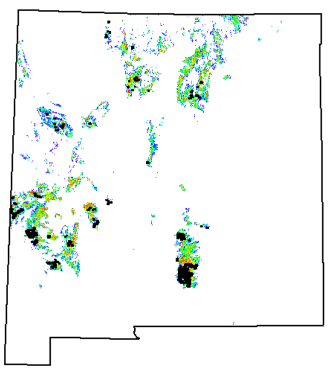
Site prioritization

Landscape Analysis & Management
Conservation GIS

Habitat models



Mahalanobis distances (cells most similar to the observed owl locations)

best, worst habitat

The Challenge

- To what extent can careful planning compensate for habitat loss?
  - What *kinds* of sites should be targeted?
  - How can habitat *configuration* mitigate for lack of *area*?

Scales of activity

- *Sites* (parcels, management units)
- Landscape (especially if habitats outside the sites are important)
- Reserve *networks* or portfolios (functional)
- National
- Transnational

Task: Reserve system design

- What should be the goal of a system of reserves (portfolio)?
- How to assemble a portfolio?
  - Logic and criteria?
  - Implementation (process)?

Structured decision-making

- What do we want?
  - Objectives hierarchy
- How can we meet our objectives?
  - Means-end models (path diagrams)
- How to decide among alternatives?
  - Structured decision framework

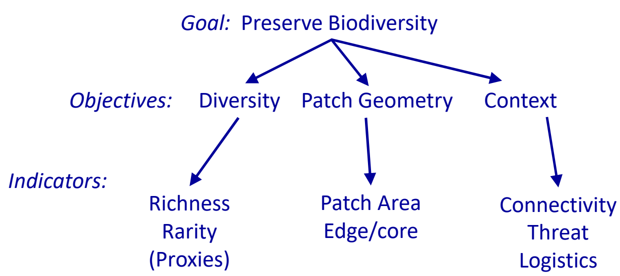
Objectives hierarchy

- Goal: the overall aim (warm, fuzzy)
- Objectives:
  - Categorical elements of the goal
- Indicators:
  - Empirical measures of objectives
  - Indicators have high signal/noise, are specific, free of observer bias, reproducible, … (and independent)

Objectives/criteria for conservation

- Ecological uniqueness (rarity, vulnerability, endemism)
  - Applied to species, communities, or habitats
- Habitat size, condition, geometry
- Viability (likelihood of persistence, given protection), especially connectivity
- Threats (regional pressures)
- Feasibility (acquisition, management)

Objectives hierarchy



Objectives and indicators

Diversity …

- Richness: tallies of Element Occurrences
- Rarity: richness weighted by rarity (S- or G-ranks)
- Modeled/predicted occurrences
- Proxies: habitat types, or environmental zipcodes (weighted by association with Element Occurrences)

Objectives and indicators

Geometry …

- Patch size (simple area)
- Core Area (area minus an edge buffer)
- Edge/area ratio or shape complexity
- Habitat quality (effective area, e.g., from maxent or other SDM)

Objectives and indicators

Spatial context …

- Connectivity (based on distance or LCPs)
- Threat (distance to, or amount of nearby development pressure)
- Logistics (e.g., proximity to other protected areas; ease of access for monitoring)

Objectives hierarchy: decisions

1. How to assign relative value to the indicators on each criterion?
   E.g., how much do we care about different numbers of species?
2. How to assign relative value across different criteria?
   E.g., how much do we care about species compared to connectivity?

Evaluating options: a simple method

- The data: each candidate site has a value for a set of indicators
- Approach:
  - Assign relative weights to each indicator per objective; weights must sum to 1.0 (or 100%)
  - Assign relative weights to each objective (weights must sum to 1.0 or 100% again)
  - Each candidate site has a score (rank them)
  - [fiddle with the weights to assess uncertainty]

The process

Prioritization Algorithms

The process: alternatives

1. Solve exactly (optimization)
2. Solve approximately (various options)
3. Support the decision (but do not solve)

Example application: minimum coverage

- Capture all species in the minimum total area (or number of sites, if all the same size)

Alternatives: (1) optimization

Linear programming methods

- Advantage:
  - finds the optimal solution (usually)
- Disadvantage:
  - very slow for large data sets
  - fails for some problem sets
  - "black box" method

Alternatives: (2) Simulated annealing

Algorithm:

1. begin with initial set of sites
2. choose a random site, in or out of portfolio
3. compute change in objective function if that site is added (or removed, if already in)
4. restrict 'acceptable' changes over time
5. repeat (2-4) until solution converges

Repeat (1-5) many times to yield a set of solutions

Alternatives: (2) Simulated annealing

- Advantages:
  - finds a good (maybe near optimal) answer
  - can assess very large data sets
  - can provide multiple solutions (alternative, near-optimal)
  - flexible about the objective function
- Disadvantages
  - (now) doesn't address all criteria (e.g., connectivity)

MARXAN

The objective function to be minimized:

- Σ costs, per site +
- Σ of penalties, per target, for not meeting conservation target(s) +
- Boundary weight*Σ boundary length (optional) +
- Penalty for not meeting cost threshold (optional)

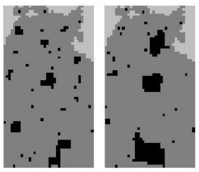
MARXAN

Tuning and options:

- "species protection factors" allow user to focus on target species
- "boundary length modifier" allows user to force focal adjacencies or the relative importance of boundary length
- also includes options for minimum parcel size and distances among parcels

MARXAN

Boundary length penalty



"Summed Irreplaceability"
(% of solutions that included it)

Ball & Possingham (MARXAN website)

Alternatives: (3) inform but don't solve

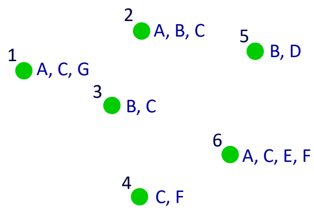
Greedy heuristic algorithm:

- An heuristic algorithm *learns* by itself
- A greedy algorithm seeks the most *parsimonious* solution, as quickly as possible

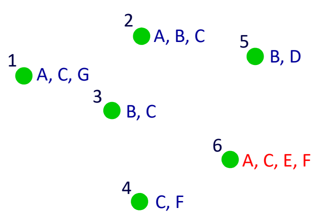
The greedy heuristic algorithm

1. Choose the best site (most species)
2. Tally, for each other site, the number of *new* species it contains
3. Choose the site with the most new species
4. Repeat (2&3) until all species are accounted

Greedy heuristic algorithm

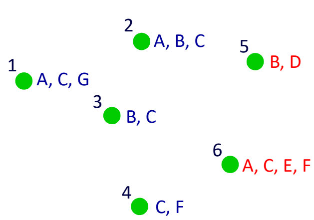


Greedy heuristic algorithm



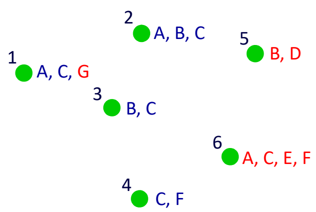
Sites in:
6

Greedy heuristic algorithm



Sites in:
6
5

Greedy heuristic algorithm



Sites in:
6
5
1

Greedy heuristics …

- Advantages:
  - intuitive
  - easy for small number of sites and targets (perhaps by inspection)
- Disadvantages:
  - hard or slow for large number of sites or targets (needs to be automated)
  - might not get the right answer (!)

Greedy heuristics …

Species	A	B	C
shrike	1	1	1
owl	1	1	0
g. sparrow	1	0	1
hawk	1	1	0
thrasher 1	1	0	0
grouse	1	0	1
s. sparrow	1	1	0
pelican	1	1	0
eagle	0	0	1
tern	0	1	0
Total S	8	7	4

Optimal solution:
sites B & C

Greedy solution:
sites A, B, & C
(A first)

Minimum representation: Extensions

- Examples thus far have considered only species richness, but …
  - all species are not equally compelling
  - we may want some redundancy
  - context and structure matter
  - connectivity might be important

Extensions: (a) Rarity

Rarity algorithm …

1. Assign a rarity weight or score to each target
2. Proceed with greedy algorithm

Solution: capture rare species first, but ultimately get them all

Extensions: (b) Redundancy

Redundancy algorithm …

1. Decide on a minimum number of occurrences per species
2. Proceed with greedy algorithm

Solution: capture multiple populations or occurrences (buffering under environmental variability)

Extensions: (c) Geometry

Reserve structure or context …

1. Assign a weight for edge length or similar adjacency metric
2. Proceed with greedy algorithm

Solution: capture compact reserves (and adjacent, with shared edges)

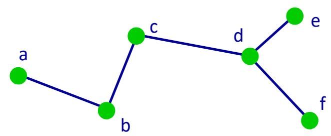
Extensions: (d) Connectivity

Connectivity algorithms …

1. Consider joining distance or functional definition
2. Proceed with greedy algorithm

Solution: build a connected network of reserves (e.g., a minimum spanning tree for the landscape)

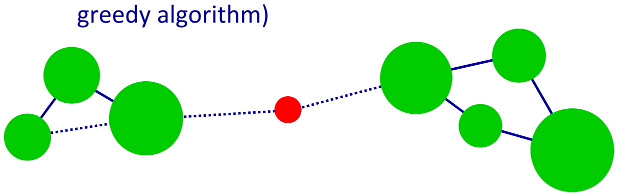
Extensions: (4) Connectivity



the minimum spanning tree for the graph: greedy algorithm on minimum edge length

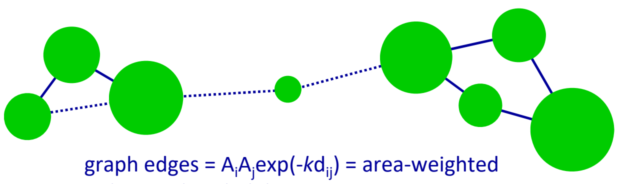
Connectivity (extensions)

- A stepping-stone has a *huge* effect on connectivity (but not obvious in simple greedy algorithm)



Connectivity (extensions)

- Define connectivity (total length of connected subgraph):



graph edges = $A_iA_j\exp(-kd_{ij})$ = area-weighted dispersal probabilities

Connectivity (extensions)

1. Define connectivity (dispersal-weighted connected area)
2. Compute total connectivity with sites already in the portfolio
3. Recompute connectivity with a site added (or removed)
4. Score the site on its *difference*

Extensions to the heuristic

- This approach of modeling the change expected on gain/loss of a site can be generalized to any model of conservation value, where value is contingent on the other sites in the system

Erosion potential: RUSLE

A (predicted soil loss, T/Ac/yr) =

R rainfall-runoff factor (local)

K soil erodability (local)

L slope length (local terrain)

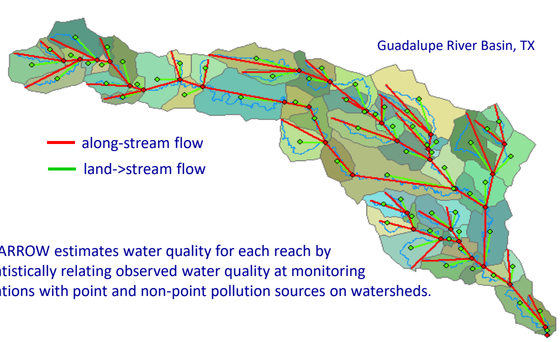
S slope gradient (local terrain)

C cover & management (vegetation)

P erosion control practices (land use)

"manage"

Watershed integrity (SPARROW)

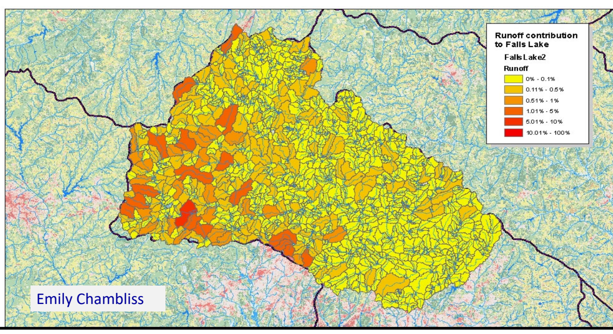


SPARROW estimates water quality for each reach by statistically relating observed water quality at monitoring stations with point and non-point pollution sources on watersheds.

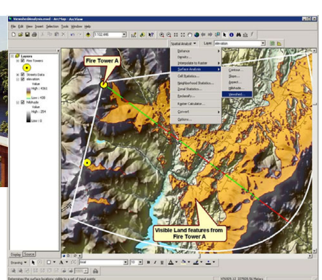
Watershed Analysis: Logic

- Compute N loadings to downsream target waterbody
- Loop over contributing catchments …
  - "Restore" (or "degrade") land cover
  - Rerun SPARROW
  - Recompute N loadings
- Rank catchments in terms of delta-N

Watershed Analysis



Recreation/aesthetics (viewsheds)



Viewshed analysis:
line-of-sight visibility

Greedy heuristics

The logic is reasonably robust and generalizable:

- Simple implementation captures many (most?) cases for site-level attributes
- Forward/backward "modeled delta" implementation covers cases where the network context matters

Site prioritization: alternatives

- Exact solution: optimization
  - nobody really does this in conservation
- Approximate solutions
  - Simulated annealing (MARXAN)
    - Relative penalties on targets and boundary length provide weights to the objectives
    - Some other options provide more creative solutions
- Decision support only (without solution)

Decision support: PORTFOLIO

PORTFOLIO basics …

- Greedy heuristics on:
  - patch area and geometry (core area or habitat quality)
  - diversity: richness, rarity, or proxies
  - connectivity (area or core)
- Allows user to explore options; user makes any decision

PORTFOLIO: Data requirements

- Site attributes
  - Area, core area (and some optional attributes)
- Connectivity
  - Between-site distances as a matrix or edge list
- Species attributes
  - Rarity score (weight)
- Census
  - which species occur on each site

PORTFOLIO: Processing

- Read input data
- Compute weighted edges: $W_{ij} = A_i A_j exp(kD_{ij})$ for connectivity
- At each iteration:
  - compute change in each criterion, for each site
  - present the choices to user
  - user chooses, and program updates

<PORTFOLIO>

Conclusions

- You have the geospatial tools to build the data sets needed to do site prioritization
  - simple rankings based on single attributes
  - rankings based on weighted averages of indicators for multiple attributes
  - MARXAN (approximate solution)
  - PORTFOLIO (user-selected but self-documenting solution)