British Columbia Exercise: How to go from defining your problem to running Marxan

Delivered by: Trevor Wiens

Materials provided by:

PacMARA

info@pacmara.org

Version: April 2014

PacMARA
Pacific Marine Analysis
& Research Association

Based on materials developed by:

Matthew Watts, Lindsay Kircher, and Hugh Possingham

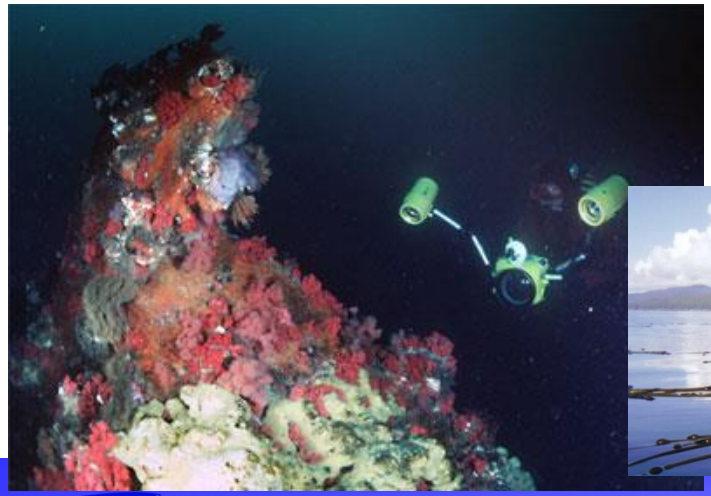
aεδa
Applied Environmental Decision Analysis
Commonwealth Environmental Research Facility

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

Define problem

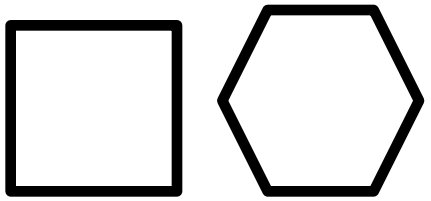
Identify **cost-effective areas** for inclusion in a reserve system in British Columbia that **protects a pre-defined target amount of each conservation feature** (e.g. 30% of each benthic habitat type, and 30% of known sponge reefs and kelp)

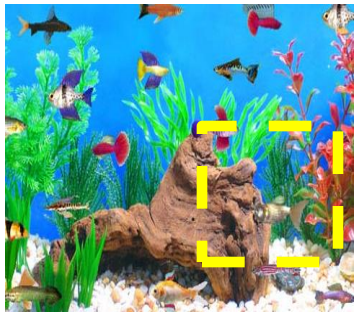
Planning Region

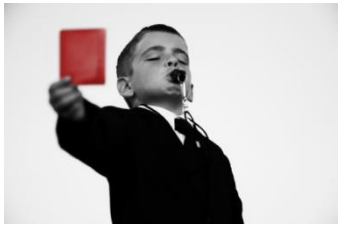


Marxan Input Data and Parameters

- Planning units
- Features
- Targets
- Costs
- Boundary Length
- Species Penalty Factor
- Boundary Length Modifier (BLM)
- Number of solutions
- Number of iterations

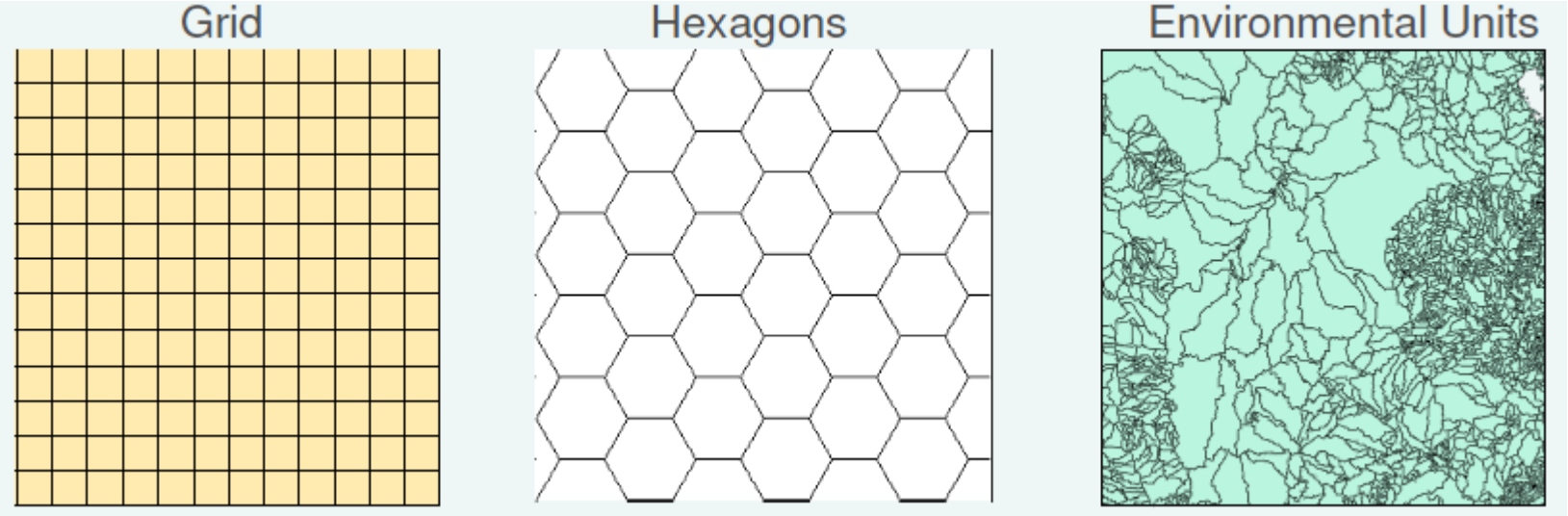






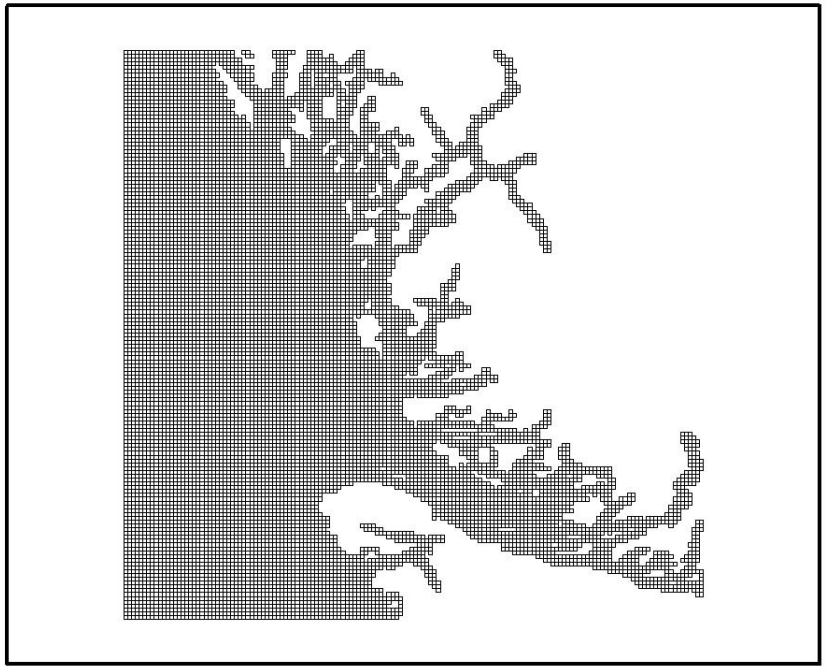
Planning Units

Size and shape of PUs is informed by the scale of planning and resolution of datasets being used



Planning units

For this exercise you will be using a 20 km² square grid



Conservation Features

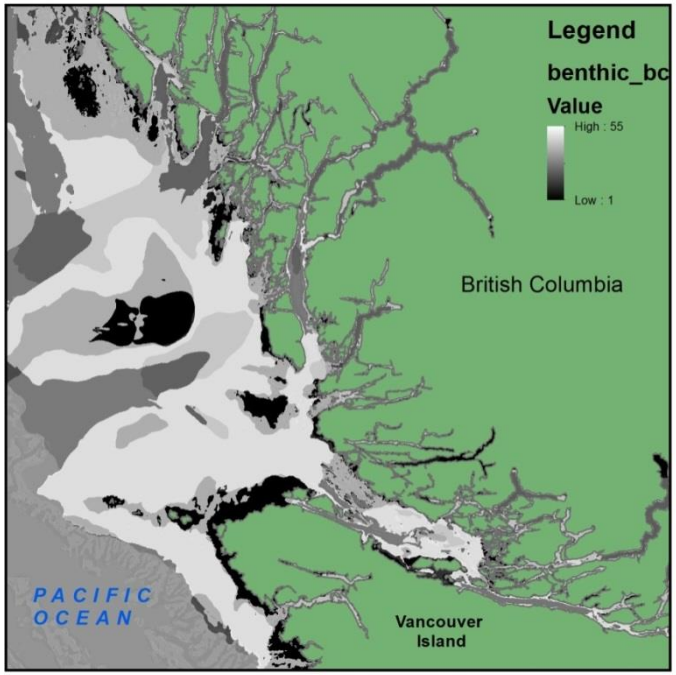
Which data to include?

- Criteria need to be informed by project goals and objectives
- Data availability
- Data consistency (spatial, temporal, representational)

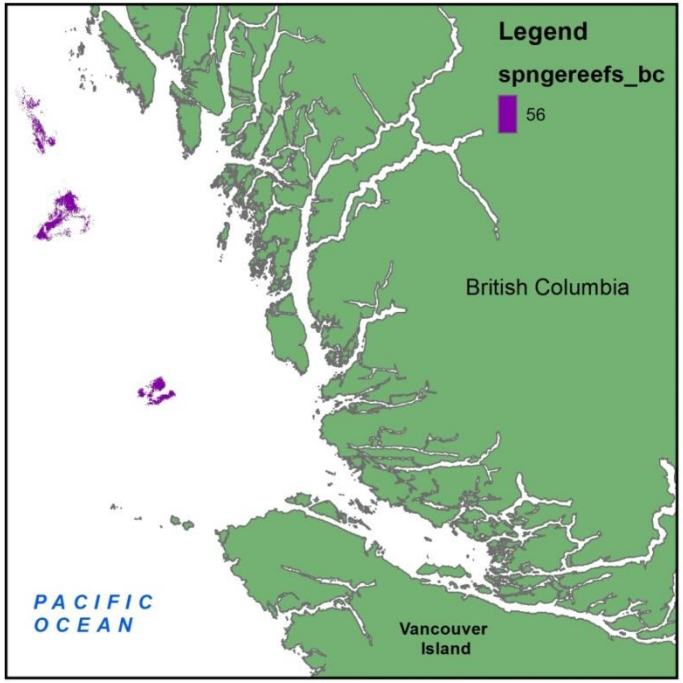
Source: H. Grantham (CI)

BC Exercise: Conservation feature data

Benthic habitat types



Sponge reef distribution

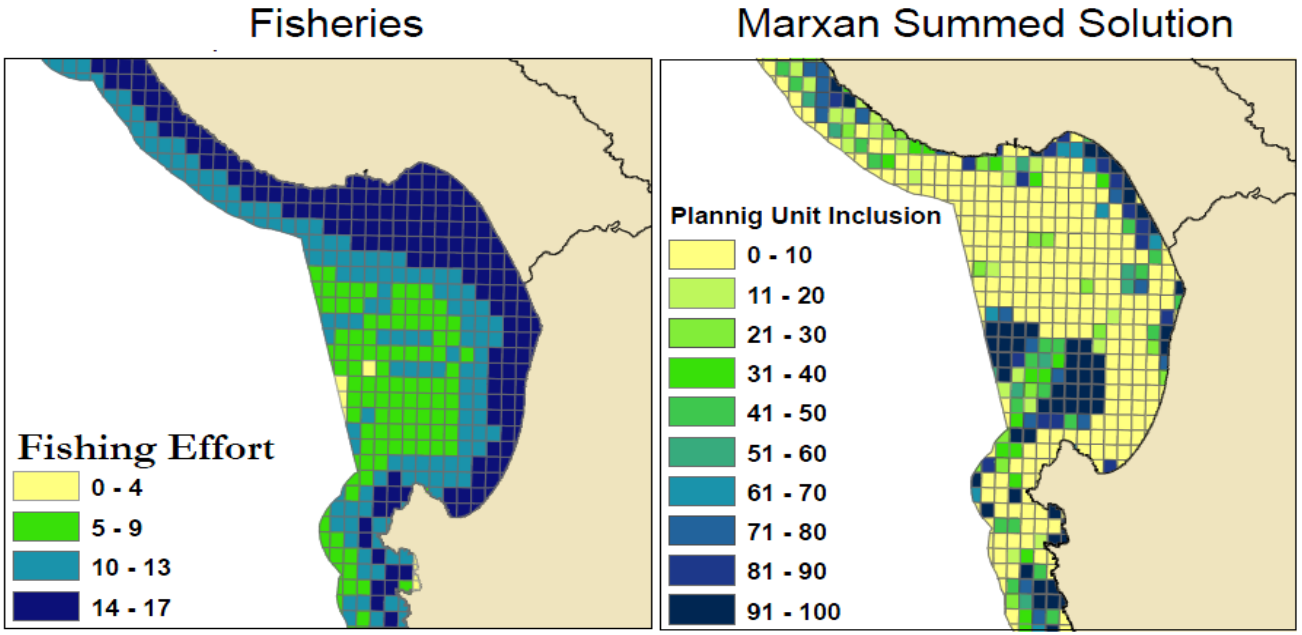


and **6 other conservation features (both rare and common)**

Planning Unit Cost

It is critical to identify the right cost(s), as the choice will impact the location of reserves

- But they are usually the hardest GIS data to find
- Marxan can only include one cost layer



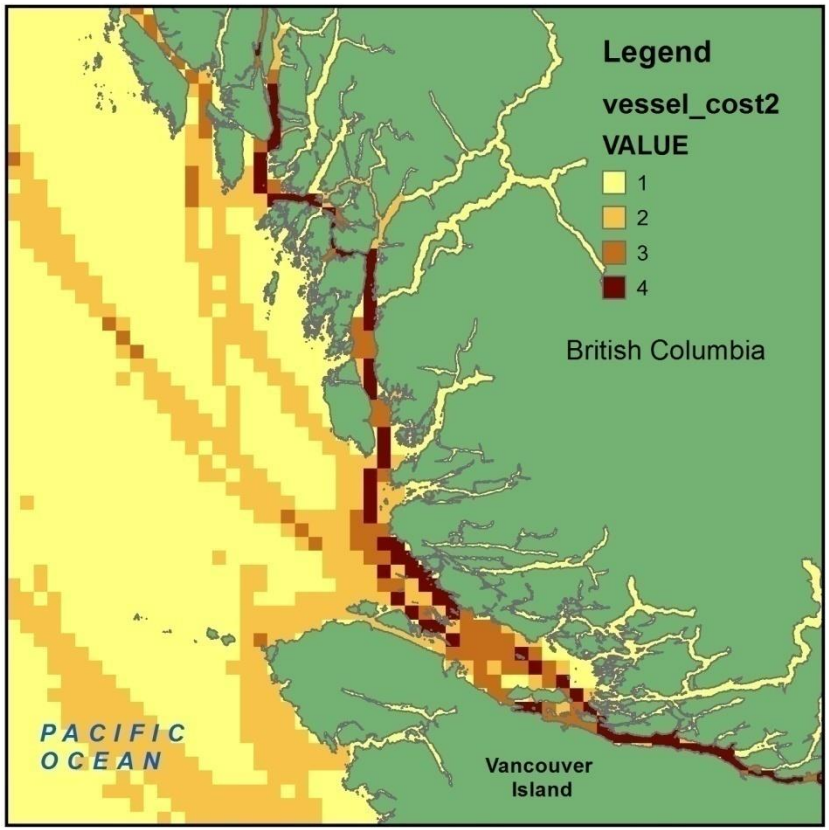
Source: Klein et al., 2009; Carwardine et al., 2008

PacMARA
Pacific Marine Analysis & Research Association

BC Exercise: Cost data

Vessel movement data (summer 2007)

- Represents **socio-economic cost**
- Total **vessels movements over the summer**. Vessels included: Fishing, carrier, ferry, tug, oil tanker, cruise ship



Process data into Marxan format

For Marxan to solve the problem you have defined, you need to organize the data you have collected into a specific format

This has been done already

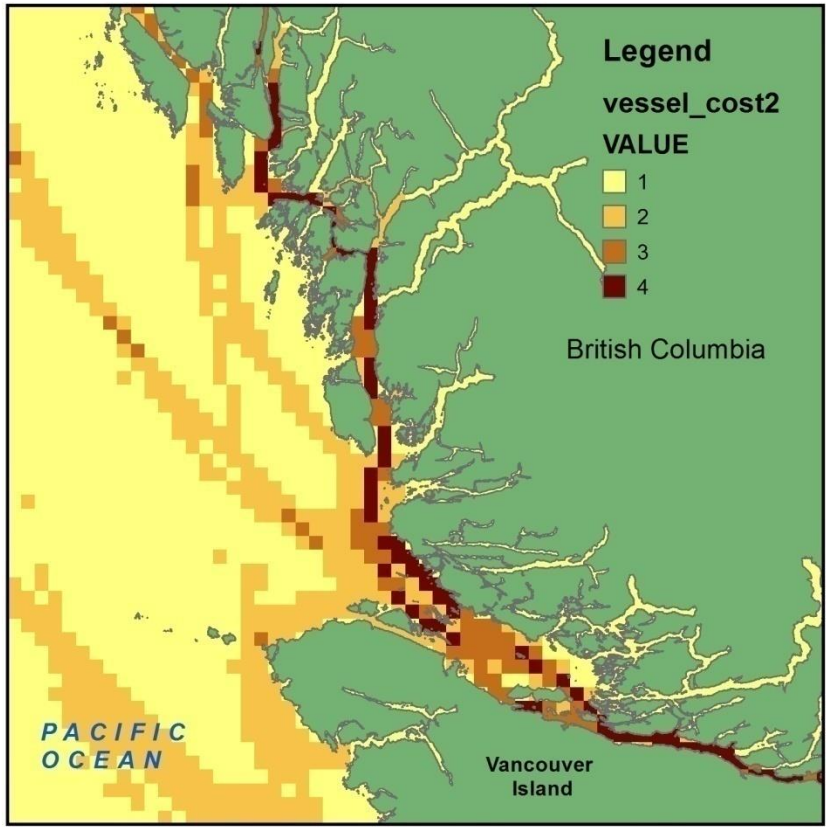
Can be done with a variety of tools including: ArcGIS, QGIS, IDRISI & R

Key data files & contents

- **pu.dat:** Planning units, cost and status
- **spec.dat & puvsp.dat:** Amount of each conservation features in each planning unit
- **bound.dat:** Boundaries between planning units

Pu.dat

- **id** = Planning units
- **cost** = How much each planning unit "costs"



id	cost	status
1	256	0
2	320	0
3	320	0
4	320	0
5	320	0
6	320	0
7	320	0
8	320	0
9	320	0
10	320	0
11	320	0
12	320	0
13	320	0
14	320	0
15	320	0
16	320	0
17	320	0
18	320	0
19	320	0
20	320	0

Pu.dat

- "**status**"
  - Available (0): Default
  - Seed (1): Starting point (Not often used)
  - Lock in (2): e.g. for areas already protected
  - Lock out (3): for areas that can't be included (e.g. urban areas)

id	cost	status
1	256	0
2	320	0
3	320	0
4	320	0
5	320	0
6	320	0
7	320	0
8	320	0
9	320	0
10	320	0
11	320	0
12	320	0
13	320	0
14	320	0
15	320	0
16	320	0
17	320	0
18	320	0
19	320	0
20	320	0

Spec.dat

A list of all your conservation features (**id**), the targets you have set for them (**prop**), and the species penalty factor (**spf**)

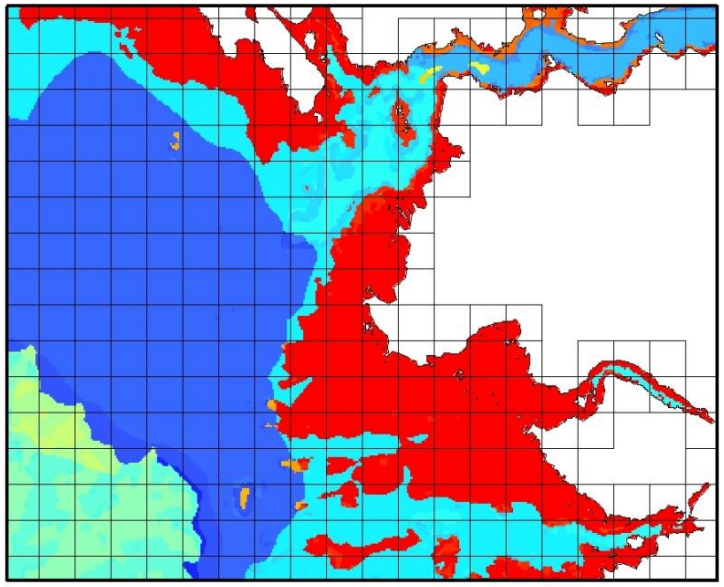
id	name	prop	spf
1	0_20 Hard Depression	0.3	1
2	0_20 Hard Flat	0.3	1
3	0_20 Hard Ridge	0.3	1
4	0_20 Hard Slope	0.3	1
5	0_20 Muddy Depression	0.3	1
6	0_20 Muddy Flat	0.3	1
7	0_20 Muddy Ridge	0.3	1
8	0_20 Muddy Slope	0.3	1
9	0_20 Sandy Depression	0.3	1
10	0_20 Sandy Flat	0.3	1
11	0_20 Sandy Ridge	0.3	1
12	0_20 Sandy Slope	0.3	1
13	20_50 Hard Depression	0.3	1
14	20_50 Hard Flat	0.3	1
15	20_50 Hard Ridge	0.3	1
16	20_50 Hard Slope	0.3	1
17	20_50 Muddy Depression	0.3	1
18	20_50 Muddy Flat	0.3	1
19	20_50 Muddy Ridge	0.3	1
20	20_50 Muddy Slope	0.3	1
21	20_50 Sandy Depression	0.3	1
22	20_50 Sandy Flat	0.3	1
23	20_50 Sandy Ridge	0.3	1
24	20_50 Sandy Slope	0.3	1
25	200plus Hard Depression	0.3	1
26	200plus Hard Flat	0.3	1
27	200plus Hard Ridge	0.3	1
28	200plus Hard Slope	0.3	1

Species penalty factor

- Marxan calculates whether the target for each conservation feature is met
- Includes a cost for any target that has not been met (specified by the species penalty factor - SPF)
- SPF can be generic or feature-specific
- If targets are not met, increasing the SPF value in the spec.dat file will help in achieving conservation targets

Puvsp.dat

Contains how much of each feature is in each planning unit



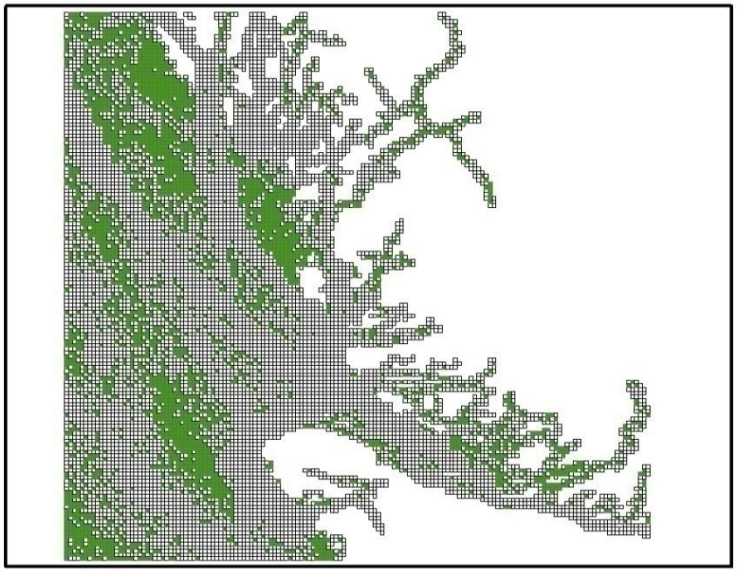
species	pu	amount
36	1	2210000
39	1	400000
40	1	220000
41	1	1910000
42	1	10000
36	2	1210000
39	2	60000
41	2	70000
42	2	10000
36	3	2450000
39	3	2190000
40	3	180000
41	3	2770000
64	3	7860000
36	4	8730000
38	4	130000
39	4	1340000

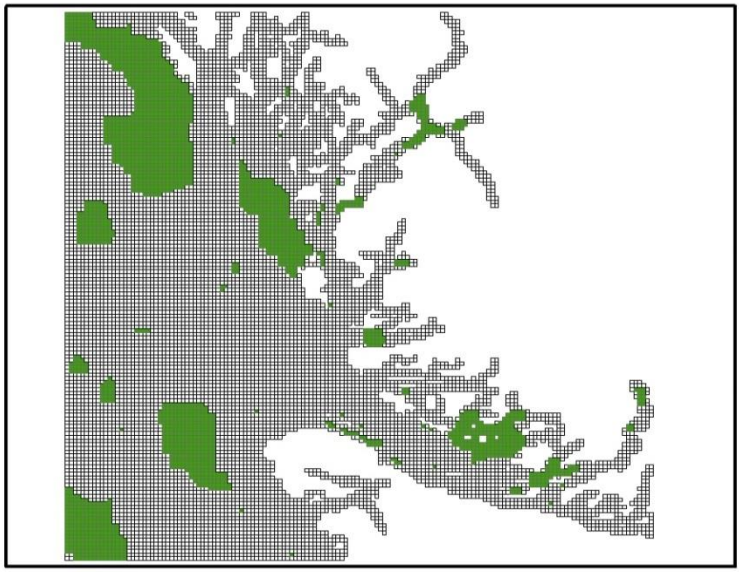
The boundary cost

- The total amount of edge that the planning units share with unprotected units (e.g., unconnected planning units will have a large cost)
- The cost is usually quantified as the length of the PU edge
- Boundary cost is multiplied by the boundary length modifier (BLM), which is a user-defined value
- Increasing the BLM increases the cost of having a fragmented set of conservation areas

Bound.dat

- Marxan can find clumped solutions

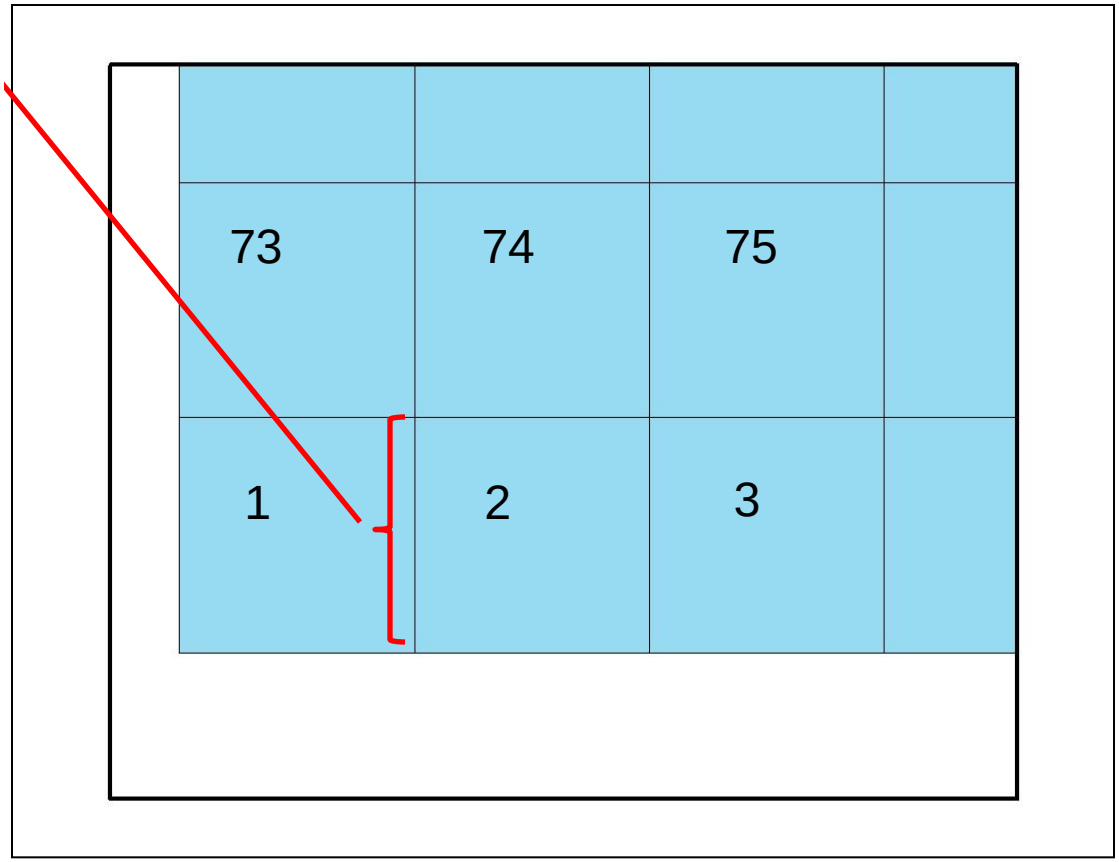




- But it needs spatial information about the boundaries between the planning units to do this

Bound.dat

id1	id2	boundary
1	2	2000
1	73	2000
1	1	4000
2	3	2000
2	74	2000
2	2	2000
3	4	2000
3	75	2000
3	3	2000
4	5	2000
4	76	2000
4	4	2000
5	6	2000
5	77	2000
5	5	2000
6	7	2000
6	78	2000
6	6	2000
7	8	2000
7	79	2000
7	7	2000
8	9	2000
8	80	2000
8	8	2000
9	10	2000
9	81	2000
9	9	2000
10	11	2000
10	82	2000
10	10	2000
11	12	2000



Input.dat

Instructions for Marxan

Parameters typically changed:

General Parameters:

- BLM: (clumping)
- NUMREPS: # of runs (or solutions)

Annealing Parameters:

- NUMITNS: # of iterations

```
Input parameter file for Marxan.

This file generated by Zonae Cogito.
written by Matt Watts
m.watts@uq.edu.au

General Parameters
BLM 0
PROP 0.5
RANDSEED -1
NUMREPS 10

Annealing Parameters
NUMITNS 1000000
STARTTEMP -1
NUMTEMP 10000

Input Files
INPUTDIR input
PUNAME pu.dat
SPECNAME spec.dat
PUVSPRNAME puvspr.dat
MATRIXSPORDERNAME sporder.dat
BOUNDNAME bound.dat

Save Files
SCENNAME output
```

Input.dat

Instructions for Marxan

Input files:

- INPUTDIR: Input directory
- Input file names

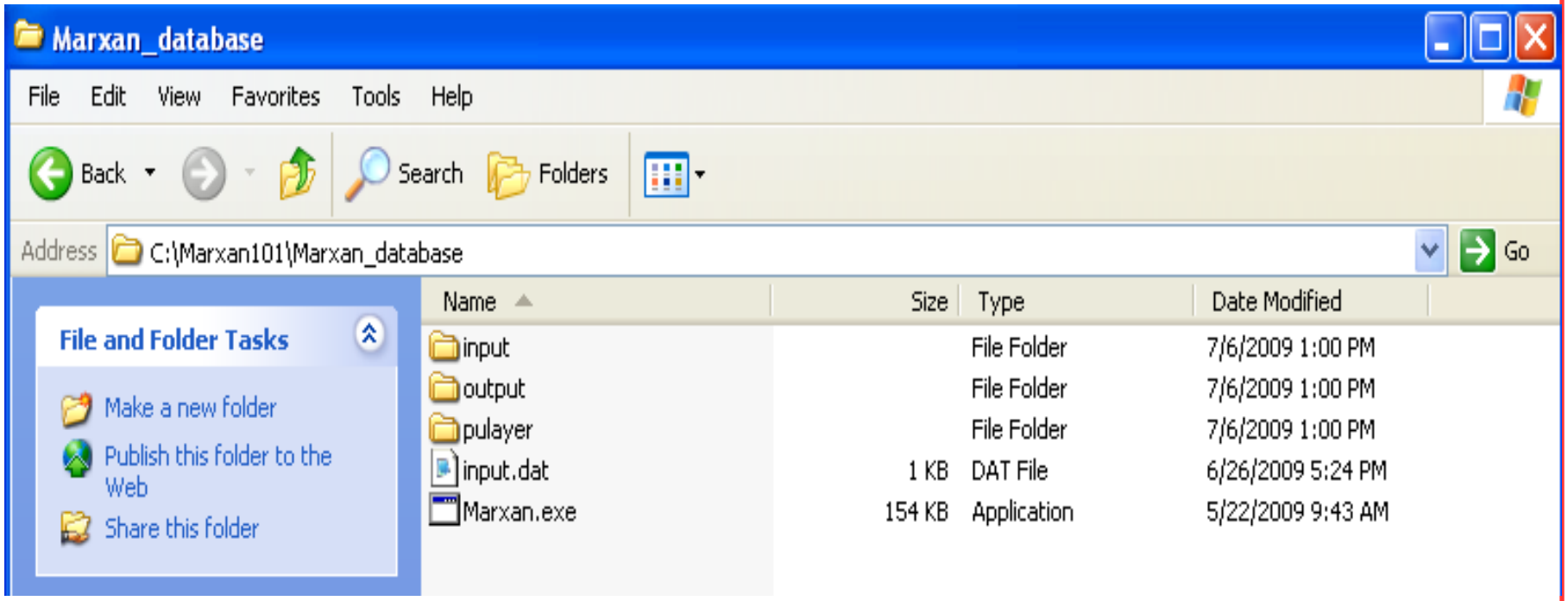
Output files:

- Preprocessing file names
- Output directory

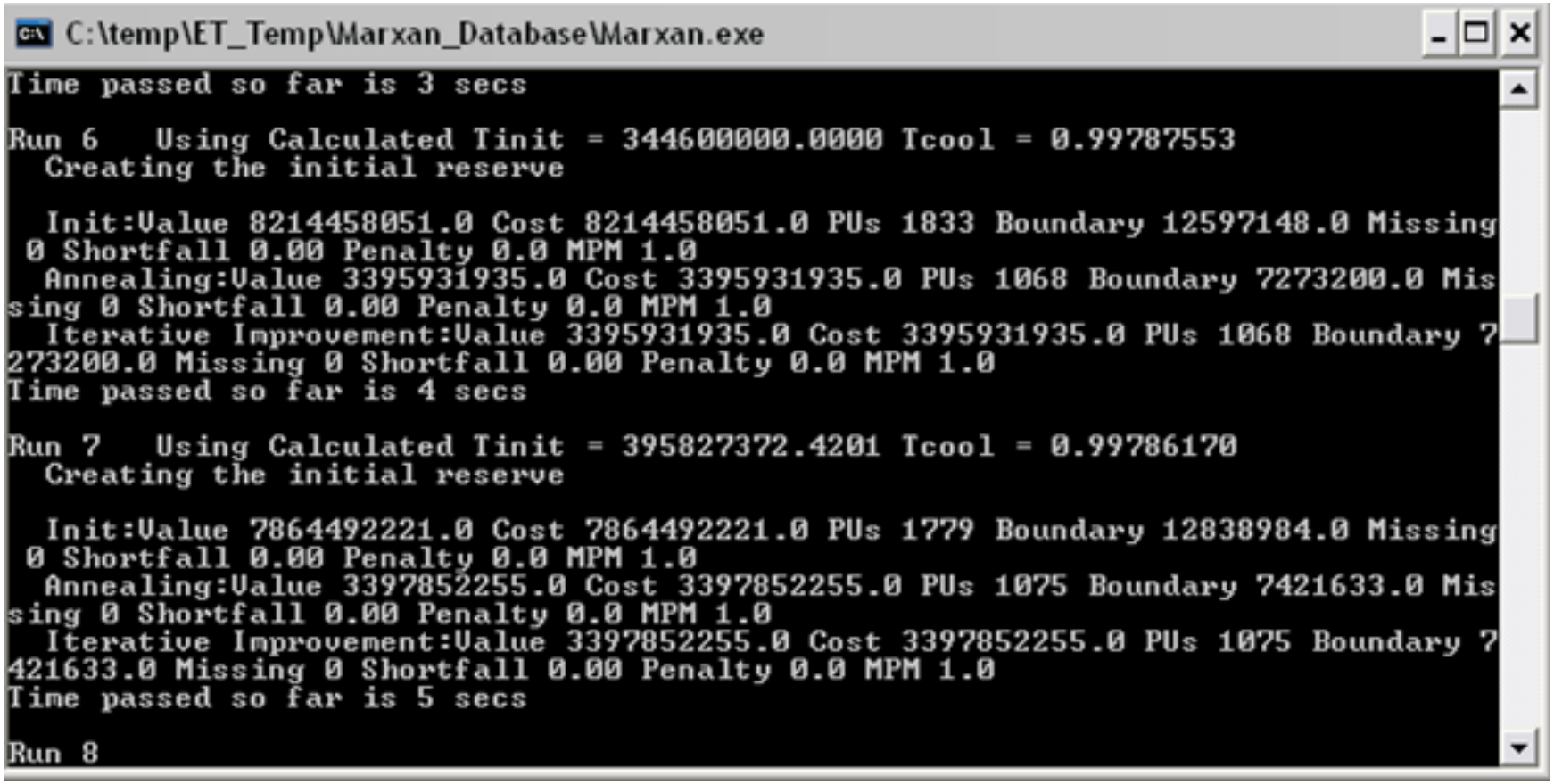
Program control (not often changed)

```
General Parameters
BLM 0
PROP 0.5
RANDSEED -1
NUMREPS 10

Annealing Parameters
NUMITNS 1000000
STARTTEMP -1
NUMTEMP 10000

Input Files
INPUTDIR input
PUNAME pu.dat
SPECNAME spec.dat
PUVSPRNAME puvspr.dat
MATRIXSPORDERNAME sporder.dat
BOUNDNAME bound.dat

Save Files
SCENNAME output
SAVERUN 3
SAVEBEST 3
SAVESUMMARY 3
SAVETARGMET 3
SAVESUMSOLN 3
SAVEPENALTY 3
SAVESCEN 1
SAVELOG 1
OUTPUTDIR output

Program control
RUNMODE 1
MISSLEVEL 1
ITIMPTYPE 0
HEURTYPE -1
CLUMPTYPE 0
VERBOSITY 3
SAVESOLUTIONSMATRIX 3
```

Database structure



Run Marxan!


C:\temp\ET_Temp\Marxan_Database\Marxan.exe
Time passed so far is 3 secs
Run 6 Using Calculated Tinit = 344600000.0000 Tcool = 0.99787553
Creating the initial reserve
Init:Value 8214458051.0 Cost 8214458051.0 PUs 1833 Boundary 12597148.0 Missing 0 Shortfall 0.00 Penalty 0.0 MPM 1.0
Annealing:Value 3395931935.0 Cost 3395931935.0 PUs 1068 Boundary 7273200.0 Missing 0 Shortfall 0.00 Penalty 0.0 MPM 1.0
Iterative Improvement:Value 3395931935.0 Cost 3395931935.0 PUs 1068 Boundary 7273200.0 Missing 0 Shortfall 0.00 Penalty 0.0 MPM 1.0
Time passed so far is 4 secs
Run 7 Using Calculated Tinit = 395827372.4201 Tcool = 0.99786170
Creating the initial reserve
Init:Value 7864492221.0 Cost 7864492221.0 PUs 1779 Boundary 12838984.0 Missing 0 Shortfall 0.00 Penalty 0.0 MPM 1.0
Annealing:Value 3397852255.0 Cost 3397852255.0 PUs 1075 Boundary 7421633.0 Missing 0 Shortfall 0.00 Penalty 0.0 MPM 1.0
Iterative Improvement:Value 3397852255.0 Cost 3397852255.0 PUs 1075 Boundary 7421633.0 Missing 0 Shortfall 0.00 Penalty 0.0 MPM 1.0
Time passed so far is 5 secs
Run 8