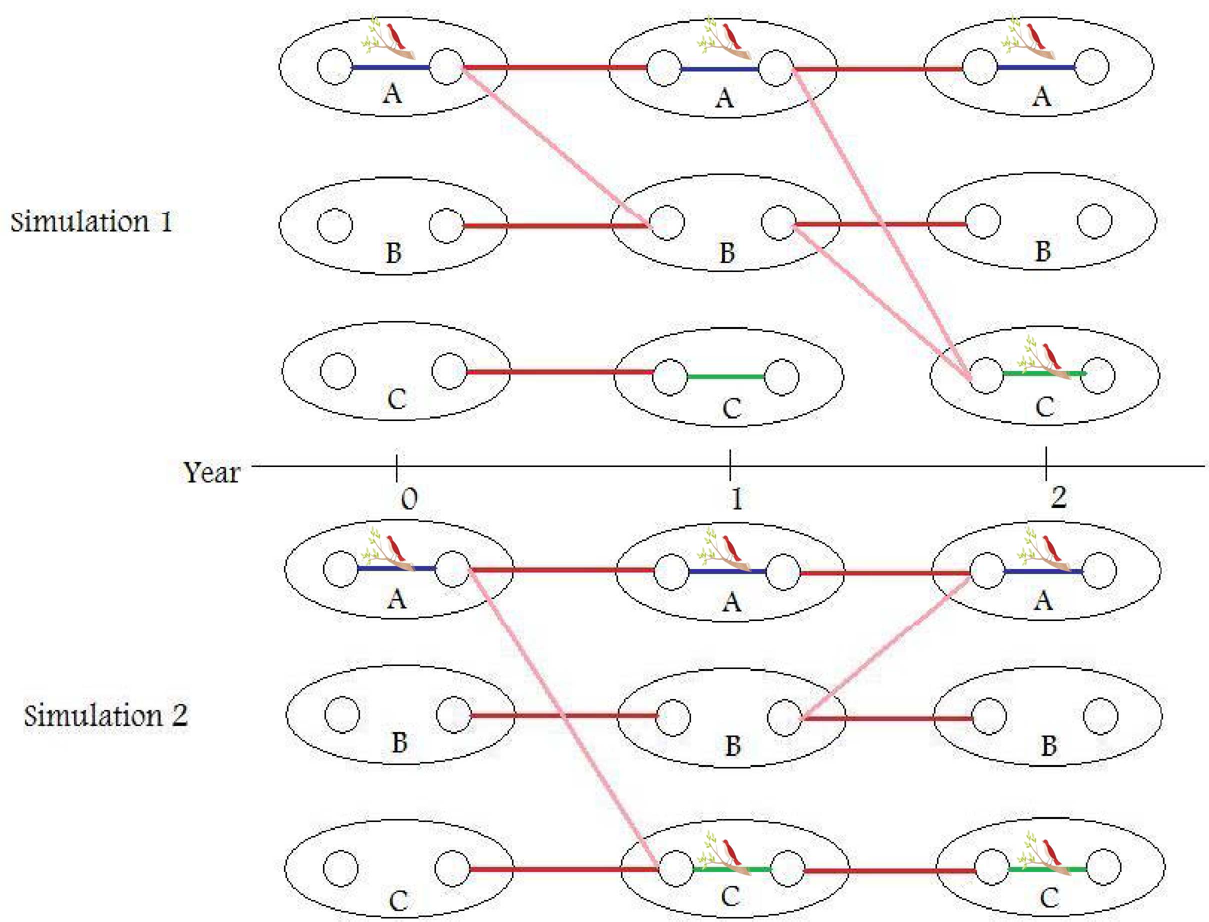
- If there are birds at B in year 1 in sim. 1, they will colonize C

Blue line represents territories that are already suitable (and occupied in this example)



- The reason for having two nodes represent each territory is to indicate whether or not it is suitable
- If suitable, then there will be a line, allowing the birds to inhabit the territory from one time step to the next

Green lines indicate which territory which we should purchase



- Start with only territory A occupied
- Now we want to decide which territory to purchase, B or C?
- What maximizes average number of occupied nodes at time 2?
- In simulation 2, the birds from A can never get to B. In both simulations we can get to C, so C is better

The Flow-Type IP Formulation

•

$$\text{maximize} \quad (1/K)\sum_{k=1}^{K}\sum_{i\in\mathcal{R}} x^{ik}(i,T)$$

$$\text{subject to} \quad \sum_{i\in\mathcal{R}}\sum_{t=1}^{T} b(i,t)y(i,t) \leq B,$$

Budget constraint

Purchase constraints

$$\sum_{t=1}^{T} y(i,t) \leq 1, \quad \forall i$$

Suitability constraints

$$x^{rk}(i,t) \leq \sum_{s=1}^{t} y(i,s), \quad \forall\, r,k,i,t$$

Colonization constraints

$$z^{rk}(i,j,t) \leq a^{k}(i,j,t), \quad \forall\, r,k,i,j,t$$

Flow constraints

$$\sum_{i\in\mathcal{R}} z^{rk}(i,j,t) = x^{rk}(j,t), \quad \forall\, r,k,j,t$$

$$x^{rk}(i,t) = \sum_{j\in\mathcal{R}} z^{rk}(i,j,t+1), \quad \forall\, r,k,i,t$$

$$x^{rk}(i,t),\ y(i,t),\ z^{rk}(i,j,t) \in \{0,1\} \qquad \forall\, r,k,i,j,t$$

LP Rounding of Flow-Type Formulation

A Greedy Approach to Rounding:

- Solve the LP relaxation – design variables are y(i,t)
- Find design variable with highest current value & set to 1
- Find all variables with value < .1 and set to 0
- Repeat

Why Do Something Different?

The budget constraint is a "fractional" knapsack problem in our setting, and this adds subtleties in maintaining feasible solutions

Results for a Toy Example

10 year time horizon, 5 simulations, 33 territories, cost of each is U[20,120] and decreases 10% a year

Budget	IP	LP Rounding	%optimal	Initial LP solution
300	6.6	5.8	87.9%	6.70
400	8.4	6.6	78.6%	8.53
500	10.2	10	98.0%	10.34
600	12	11.4	95.0%	12.13
700	13.6	13.2	97.1%	13.89
		Average	91.3%	

Just the Beginning (Not Even That)

Suppose that we want to solve a “real” input:

How many simulations/samples suffice?

Is it sufficient to just observe convergence (if it does)?

What do we do when the IP gets too big to solve?
(LP rounding results are quite promising)

Thank you!