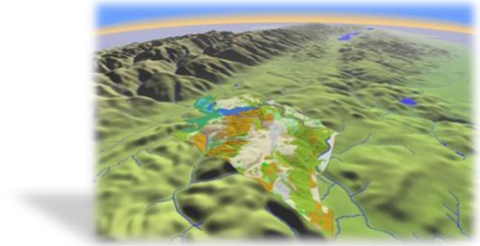
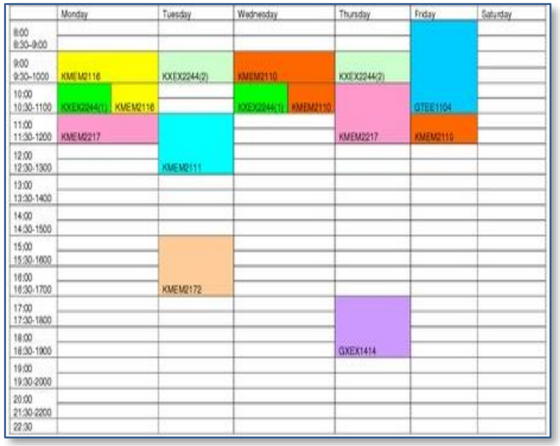
NICHOLAS SCHOOL OF THE ENVIRONMENT AND EARTH SCIENCES
DUKE UNIVERSITY



ENVIRON 761: Prioritization, Optimization & Conservation Area Selection

Instructor: John Fay

Choosing classes:



- Can't take them all…

- Select classes that favor *criteria*
  - Topic (applicable or not-related)
  - Interest (captivating or boring)
  - Logistics (schedule conflicts)
  - Requirements (GIS certificate?)

- *What is the best combination of classes?*
  - Depends on your preferences, your options, your situation

How do you decide?

Choosing *conservation sites*:

- Can't protect them all…

- Select sites that favor *criteria*

Geometry	(core area, shape index)
Threats	(unthreatened, desperate for protection)
Connectivity	(betweenness centrality, corridor potential)
Biodiversity	(richness, rarity)
Cost	($$$)

- *What is the best portfolio of potential sites?*
  - Depends on mission, options, situation

How do you decide?

Prioritization & Reserve Design

General question…

How can we optimize the selection of sites for protection to meet multiple conservation objectives?

Multi-Attribute Decision Analysis

Methods for making decisions when:

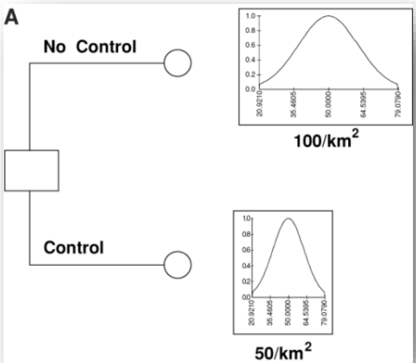
- *Outcome is uncertain*
- *Multiple [conflicting] objectives*
- *Multiple [conflicting] stakeholder interests*

Approaches include:

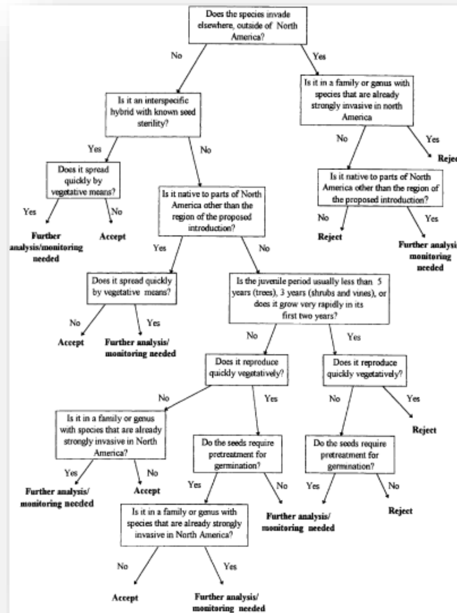
- *Articulating goals*
- *Expressing priorities among these goals*
- *Providing framework for communication*

[*Maguire 2004*]

Multi-Attribute Decision Analysis



Probability models



Decision trees

Best management for feral pigs

Native flora and fauna
Forest birds
Diversity
Density
Disease incidence
Understory plants
Density
Diversity

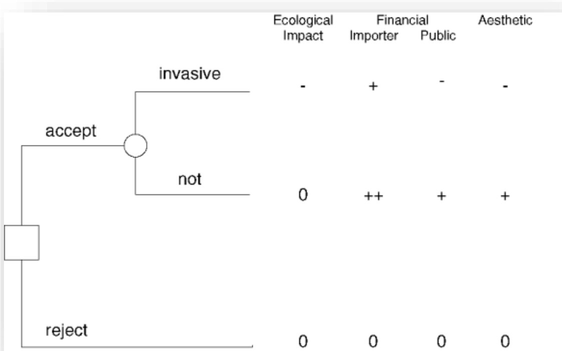
Hawaiian culture
Traditional medicine/food
Plant diversity
Plant density
Life event celebrations
Pig density
Respectful resource use
Pigs killed but not eaten

Safety of nontargets
Humans
Mortality
Injury
Pets/hunting dogs
Nontarget fauna

Humane methods
Minimum mortality
Pigs killed
Minimum suffering
Time to death

Cost-effective
$ to maintain low pig density

Objectives hierarchy



Values models

Maguire 2004

Landscape Prioritization: Attributes

Patch/HUC Geometry

Patch/HUC Threat

Patch/HUC Connectivity

Patch/HUC Biodiversity

$

Attributes of PatchAttributes

OID	HABPATCH	COST	AREA	SHAPEINDEX	PCTCORE	WTDTHREAT	HUMANKDENS	EDIST2ROAD	CDIST2DEV	BETWEENNES	RICHNESS	EVENNESS
0	1	2	194400	245	0	20	0	3964	3977	0	3	83
1	2	162	0	0	16	20	1	1273	1236	34	7	153
2	3	46	0	0	12	20	3	2124	2232	2	4	91
3	4	120	1158300	289	9	40	3	0	0	18	7	147
4	5	15	170100	229	0	34	1	1018	988	46	3	81
5	6	19	0	0	0	40	4	0	0	58	4	122
6	7	13	1911600	348	16	35	6	0	0	56	6	118
7	8	158	1393200	183	41	20	0	2072	2233	6	6	126
8	9	34	0	0	0	40	3	450	458	54	3	82
9	10	198	3061800	249	50	22	7	764	741	18	4	102
10	11	6	178200	213	0	20	2	5022	5048	2	3	79
11	12	44	907200	250	19	40	4	0	0	18	4	122
12	13	35	0	0	0	25	1	1080	1058	8	3	88
13	14	67	688500	347	1	36	0	764	844	6	3	64
14	15	37	882900	206	22	33	0	525	555	4	5	82
15	16	30	348300	206	0	20	0	0	11903	0	2	11
16	17	2	259200	221	0	40	0	180	377	6	3	45
17	18	5	162000	190	0	40	0	180	215	4	1	0
18	19	36	469800	217	7	20	5	4122	4144	18	3	77
19	20	179	1539000	247	24	40	2	0	0	6	5	101
20	21	6	170100	229	0	15	1	6004	6817	0	1	0
21	22	4	218700	231	0	10	0	6165	12720	8	3	68
22	23	23	0	0	0	10	0	3610	15776	8	2	35
23	24	1	0	0	0	10	0	5043	13706	14	3	74
24	25	87	4746600	613	11	32	0	0	0	26	4	73
25	26	18	283500	177	0	10	0	3097	14893	10	3	96
26	27	1	251100	153	3	10	0	4075	14341	8	3	59
27	28	38	348300	145	7	20	1	450	3976	22	4	115

Record: 1 Show: All Selected Records (0 out of 671 Selected) Options

Landscape Prioritization: Sorting

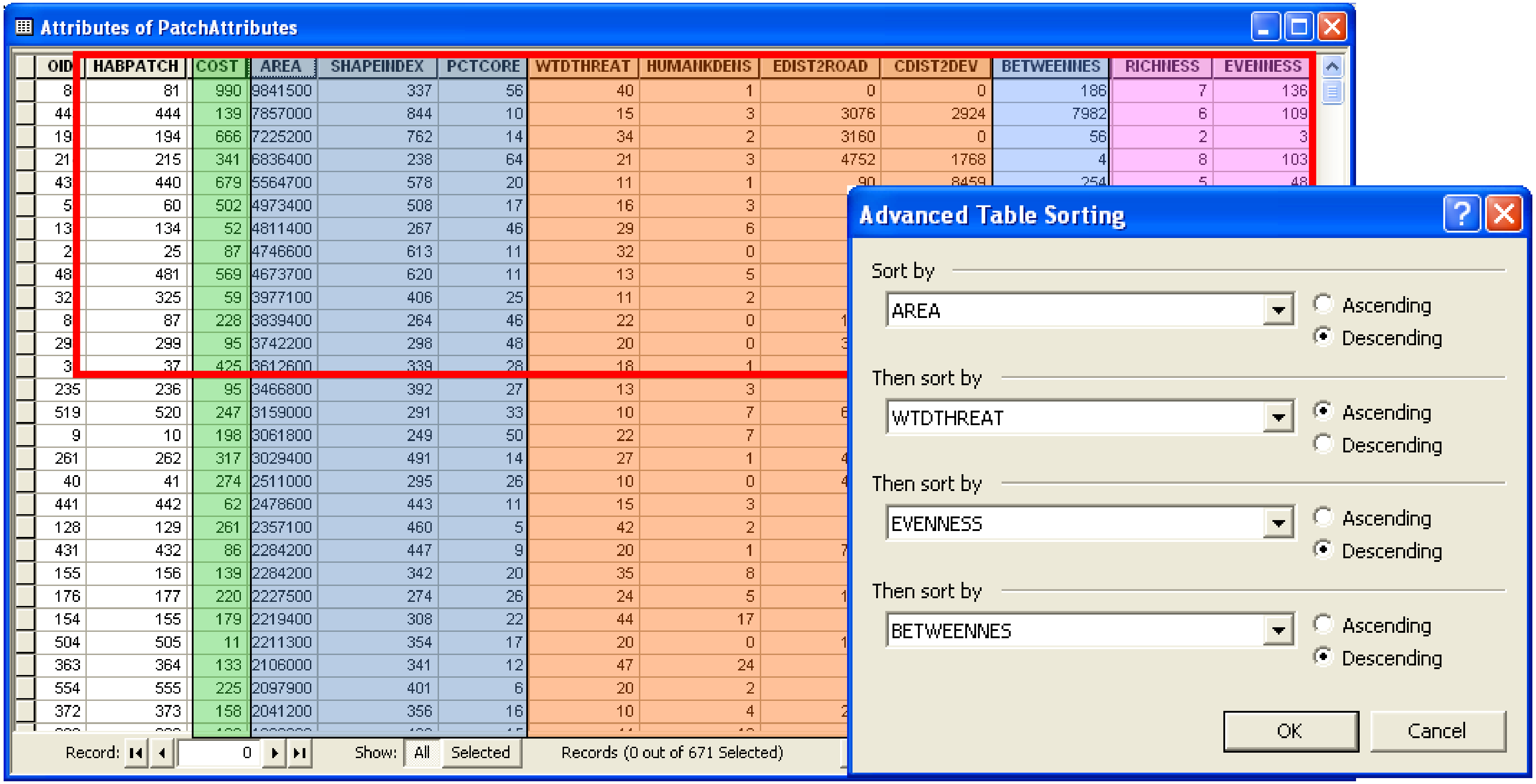
Patch/HUC Geometry

Patch/HUC Threat

Patch/HUC Connectivity

Patch /HUC Biodiversity

$



Landscape Prioritization: Sorting

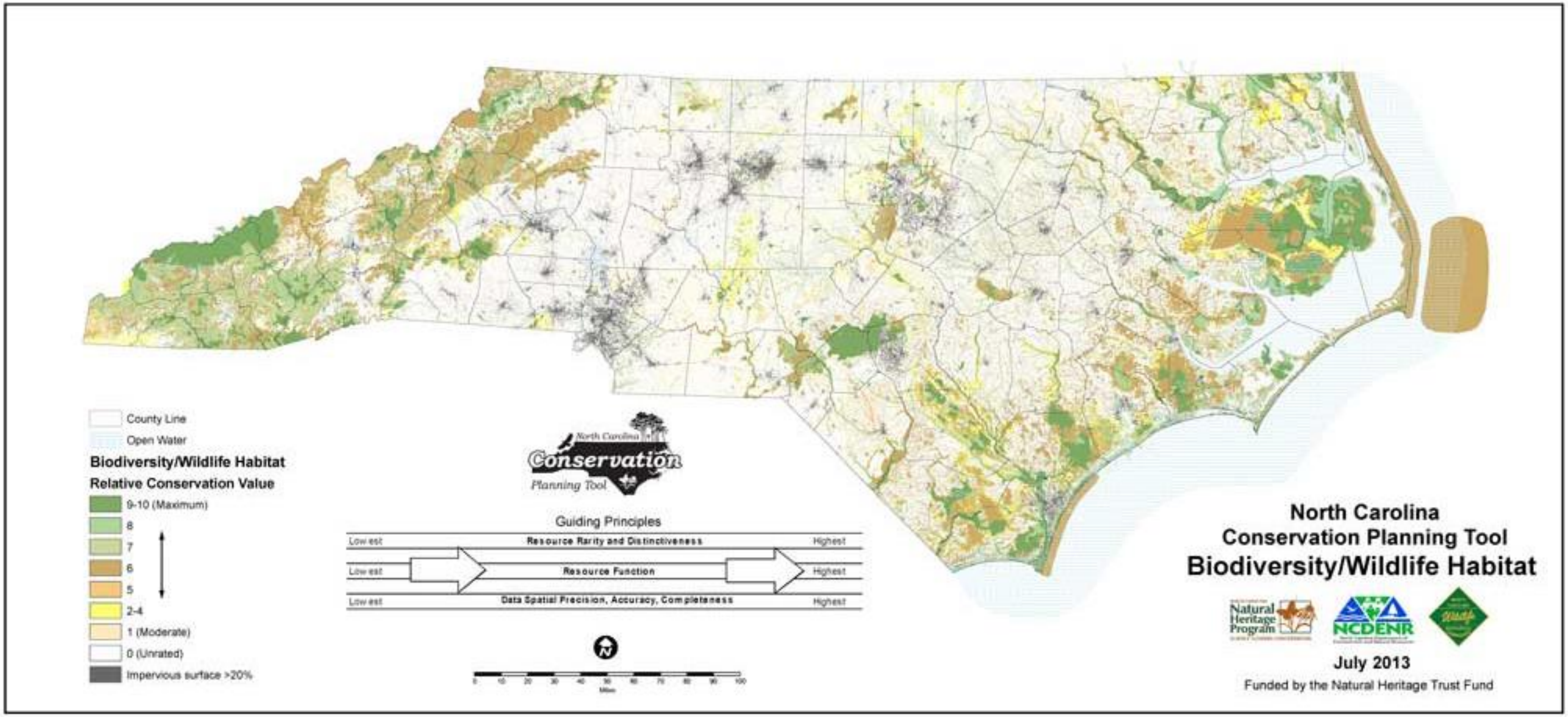
- Useful when objectives are clear and attributes accurately reflect those objectives:
  - *Objective*:
    pronghorn viability
  - *Approach*:
    protect as much unthreatened habitat area as possible, (given the budget of the program)
  - *Method:*
    - Sort by Area, then by Wtd. Threat
    - Select patches until budget is spent

Landscape Prioritization: Sorting

- Sort on most important attribute
  - Keep top *X*%..
  - Determine cost cuts still required
- Sort on 2^{nd} most important attribute
  - Keep top *X*%...
  - Determine cost cuts…

How much more important is attribute 1 than 2, and so on??

NC Conservation Plan



http://portal.ncdenr.org/web/cpt/cpt-report

NC Conservation Planning tool

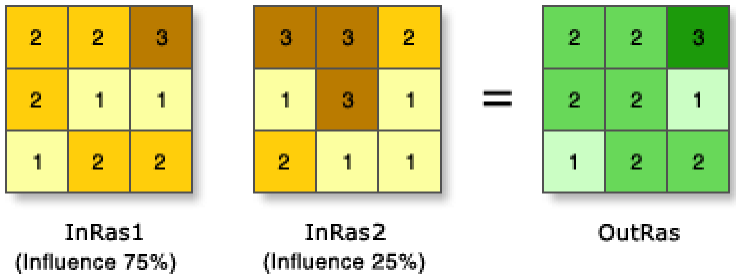
Appendix C: Relative Criteria Ranking for the BIODIVERSITY / WILDLIFE HABITAT ASSESSMENT

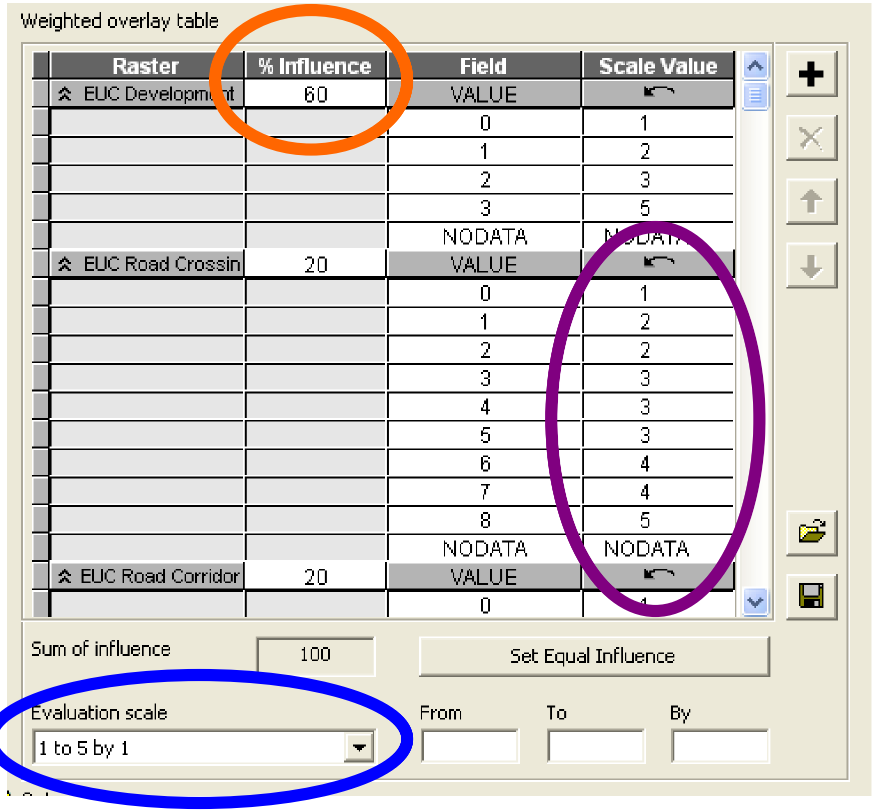
Lowest	**Resource Rarity and Distinctiveness**	Highest
Lowest	**Resource Function**	Highest
Lowest	**Data Spatial Precision, Accuracy, Completeness**	Highest

Moderate								**Maximum**	
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
	CREWS-Beneficial			NWI Wetlands	CREWS - Substantial	CREWS Exceptional			
LHI Guild (1)									LHI Guild (10)
							HQW (select)	ORW (select)	
							Oyster Sanctuaries		
					Submerged Aquatic Vegetation		Fish Nursery Areas		
				Shellfish Closed - Shellbottom		Hard Bottom	Shellfish Open - Shellbottom		
						DWQ Stream Bioclass Good		DWQ Stream Bioclass Excellent	
							Anadromous Fish Spawning	Wild Trout Waters	
All Streams		Streams in Priority Watersheds				Stream Buffer tribs. to T&E			
					Important Bird Areas				
					SNHA – General Rating		SNHA – High and Moderate Ratings		SNHA – Outstanding and Very High Ratings
			EO - Other	EO - High					

Page 6 of Chapter 4:
Maximum Ranking Approach

Landscape Prioritization: Weighted Overlay

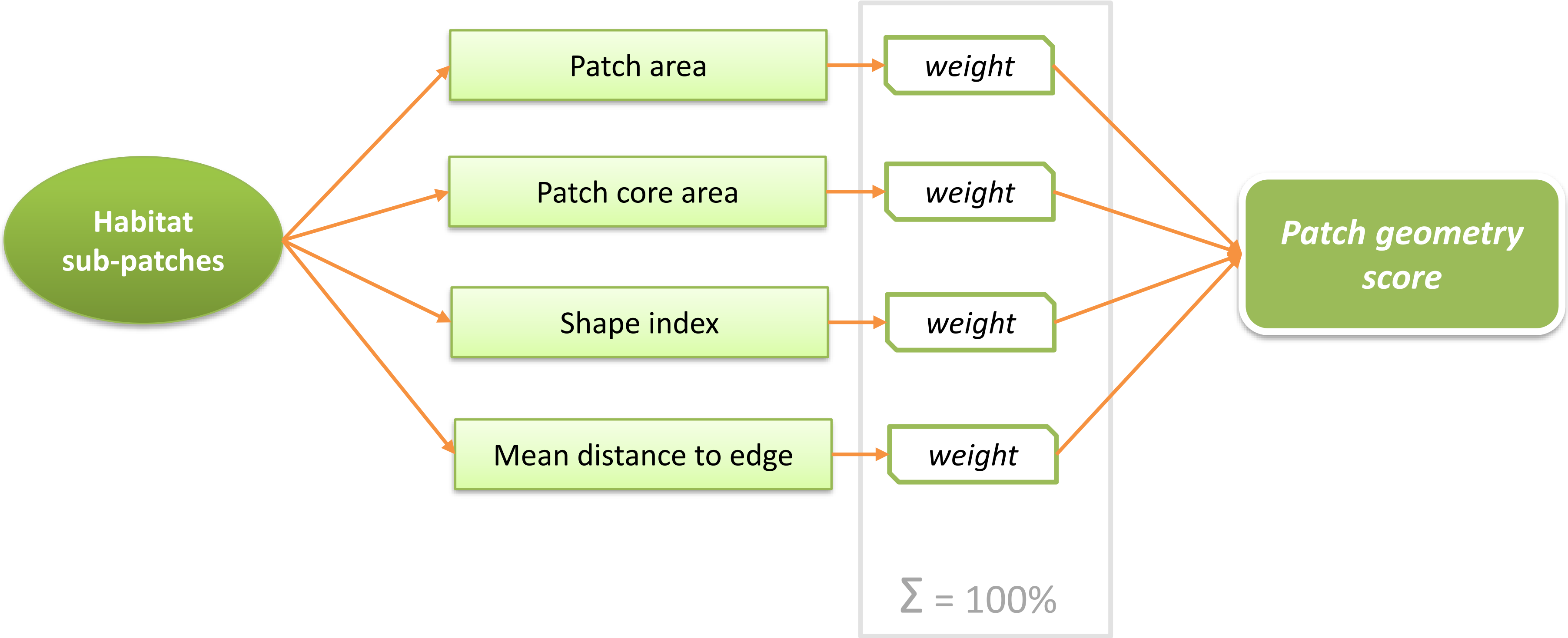




- Number of output classes
- % influence of each input
  - Development = 3x others
- Scale values
  - Extreme impact = 5
  - Minimal impact = 1

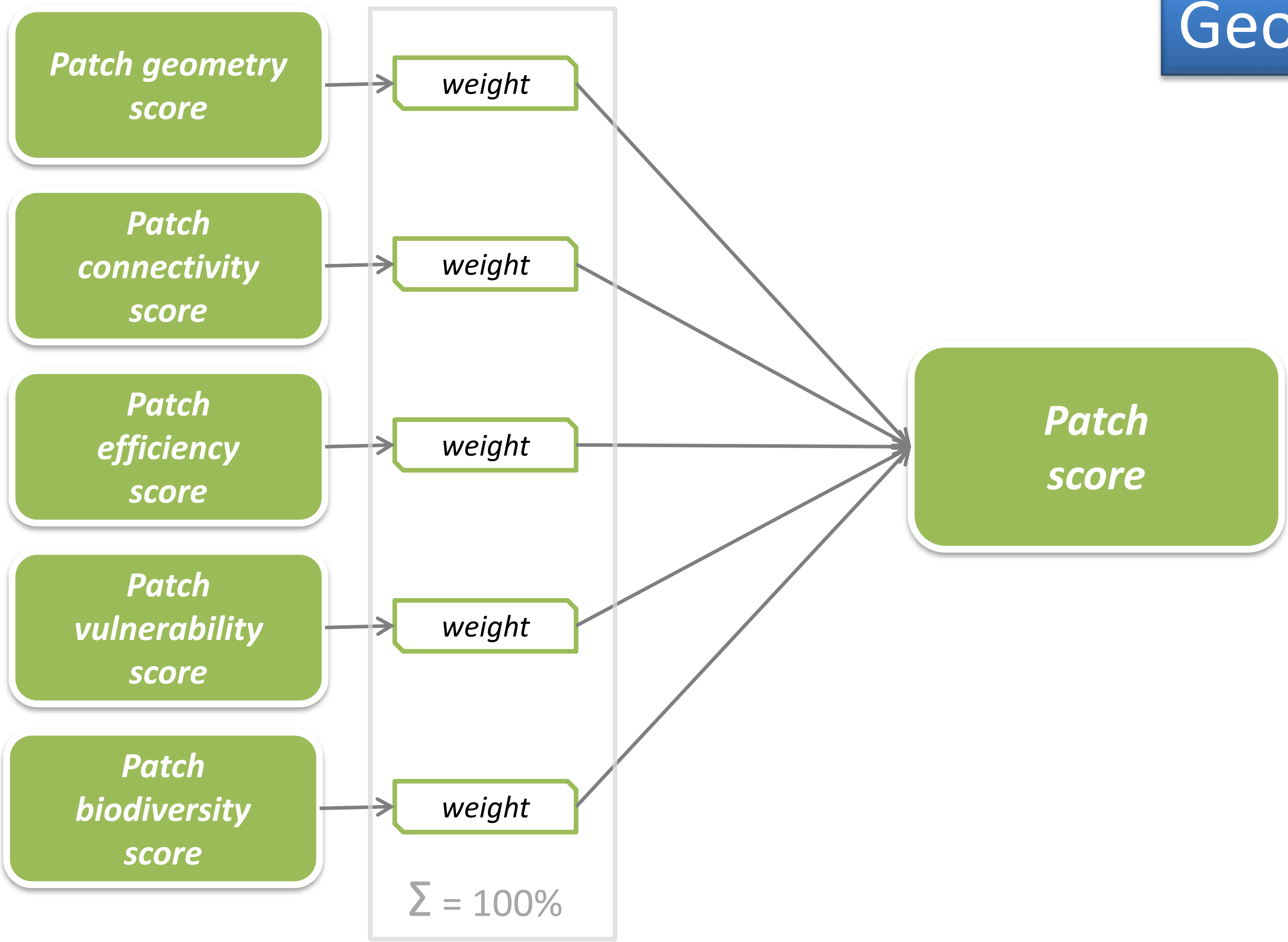
Landscape Prioritization: Weighted Overlay

GeoHAT

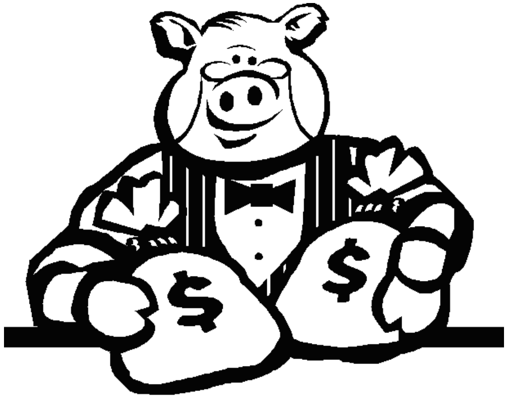


Multi-attribute synthesis

GeoHAT

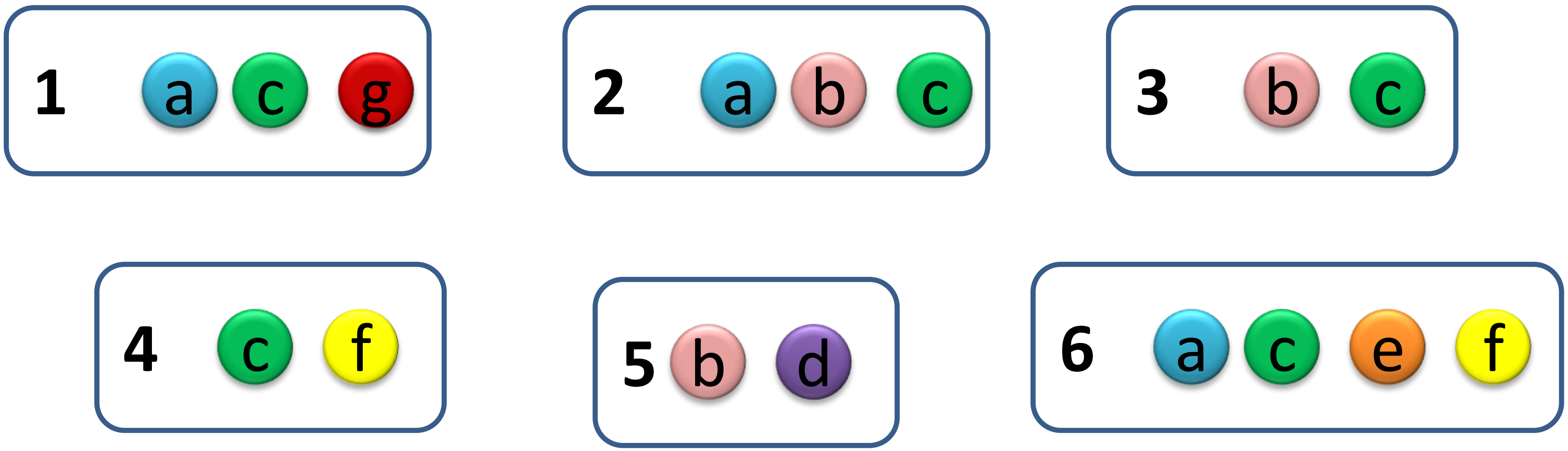


Selection algorithms: *Greedy*



- Sorting (on raw data or weighted overlay data) is a purely ***greedy*** approach:
  - Searches for (via sorting) and takes the best *X* number of patches until criteria is met
  - It does not necessarily arrive at most parsimonious solution!

Selection algorithms: *Greedy*



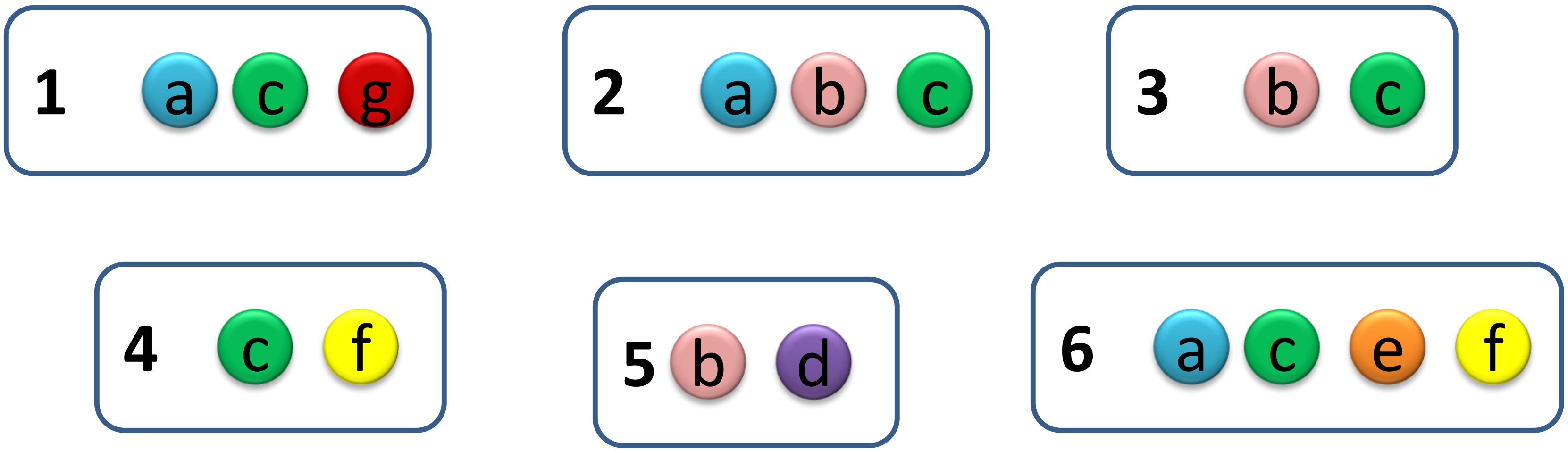
Pick #1: *Site 6*

Pick #2: *Site 1* or *2*

Pick #3: *Site 1* or *2*

Choose the *richest* remaining site…

Selection algorithms: *Greedy Heuristic*



Pick #1: *Site 6*

Pick #2: *Site 5*

Pick #3: *Site 1*

Choose the best remaining site
that <u>complements</u> chosen sites

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
  - Have to examine each alternative; sequence matters…
- may 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
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)

Selection Algorithms: *Greedy*

Greedy

Attributes of PatchAttributes

OID	HABPATCH	COST	AREA	SHAP
80	81	990	9841500	
443	444	139	7857000	
193	194	666	7225200	
214	215	341	6836400	
439	440	679	5564700	
59	60	502	4973400	
133	134	52	4811400	
24	25	87	4746600	
480	481	569	4673700	
324	325	59	3977100	
86	87	228	3839400	
298	299	95	3742200	
36	37	425	3612600	
235	236	95	3466800	
519	520	247	3159000	
9	10	198	3061800	
261	262	317	3029400	
40	41	274	2511000	
441	442	62	2478600	
128	129	261	2357100	
431	432	86	2284200	

“The top X patches are selected”

Each patch selection is made independent of what is found in other patches.

Selection Algorithms: *Greedy* ***Heuristic***

Greedy Heuristic

Attributes of PatchAttributes

OID	HABPATCH	COST	AREA	SHAP
80	81	990	9841500	
443	444	139	7857000	
193	194	666	7225200	
214	215	341	6836400	
439	440	679	5564700	
59	60	502	4973400	
133	134	52	4811400	
24	25	87	4746600	
480	481	569	4673700	
324	325	59	3977100	
86	87	228	3839400	
298	299	95	3742200	
36	37	425	3612600	
235	236	95	3466800	
519	520	247	3159000	
9	10	198	3061800	
261	262	317	3029400	
40	41	274	2511000	
441	442	62	2478600	
128	129	261	2357100	
431	432	86	2284200	

Heuristic = self learning

Patch selections are made to complement existing patch selections

Patch #194 complements patch #81 better than patch #444 …

81
a b c d
x y z

444
a b d
x y

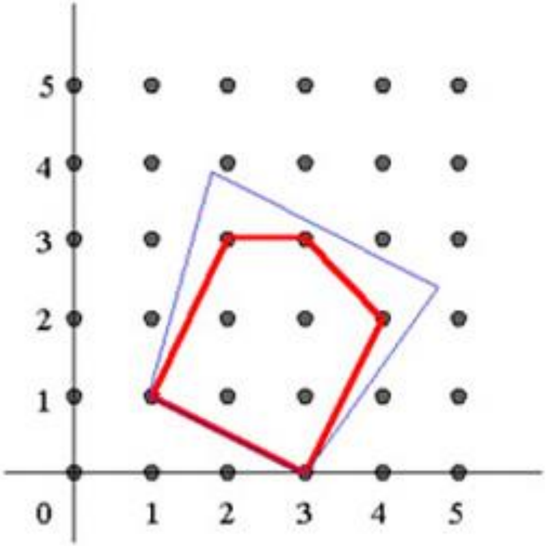
194
p q r s

Greedy heuristics & linear programming

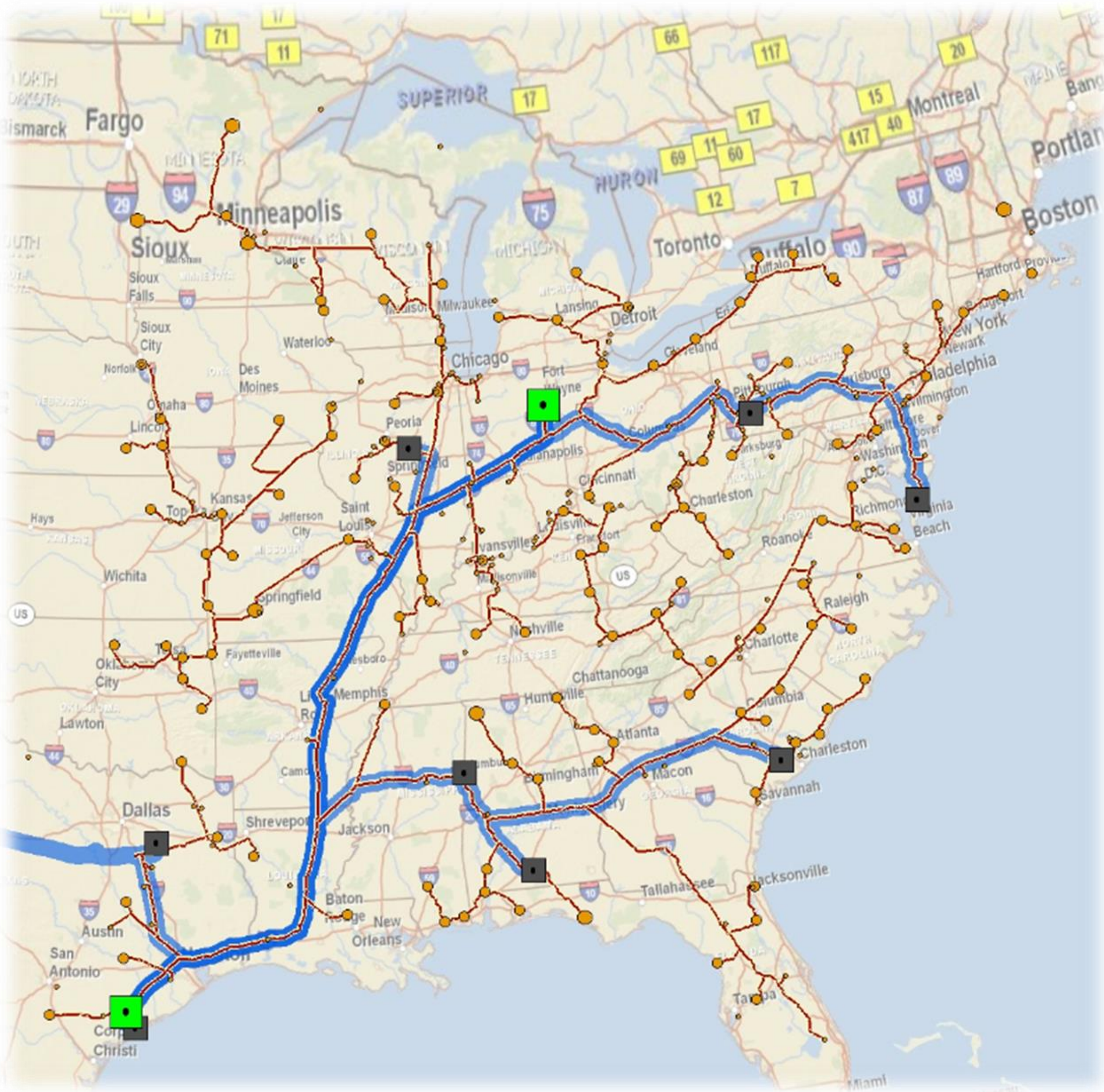
- ***Integer programming*** **and** ***Integer goal programming*****:** Integer programming are linear programming techniques derived from operations research that iterate through possible combinations of variable until the optimal result (as defined in the decision rules) is achieved.

- *Pluses:* Offers a ***comparable***, optimal solution

- *Minuses:* can be very CPU intensive and ***potentially impractical*** to employ



Linear programming, an example



Find optimal pipeline configuration to line coal-fired power plants with carbon sequestration sites…

- *Where to start routes?*
- *Order of linking sites to pipeline?*
- *Redundancy?*
- *New sites?*

Selection Algorithms: Integer programming

Examine all unique combinations of planning units and select the one combination that's the most parsimonious.

The top ranked planning unit may not be a part of that; may lead to "local minimum".

2^{144} or $> 10^{43}$ unique combinations!!!

*Solution: **Simulated Annealing** - a compromise to finding the optimal combination of patches…*

Operations Research

Maximum Covering Location Problems **(MCLP)**:

"Find fewest facilities that cover the most demand"

The MCLP method substitutes the concept of species representation for covered in the algorithm.

This integer program method then *converges* on a single solution that maximizes representation for the number of site (area) prescribed.

Problem P_1: $$\min\left\{\sum_{i=1}^{n} x_j\right\}$$

s.t. $$\sum_{j=1}^{n} \phi(d_{ij})x_j \geqslant Ty_i \quad i \in N$$

$$\sum_{i=1}^{n} y_i w_i \geqslant \alpha W$$

$$x_j \in \{0,1\}; \quad y_i \in \{0,1\} \quad i,j \in N$$

Marxan : Marine Optimization

MARXAN / SPEXAN / SITES

Optimization programs designed to be used for spatially aggregating habitat patches for optimal coverage.

MARXAN is the marine version of this software, but has become the standard for terrestrial applications too…

https://marxansolutions.org/

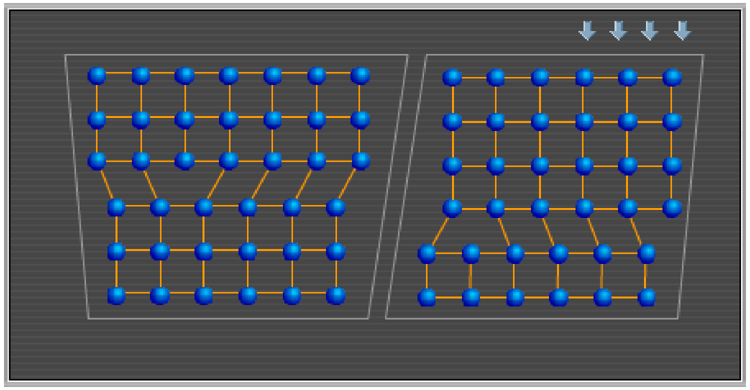
Marxan & "Simulated Annealing"

Uses "simulated annealing" to find an optimal combination of planning units (patches) that meet a specified objective function.

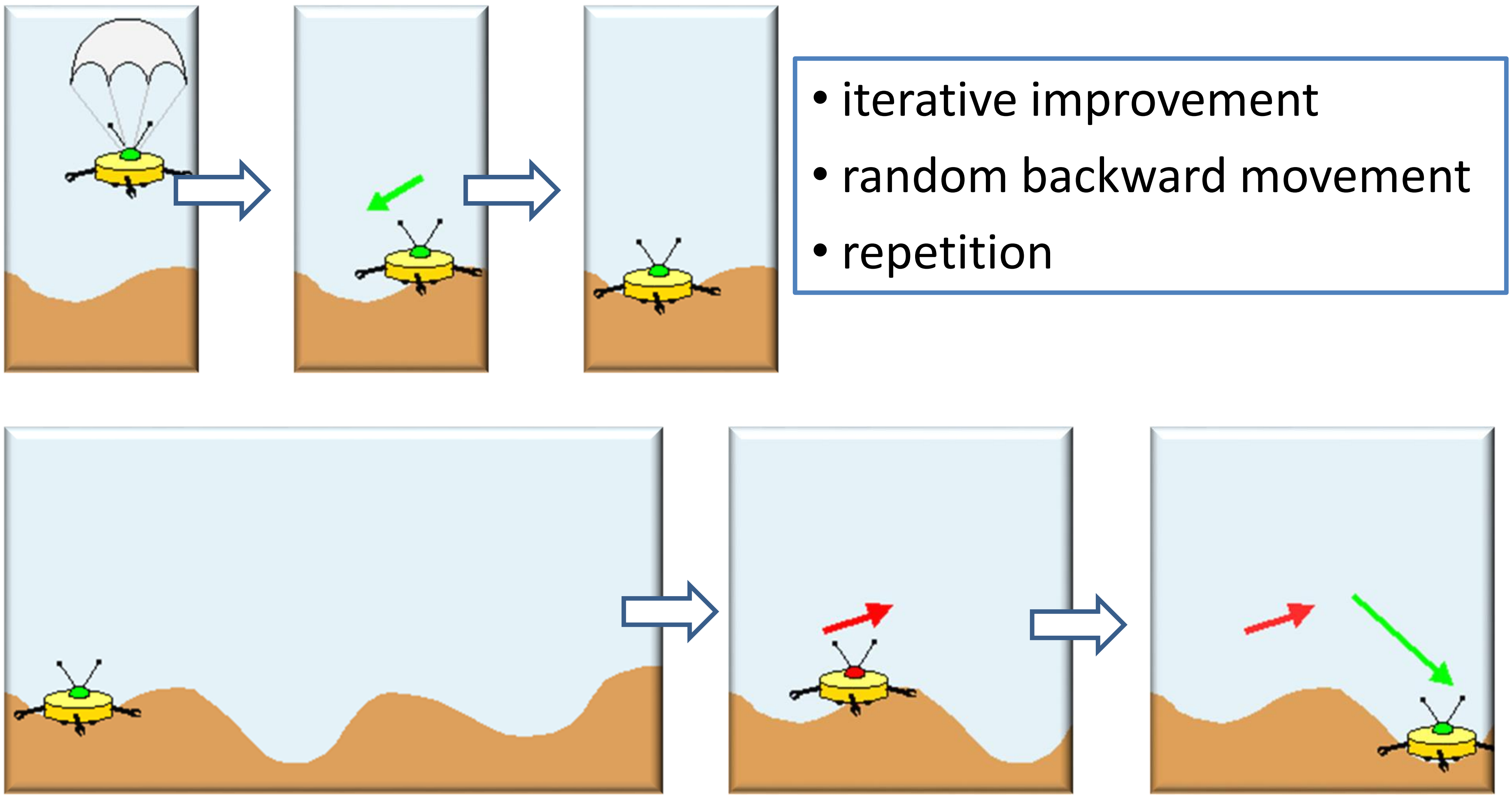
1. Begins with seed sites(random or set).
2. Randomly selects an additional site
3. Computes *gain* of adding that site towards objective function
4. Adds the site to solution set *if doing so exceeds a set threshold.*
5. The threshold gets higher as more sites are added

"Annealing"??

- From metallurgy…



Simulated annealing



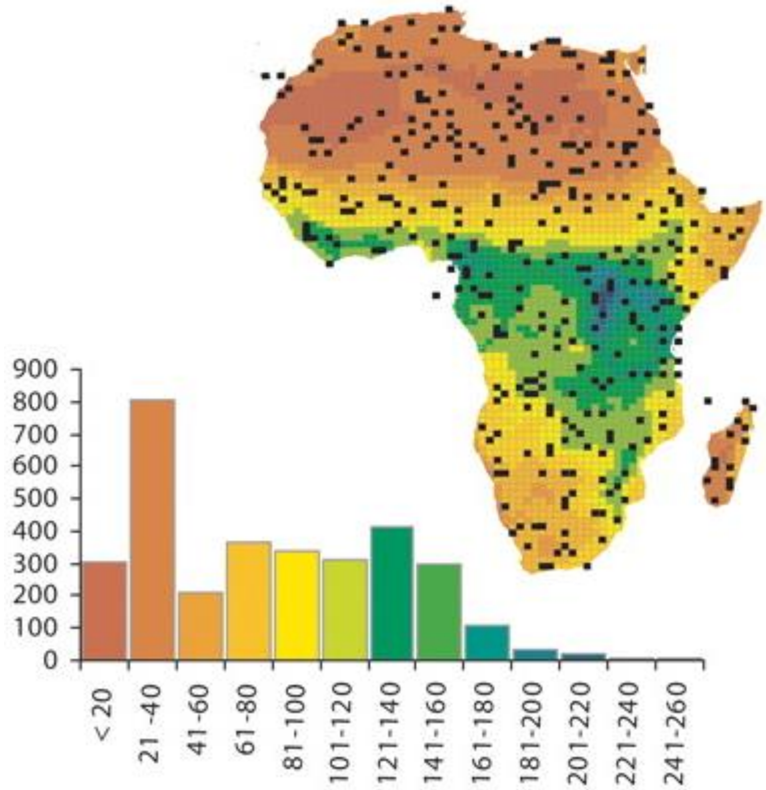
- iterative improvement
- random backward movement
- repetition

Aliens like low elevations...

Simulated annealing

- Advantages:
  - finds a good answer
  - can assess very large data sets
  - can provide multiple solutions (alternative, near-optimal)

- Disadvantages
  - less intuitive, less accessible



http://www.sciencemag.org/content/309/5734/603.full.pdf

Marxan

Purpose:

- To identify suites of planning units (i.e. HUCs) that maximizes conservation goals with the least amount of cost

Required inputs:

- List of **conservation features** to be included
- List of **planning units** (w/cost and other attributes)
- **Cross-list** of conservation features within each patch
- Specified conservation goal & other run time settings

Conservation Features file (spec.txt)

- Conservation features are features (e.g. species) you want represented in your final portfolio of patches

- The conservation feature file includes the type and the desired amount of each feature, as well as a penalty incurred if its not adequately represented.

- **id** = *feature identification number*
- **target** = *representation target (area, # occurrences)*
- **spf** = *species penalty factor*
- **name** = *feature name*

spec.txt

id	target	spf	name
5	10.2	10	Rocky_Mountain_Cliff_and_Canyon
9	0.4	10	Colorado_Plateau_Mixed_Bedrock_Canyon_and_Tableland
11	0.8	10	Inter-Mountain_Basins_Active_and_Stabilized_Dune
12	93.4	10	Inter-Mountain_Basins_Volcanic_Rock_and_Cinder_Land
14	123.2	10	Inter-Mountain_Basins_Playa

Conservation Features file (spec.txt)

- The Conservation Features file can be created manually by listing the feature types and the target representation.

- Or, you can set the target to be a proportion of existing conservation features within the planning area directly from the raster or feature attribute tables…

Creating spec.txt file from GAP Cover types:

1. Extract GAP data found within patch areas
2. Calculate targets as a proportion of cell count (e.g. 30%)
3. Add penalty factors (or use constant values)
4. Convert table to text file

NOTE: Be sure to convert spaces and commas to underscores!

Planning Unit file (pu.txt)

- Planning units are the discrete land units making up the conservation portfolio; a unit is either included or not.
  - Can be a patch, parcel, catchment, hexagon, etc.

- The planning unit file is a list of list of each planning unit id, the cost incurred by adding it to the portfolio, and a status value indicating whether the unit/patch should be “locked in” to (or out of) the solution.
  - The cost can be in actual dollars or be anything you want minimized (e.g. opportunity cost)

pu.txt

id	cost	status
1	2	0
2	162	0
3	46	0
4	120	1
5	15	0
6	19	0
7	13	0

Planning Unit file (pu.txt)

- The planning unit file can be created directly from the attribute table of the planning unit features….
  - ID = feature ID
  - Cost = a cost-related attribute
  - Status can be altered manually or all set to ‘0’.

Attributes of PatchAttribut

OID	HABPATCH	COST	AI
0	1	2	19
1	2	162	
2	3	46	
3	4	120	115
4	5	15	17
5	6	19	
6	7	13	191
7	8	158	139
8	9	34	
9	10	198	306
10	11	6	17

→

id	cost	status
1	2	0
2	162	0
3	46	0
4	120	1
5	15	0
6	19	0
7	13	0
8	158	0
9	34	1
10	198	0

Planning Unit v. Cons. Feature file (puvsp.txt)

- The planning unit vs. conservation feature file is simply a cross listing of what species are found in which planning unit - and how much.
- It can be created by *spatially combining* the planning unit (i.e. patches) with the conservation feature (i.e. GAP cover) datasets

Attributes of patchgap

Rowid	VALUE *	COUNT	HABPATCH	AZ_LANDCOVER
0	1	1	1	34
1	2	13	1	36
2	3	10	1	67
3	4	82	2	36
4	5	129	2	67
5	6	33	2	76
6	7	151	2	34
7	8	3	2	71
8	9	124	3	67
9	10	93	3	36

→

puvsp.txt

amount	pu	species
1	1	34
13	1	36
10	1	67
82	2	36
129	2	67
33	2	76
151	2	34
3	2	71
124	3	67

Optional: Boundary Length Modifier (blm.txt)

- Used to determine how much emphasis should be placed on minimizing the overall reserve system boundary length (amount of edge)
- Consists of a list of shared edge between planning units. Favoring two planning units that share an edge will reduce the overall amount of edge in the final reserve design.
- By not including an attached planning unit, the amount of boundary that would have been included is added as a cost...

id1	id2	boundary
1	2	20
1	3	0
1	4	10
2	3	3
2	4	7
3	5	3
4	5	9

Marxan inputs

spec.txt

id	target	spf	name
5	10.2	1000	Rocky_Mountain_Cliff_and_Canyon
9	0.4	1000	Colorado_Plateau_Mixed_Bedrock_Canyon_and_Tableland
11	0.8	1000	Inter-Mountain_Basins_Active_and_Stabilized_Dune
12	93.4	1000	Inter-Mountain_Basins_Volcanic_Rock_and_Cinder_Land
14	123.2	1000	Inter-Mountain_Basins_Playa

pu.txt

id	cost	status
1	2	0
2	162	0
3	46	0
4	120	1
5	15	0
6	19	0
7	13	0

bound.txt

id1	id2	boundary
1	2	20
1	3	0
1	4	10
2	3	3
2	4	7
3	5	3
4	5	9

puvsp.txt

amount	pu	species
1	1	34
13	1	36
10	1	67
82	2	36
129	2	67
33	2	76
151	2	34
3	2	71
124	3	67

How are these "raw materials" used to find "the optimal subset of patches" for a reserve?

MARXAN Objective functions

- The *objective function* is what gives one alternative a "better" score than another.

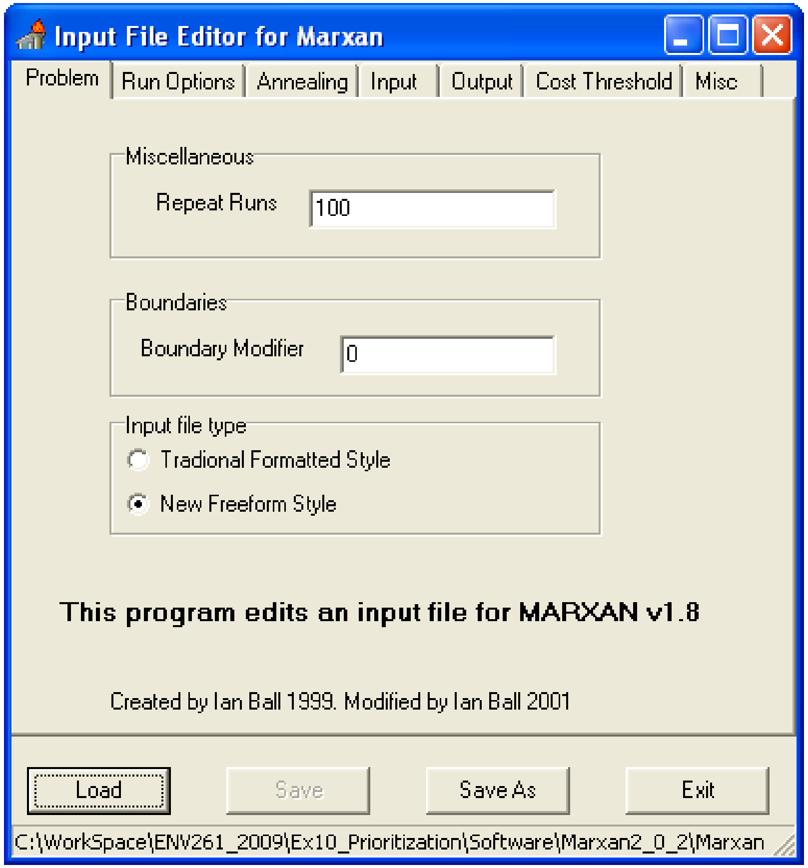
$$\underbrace{\sum_{PUs} Cost}_{1} + \underbrace{BLM \sum_{PUs} Boundary}_{3} + \underbrace{\sum_{ConValue} SPF \times Penalty}_{2} + \underbrace{CostThresholdPenalty(t)}_{4}$$

1. The total cost of the reserve network (required)
2. The penalty for not adequately representing conservation features (required)
3. The total reserve boundary length, multiplied by a modifier (optional)
4. The penalty for exceeding a preset cost threshold (optional – see footnote 3)

MARXAN Run-time parameters: Inedit.exe

- Run-time parameters are held in the input.dat file stored where the MARXAN.exe file is.
- Use Inedit.exe to modify/save settings:
  - Number of repeat runs
  - Annealing options:
    - Number of iterations per run
    - Threshold "cooling"
  - Heuristics
  - Location of inputs/outputs

Marxan run-time options



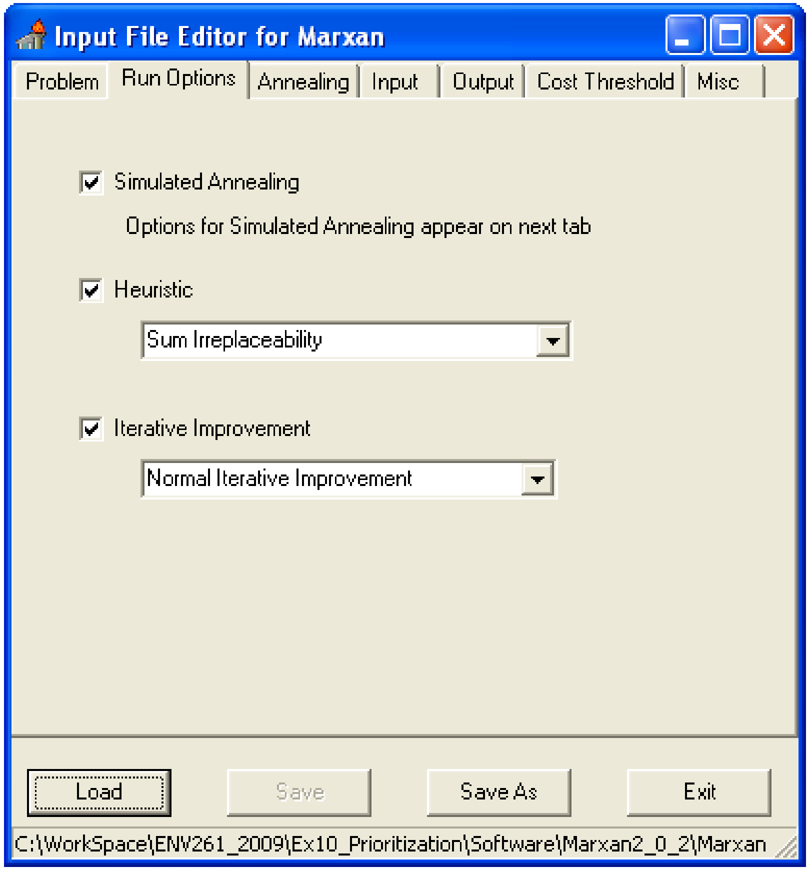
Repeat runs:

- Because *the* optimal solution is not guaranteed, we run several independent runs and compare results.

Boundary modifier:

- Adjusts boundary costs to match patch cost units
  - High value favors compactness

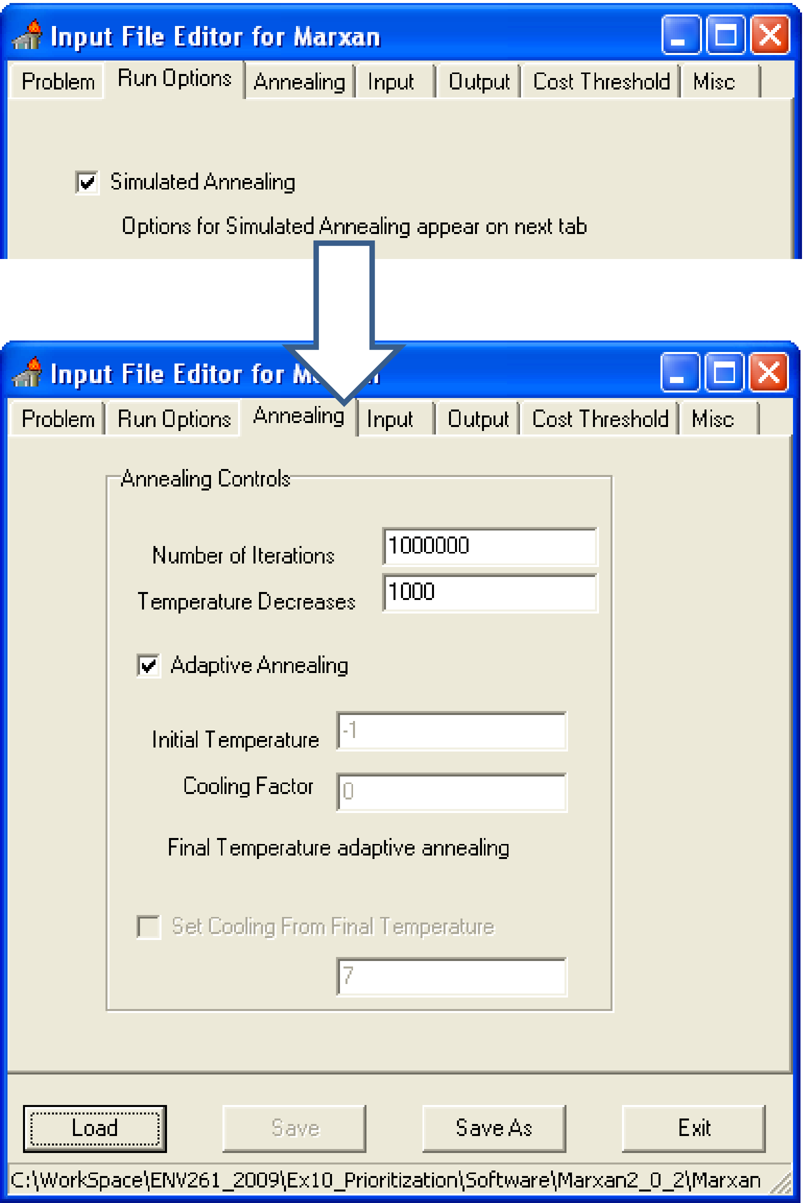
Marxan run-time options



Run Options

- Enable simulated annealing
- Heuristic method to use
- Iterative improvement method

Simulated annealing options



- Iterations – at each iteration, a planning unit is switched on (or off, if already chosen) and the change is evaluated via the objective function.

- Temperature decreases – the number of temperature decreases to occur across a set of iterations
  - Higher value – T° declines faster. Marxan seeks optimization more quickly, but may get "stuck" in a local minima.

- Adaptive Annealing – initial temperature and cooling rate are set by sampling the input data

Marxan: # runs vs. # iterations?

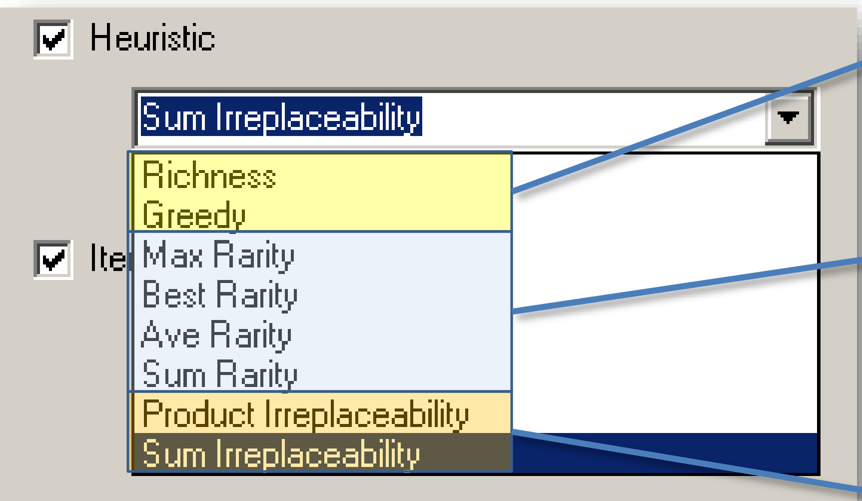
- Increasing # *runs* will
  - execute more searches "from scratch"
  - explore more complete paths to *an* optimal solution

- Increasing # *iterations* will mean
  - longer runs,
  - a deeper search for *the* "optimal" solution

Run:
new game

Iteration:
moves ahead to think

Heuristics options



- Greedy:
- Rarity:
- Irreplaceability

2

$$\sum_{ConValue} SPF \times Penalty$$

Heuristics options: Greedy

Greedy: add planning units that improve the objective function…

$$\sum_{ConValue} SPF \times Penalty$$

- Greedy *Richness*:
  - Conservation value = the sum of penalties incurred by under-representing a conservation feature; features already represented do not contribute
- *Pure* Greedy:
  - *Same as Richness when BLM is set to zero*; otherwise, boundary length is factored into the objective function, and not all species may be included because of cost restrictions

Heuristics options: Rarity

Greedy algorithms will favor common species; Rarity algorithms implement an added cost for not including rare conservation features

$$\frac{\text{Effective Abundance}}{(\text{Rarity x Planning Unit Cost})}$$

- Effective abundance = amount of feature found within a planning unit
- Rarity = fraction of planning unit in which feature is found

- **Maximum rarity** = planning unit assumes score of most rare feature
- **Best rarity** = planning unit assumes score of highest ratio for the patch (not necessarily the most rare feature)

Heuristics options: Rarity (continued)

- **Summed rarity** = sum of all feature rarity scores w/in the planning unit . Includes elements of both richness and rarity…

$$\frac{\sum_{cons.features} \frac{\text{Effective Abundance}}{\text{Rarity}}}{\text{Cost of Planning Unit}}$$

- **Avg. rarity** = average of all feature rarity scores w/in the planning unit; gives more weight to rarer conservation features…

$$\frac{\sum_{cons.features} \frac{\text{Effective Abundance}}{\text{Rarity}}}{\text{Cost x Number of Conservation Features}}$$

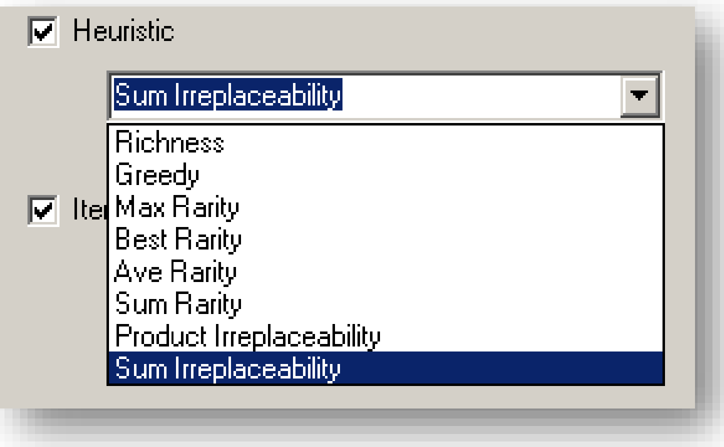
Heuristics options: Irreplaceability

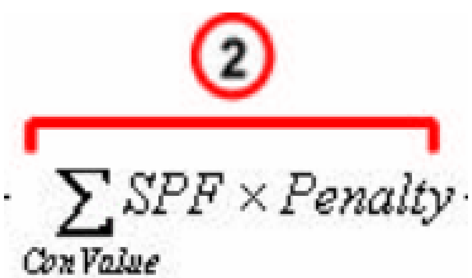
Irreplaceability looks at how necessary a patch is to achieving a given features' target

- A planning unit is irreplaceable if it contains a high proportion of the target that's not found in other planning unit .
- Calculated as how much "excess" (or buffer) is captured within the planning unit ; A planning unit that contains a mostly "buffer" or "excess" is more "replaceable"; Scores range from 0 to 1.

- *Product* irreplaceability: Sensitive to outliers; will favor planning unit with hard-to-represent features.
- *Summed* irreplaceability: Quantity of features is important; less sensitive to outliers.

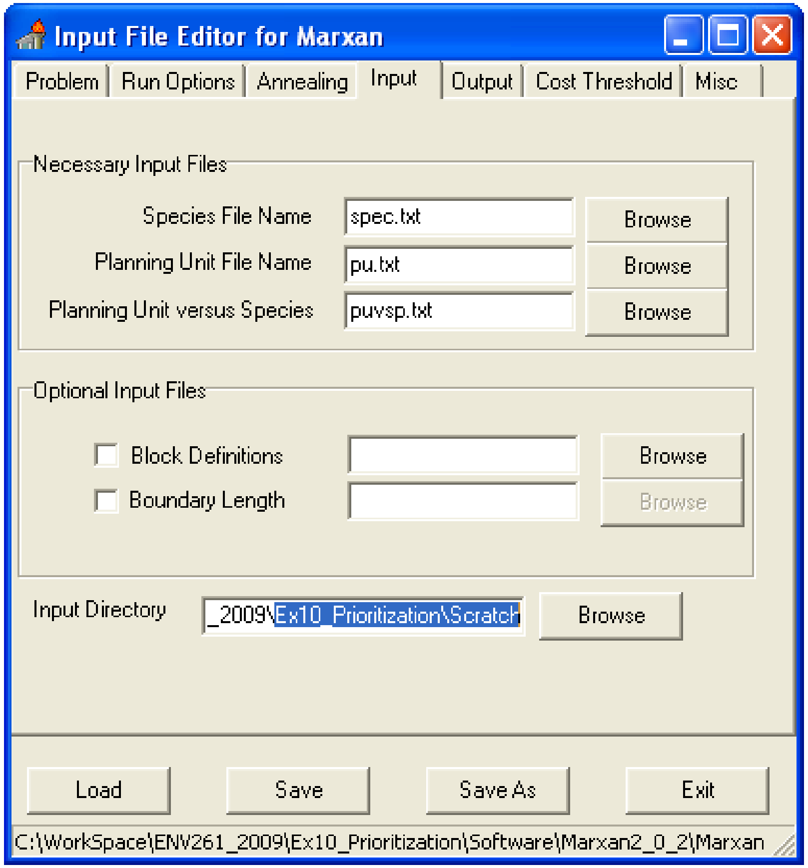
Heuristics options





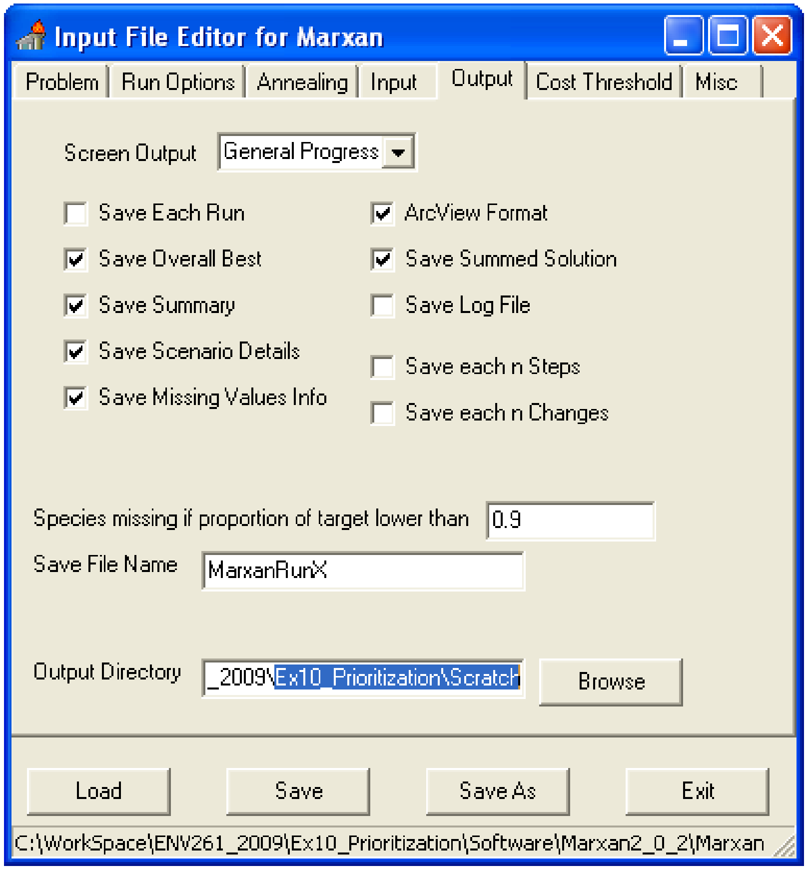
- Greedy:
  - Richness: Greedy strives to maximize richness at low cost
  - Pure Greedy: Cost (incl. boundary) play a larger role
- Rarity:
  - Outcome favors rare features; finds them first, and then adds common
  - Assigns values
- Irreplaceability
  - Examines how necessary a patch is to achieve a target for a given feature

Marxan run-time options: Inputs



- Specify input file names and location
- Block definitions can be used to set common properties to groups of conservation features (e.g. vertebrates, listed species, etc.)

Marxan run-time options: Outputs



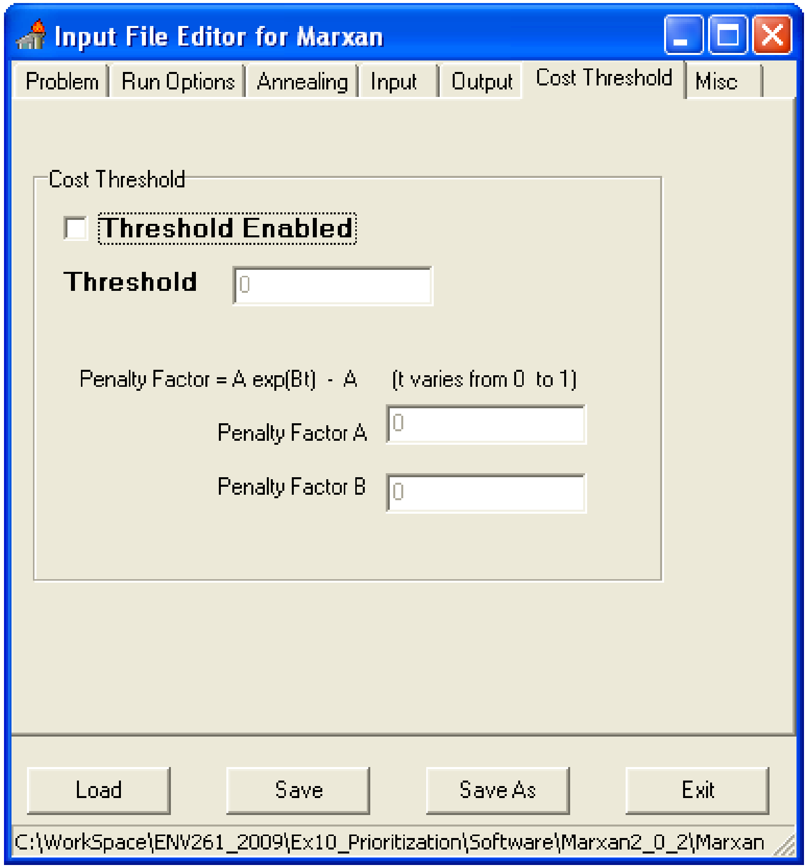
Specify output options & location

- See p. 66 of Marxan manual for what these are

Proportion Threshold…

Run name..

Marxan run-time options: Cost Threshold

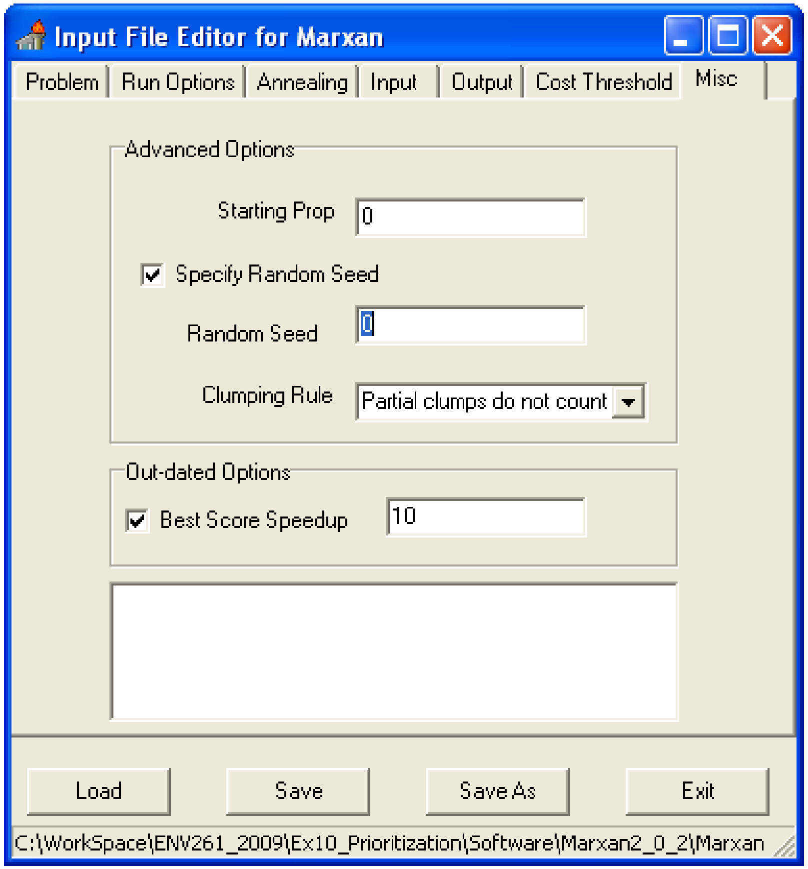


Used if you want to find a solution below a threshold cost

Otherwise, Marxan will strive to find the cheapest solution meeting all the targets.

see p. 33 of Marxan Manual

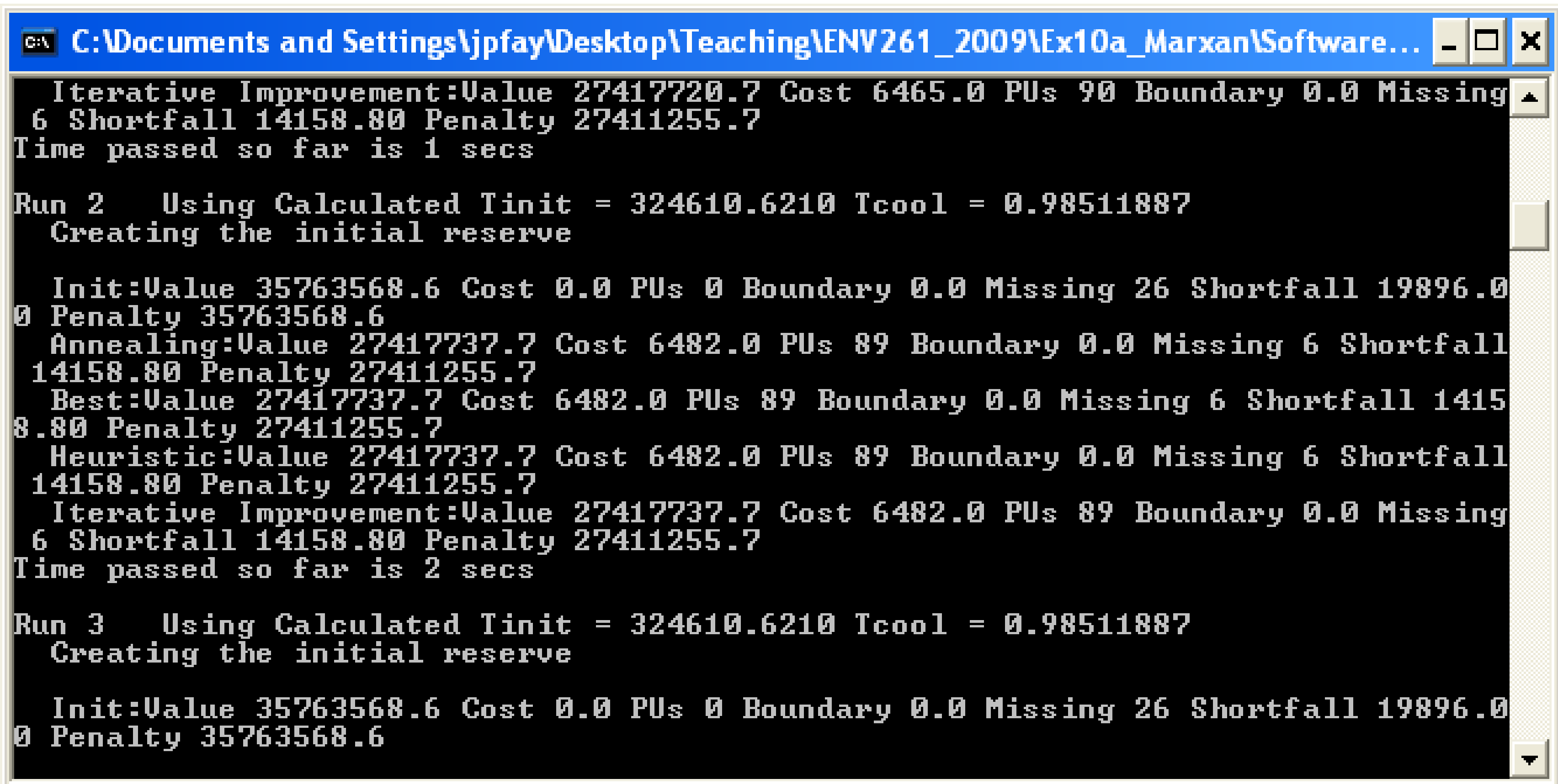
Marxan run-time options: Misc.



- Starting Prop: Proportion of planning units included in initial reserve.
- Random seed: If > 0, uses the same initial patches to start.
- Allows minimum feature size; features < minimum size are ignored.
- Marxan does not report "best" scores until a min. level is reached, thus saving time..

Running Marxan

- Double Click Marxan.exe file

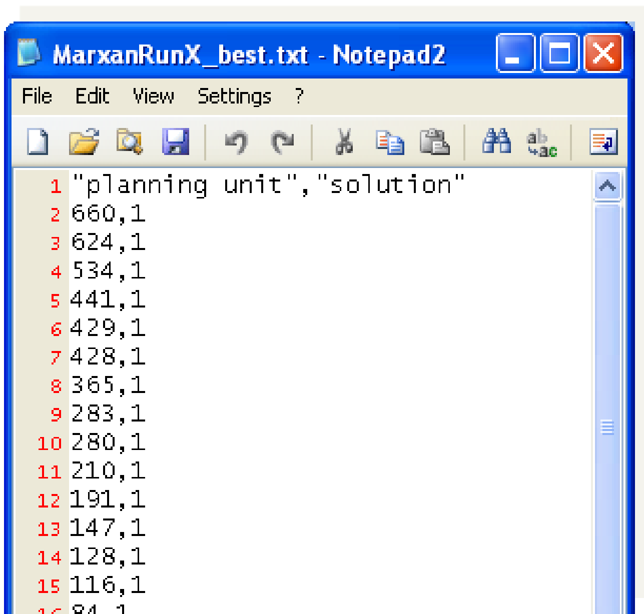


MARXAN Outputs

- run1_sen.dat
  Metadata file listing scenario run time parameters
- run1_best.dat
  list of planning units included in the optimal solution from all repeat runs
- run1_mvbest.txt
  a list of the conservation shortfalls contained in the optimal solution
- run1_ssoln.txt
  a list of each planning unit and the number of times it occurred in the optimal solution for a single repeat run
- run1_sum.txt
  a list breaking down the conservation value, costs, and overall score of each repeat run

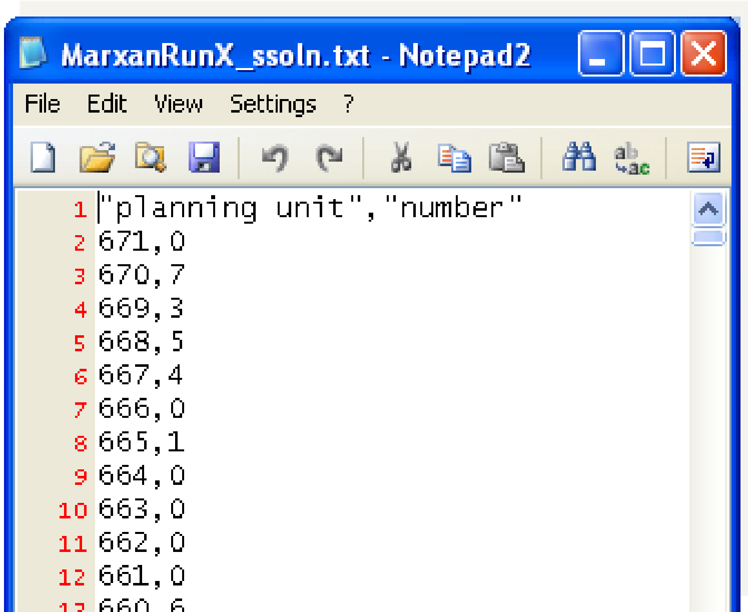
Viewing MARXAN's "best" solution

- Convert patch raster to polygon features
- Rename "planning unit" to "pu" in txt file
- Join txt file to patch attribute table
- Display patches where "solution" = 1



Viewing MARXAN's "summed solutions"

- Rename "planning unit" to "pu" in txt file
- Join txt file to patch attribute table
- Assign graduated symbology based on "number" attribute.



MARXAN: Parting thoughts

- Many additional options
  - **Read MARXAN manual!**
- Hexagons, parcels, watersheds, etc. instead of patches as planning units.
- Boundary length modifier
- Different cost measures, thresholds…
- Different conservation features…