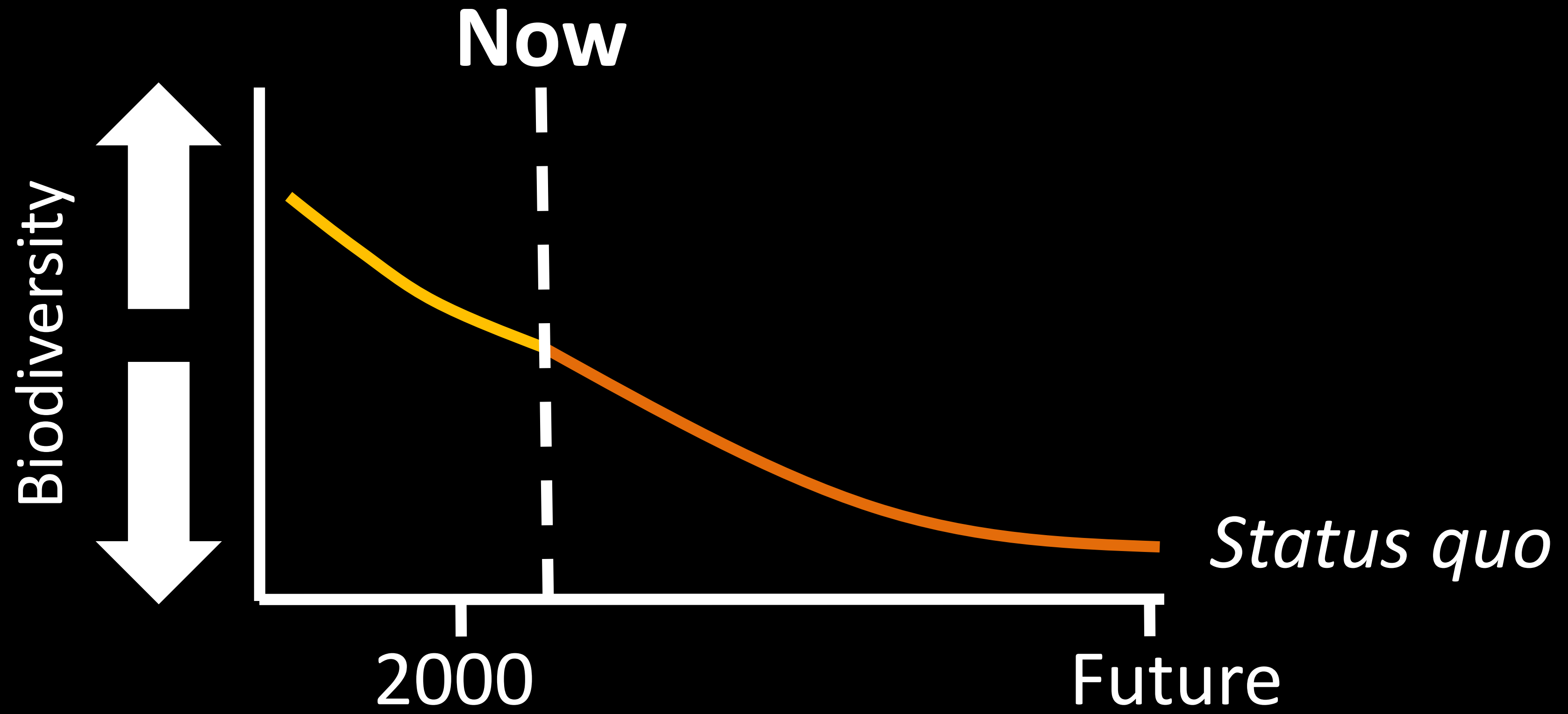
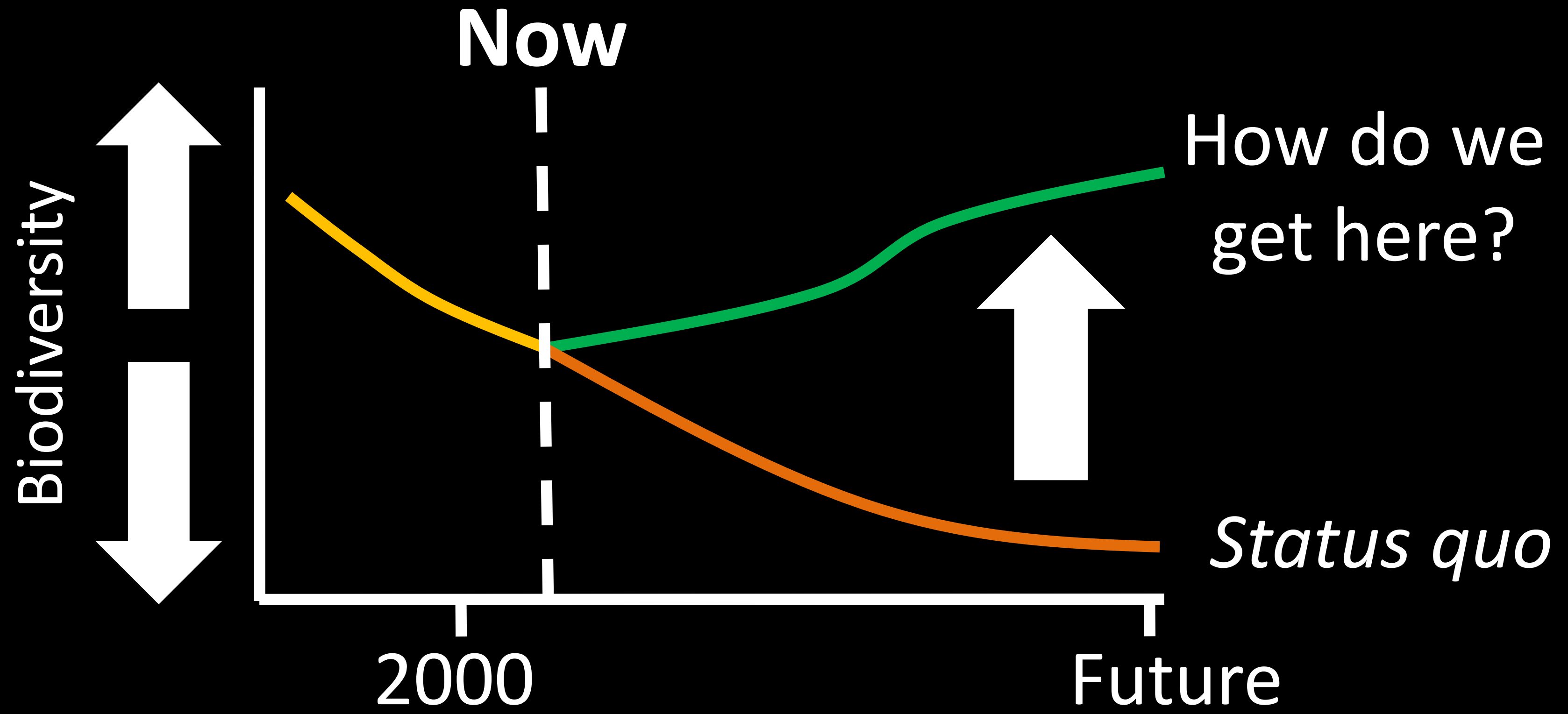
Conservation decisions with exact algorithm solvers

Jeffrey Hanson

jeffrey.hanson@uqconnect.edu.au

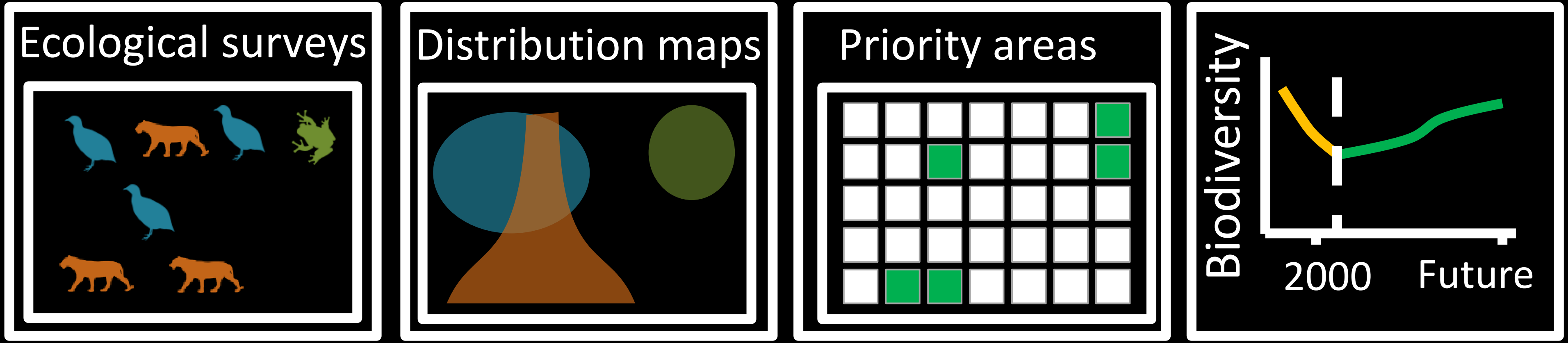
jeffrey-hanson.com


Now
Biodiversity
Status quo
2000
Future



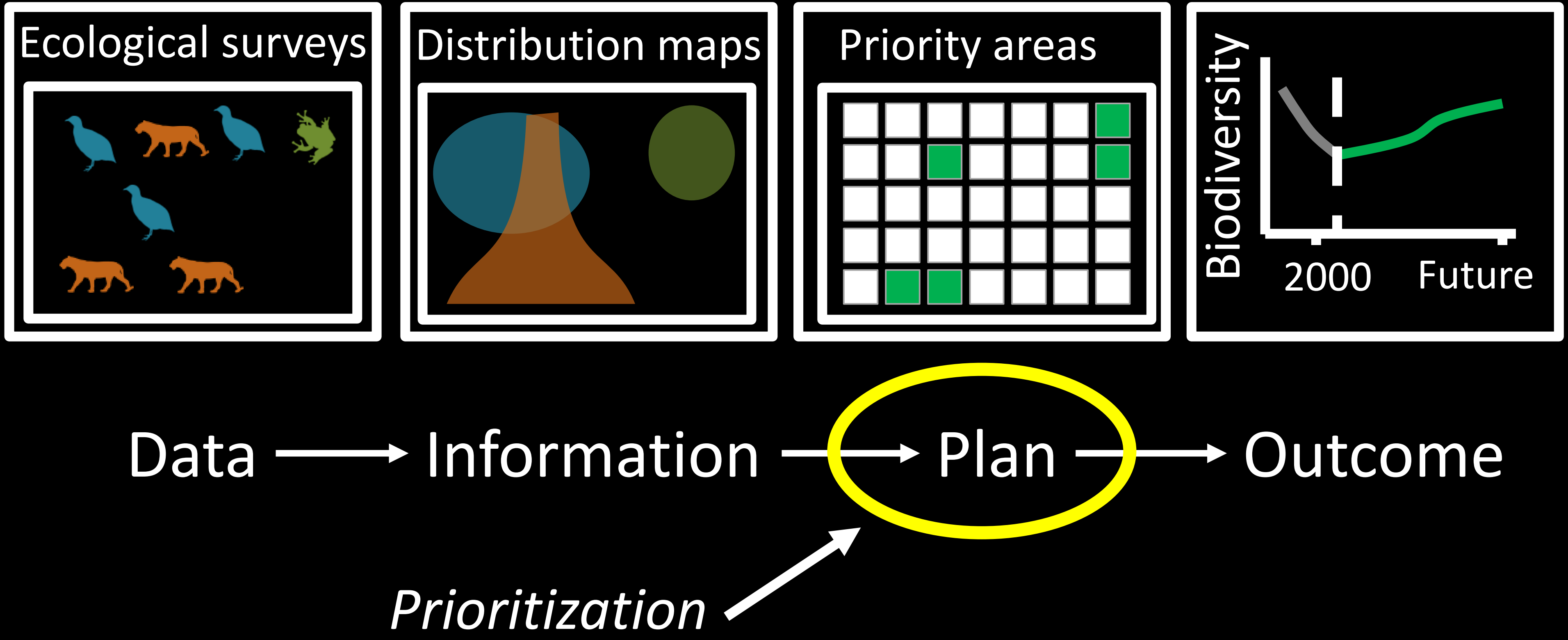
Now
Biodiversity
How do we
get here?
Status quo
2000
Future

How do we bend the curve?



Data → Information → Plan → Outcome

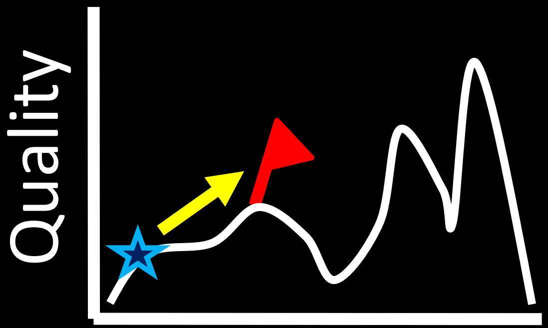
How do we bend the curve?



Framing conservation as a decision science problem

- <u>Goal</u>: what is our vision for the future?
- <u>Objective</u>: what quantity are we maximizing/minimizing to help achieve the goal?
- <u>Constraints</u>: what things must our solution do to help achieve the goal?
- <u>Decisions</u>: what actions could we do to maximize/minimize the objective?

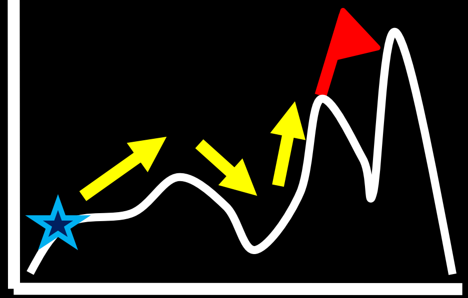
Heuristic algorithm



Different solutions

PPP

Meta-heuristic algorithms

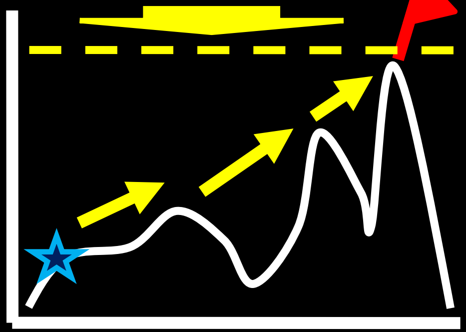


Different solutions

MARXAN
conservation solutions

Exact algorithms

Estimate of best quality



Different solutions

GUROBI
OPTIMIZATION
IBM
CPLEX
COIN|OR

Exact algorithm solvers

- Open source and commercial solvers available
  (e.g., Gurobi, IBM CPLEX, CBC, HiGHS, SYMPHONY)

- Automatically select algorithms for different problems
  (e.g., presolve, simplex, barrier, branch-and-bound)

- Broadly speaking have similar functionality, but have different implementations of the underlying algorithms, and have different performance for different kinds of problems

See here for explanation of branch and bound algorithm: https://www.youtube.com/watch?v=CdfZ4XOpSho

Exact algorithm solvers

- Solve multiple problem types:
  - linear programming (LP) =
    continuous decision variables

  - integer programming (IP/ILP) =
    binary/integer decision variables

  - mixed integer linear programming (MILP) =
    continuous + binary/integer decision variables

Case studies

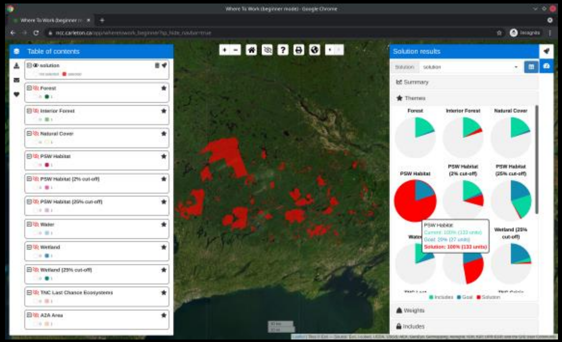
Scheduling plane flights to reduce fuel and operational costs

Keeping ATMs stocked with cash, while minimizing costs

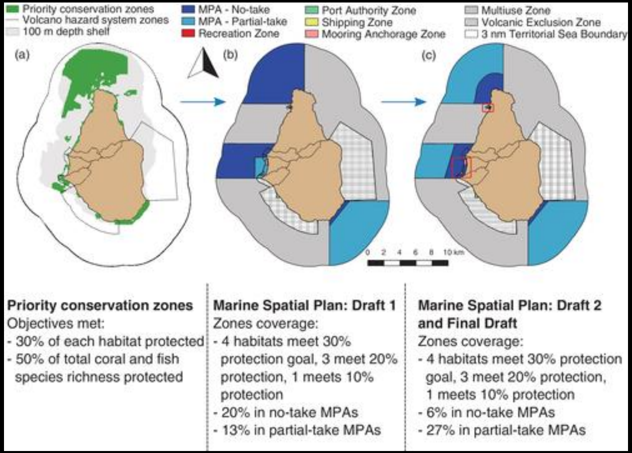
Basketball game schedules

For more examples, see https://www.gurobi.com/case_studies/

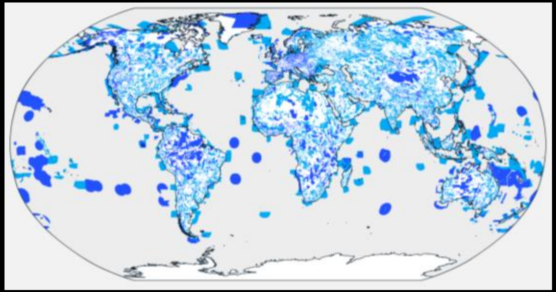
Conservation examples



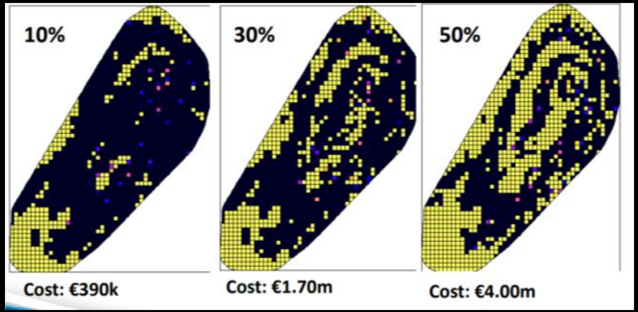
Nature Conservancy of Canada for land acquisition



Waitt Institute to help Government of Montserrat



McKinsey Consulting

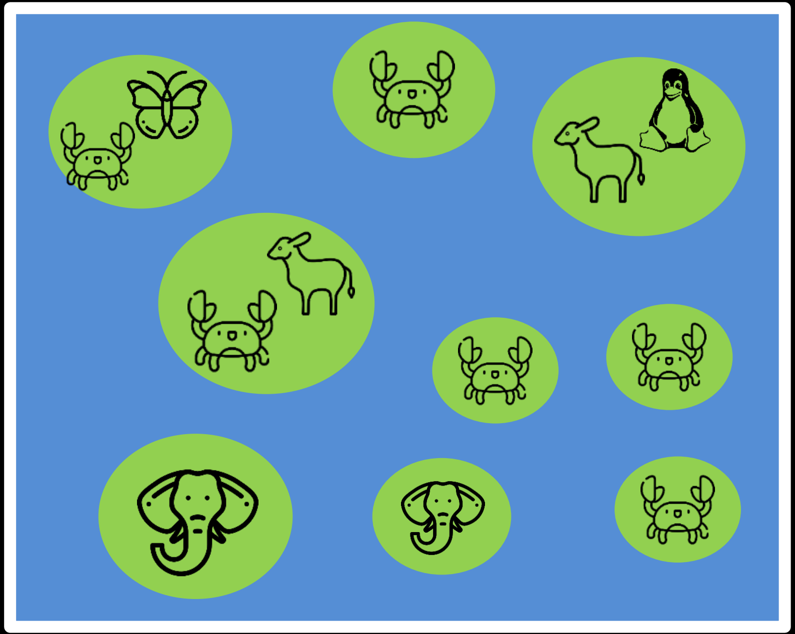


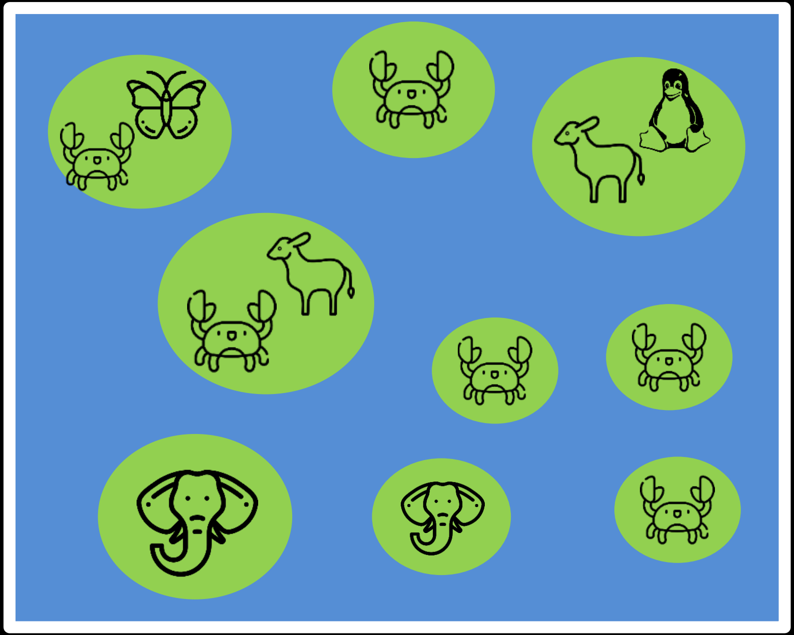
Scottish Government for marine spatial planning in Rockall Bank

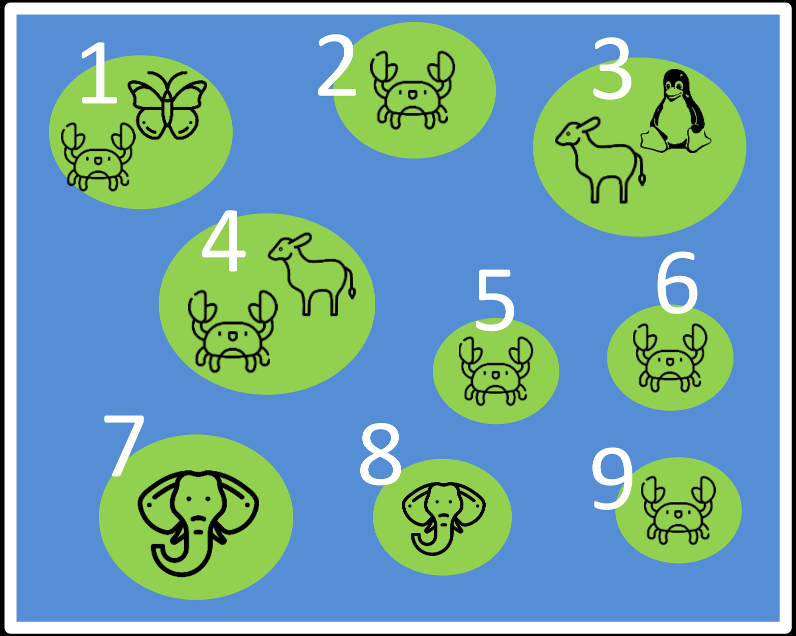
USGS to prioritize recovery areas in Hawaii

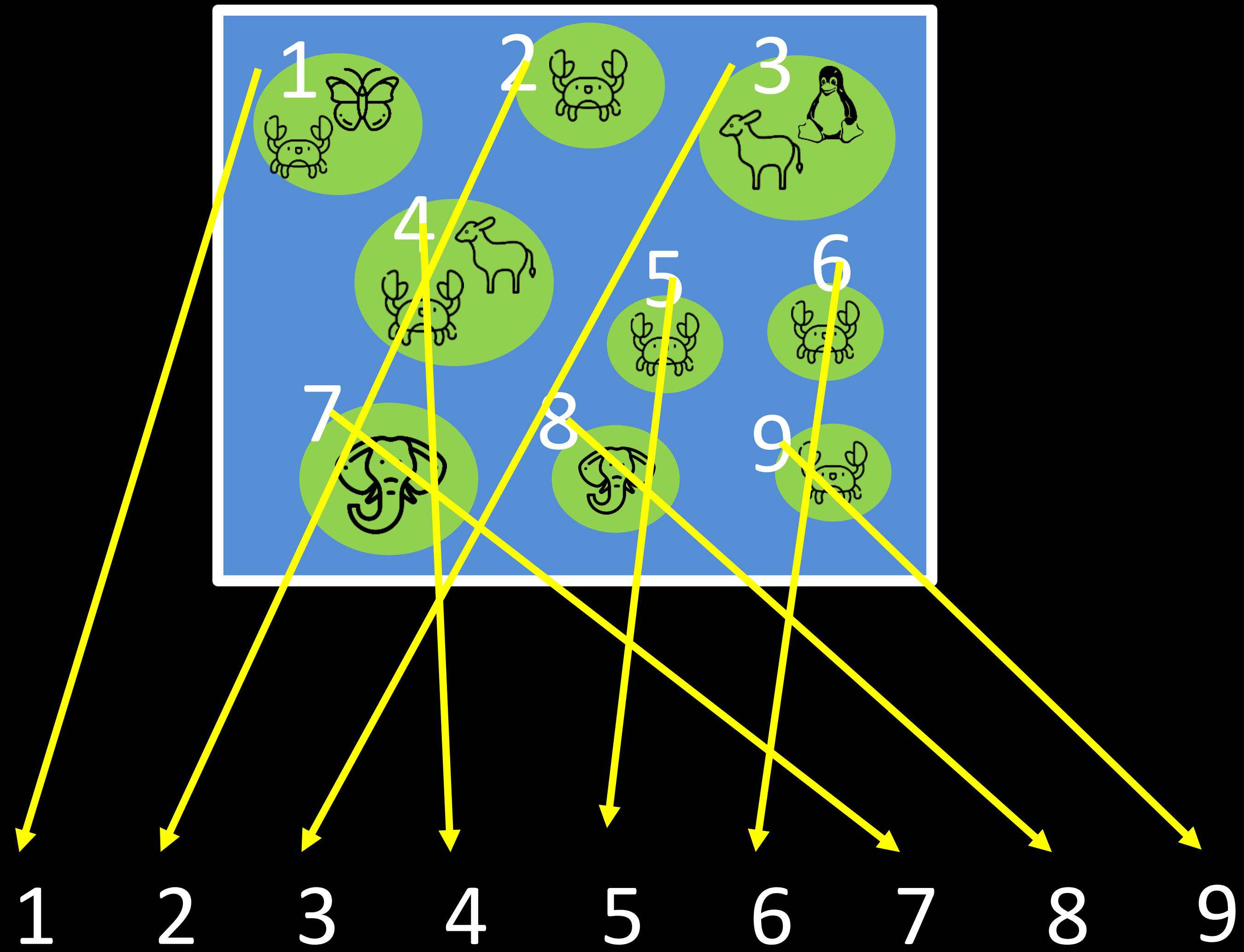
Reserve selection as optimization

- <u>Goal</u>: conserve biodiversity
- <u>Objective</u>: min. # of islands
- <u>Constraints</u>: sufficient habitat for each species
- <u>Decisions</u>: create a reserve on an island or not?





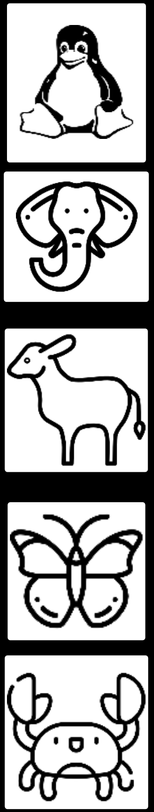

1
2
3
4
5
6
7
8
9


1
2
3
4
5
6
7
8
9
1 2 3 4 5 6 7 8 9

1 2 3 4 5 6 7 8 9

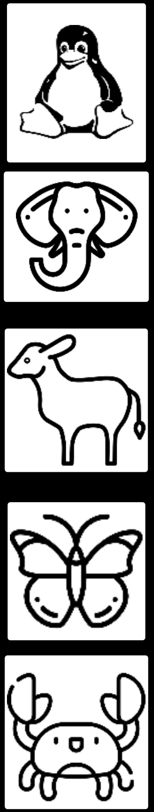
Min €:	+1	+1	+1	+1	+1	+1	+1	+1	+1
	1	2	3	4	5	6	7	8	9

Min €: +1 +1 +1 +1 +1 +1 +1 +1 +1

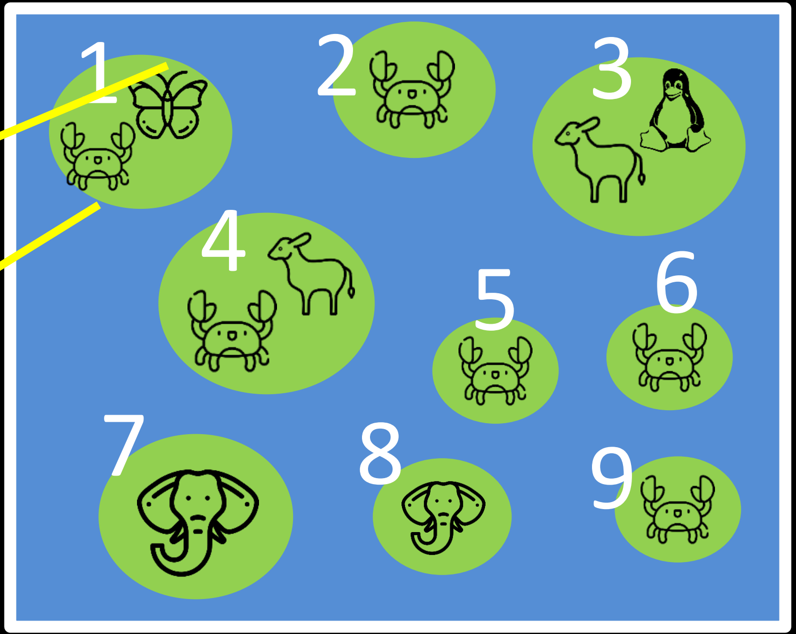


1 2 3 4 5 6 7 8 9

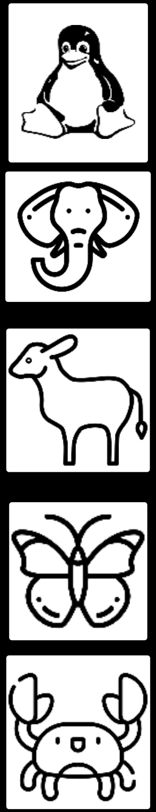
Min €: +1 +1 +1 +1 +1 +1 +1 +1 +1



	1	2	3	4	5	6	7	8	9
	+1								
	+1								



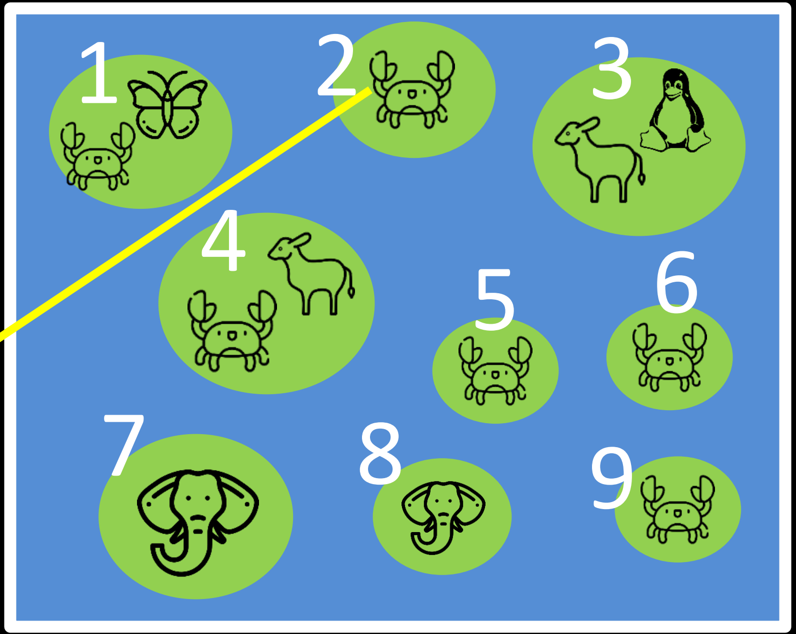
Min €: +1 +1 +1 +1 +1 +1 +1 +1 +1

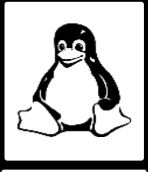
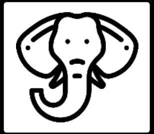
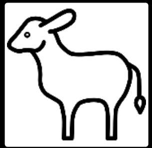


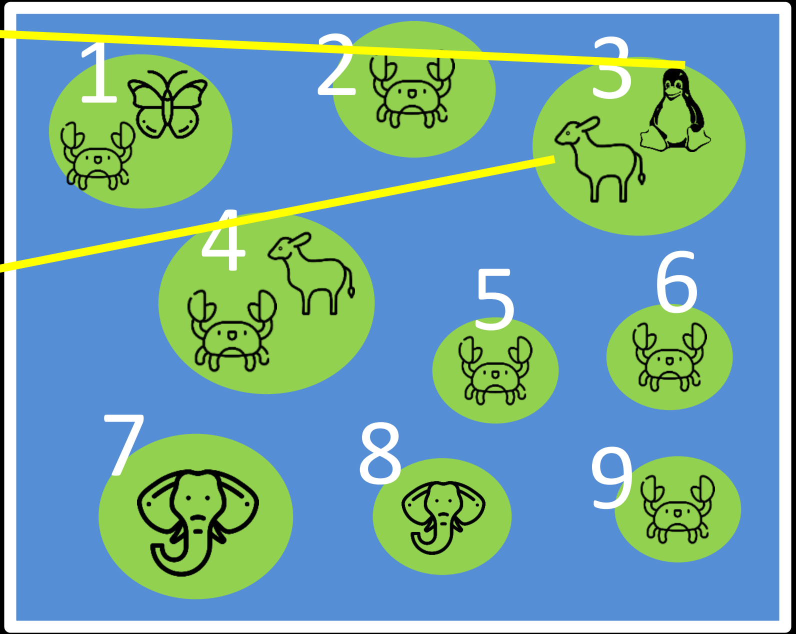
+1

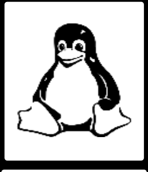
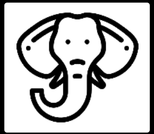
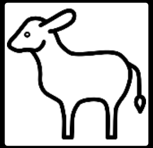
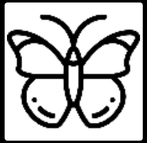
+1 +1

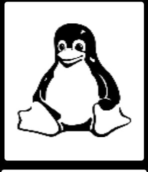
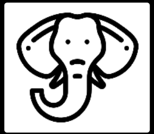
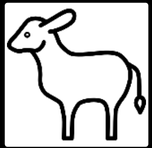
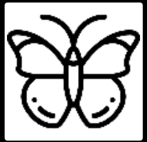
1 2 3 4 5 6 7 8 9



	1	2	3	4	5	6	7	8	9
Min €:	+1	+1	+1	+1	+1	+1	+1	+1	+1
			+1						
									
			+1						
	+1								
	+1	+1							



Min €:	+1	+1	+1	+1	+1	+1	+1	+1	+1
			+1						
							+1	+1	
			+1	+1					
	+1								
	+1	+1		+1	+1	+1			+1
	1	2	3	4	5	6	7	8	9

	1	2	3	4	5	6	7	8	9	
Min €:	+1	+1	+1	+1	+1	+1	+1	+1	+1	
			+1							≥ 1
							+1	+1		≥ 1
			+1	+1						≥ 1
	+1									≥ 1
	+1	+1		+1	+1	+1			+1	≥ 1

Min €: +1 +1 +1 +1 +1 +1 +1 +1 +1

GUROBI OPTIMIZATION

COMPUTATIONAL INFRASTRUCTURE FOR OPERATIONS RESEARCH

COIN|OR

IBM CPLEX

+1 +1 +1 +1 +1 +1

1	2	3	4	5	6	7	8	9

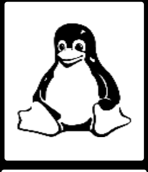
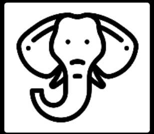
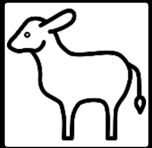
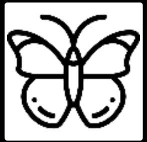
≥ 1

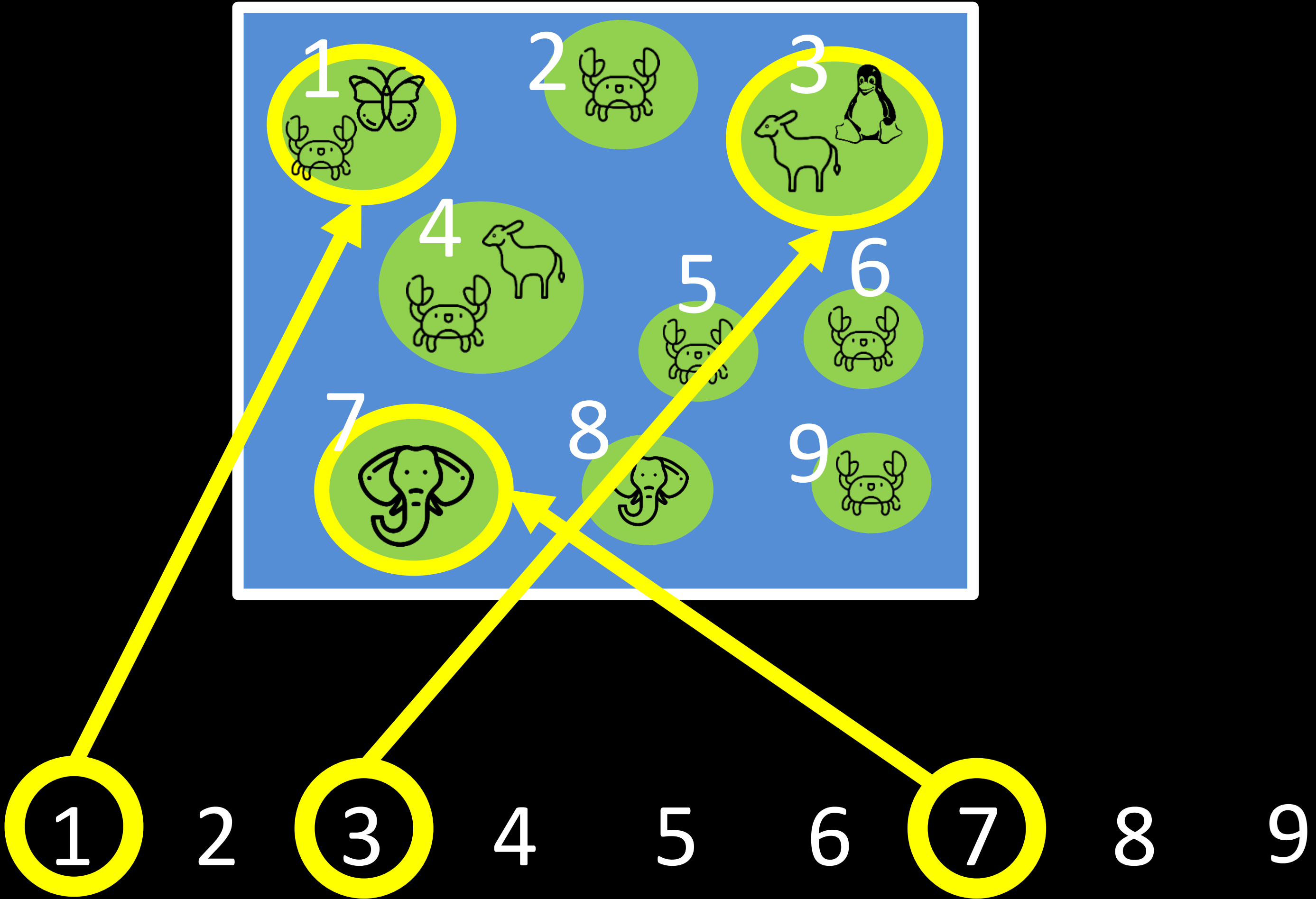
≥ 1

≥ 1

≥ 1

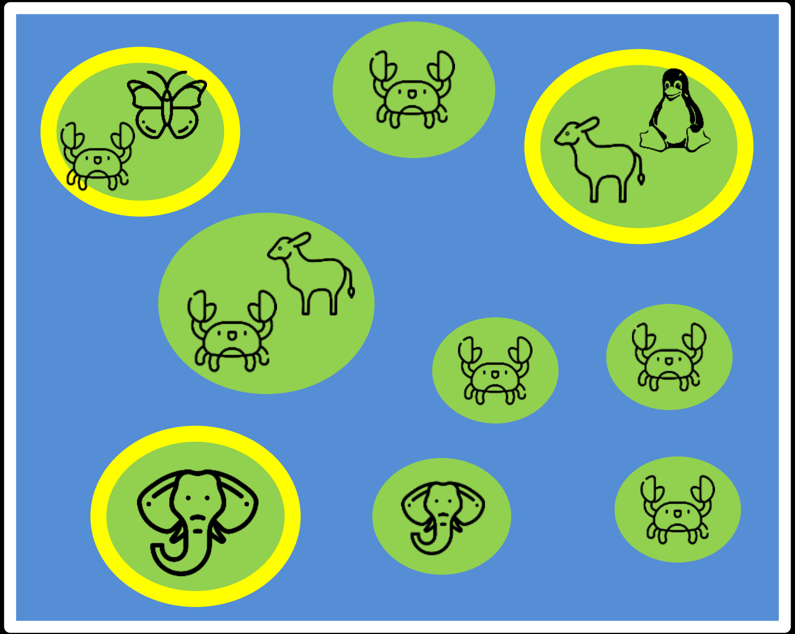
≥ 1

Min €:	+1	+1	+1	+1	+1	+1	+1	+1	+1	
			+1							≥ 1
							+1	+1		≥ 1
			+1	+1						≥ 1
	+1									≥ 1
	+1	+1		+1	+1	+1			+1	≥ 1
	1	2	3	4	5	6	7	8	9	

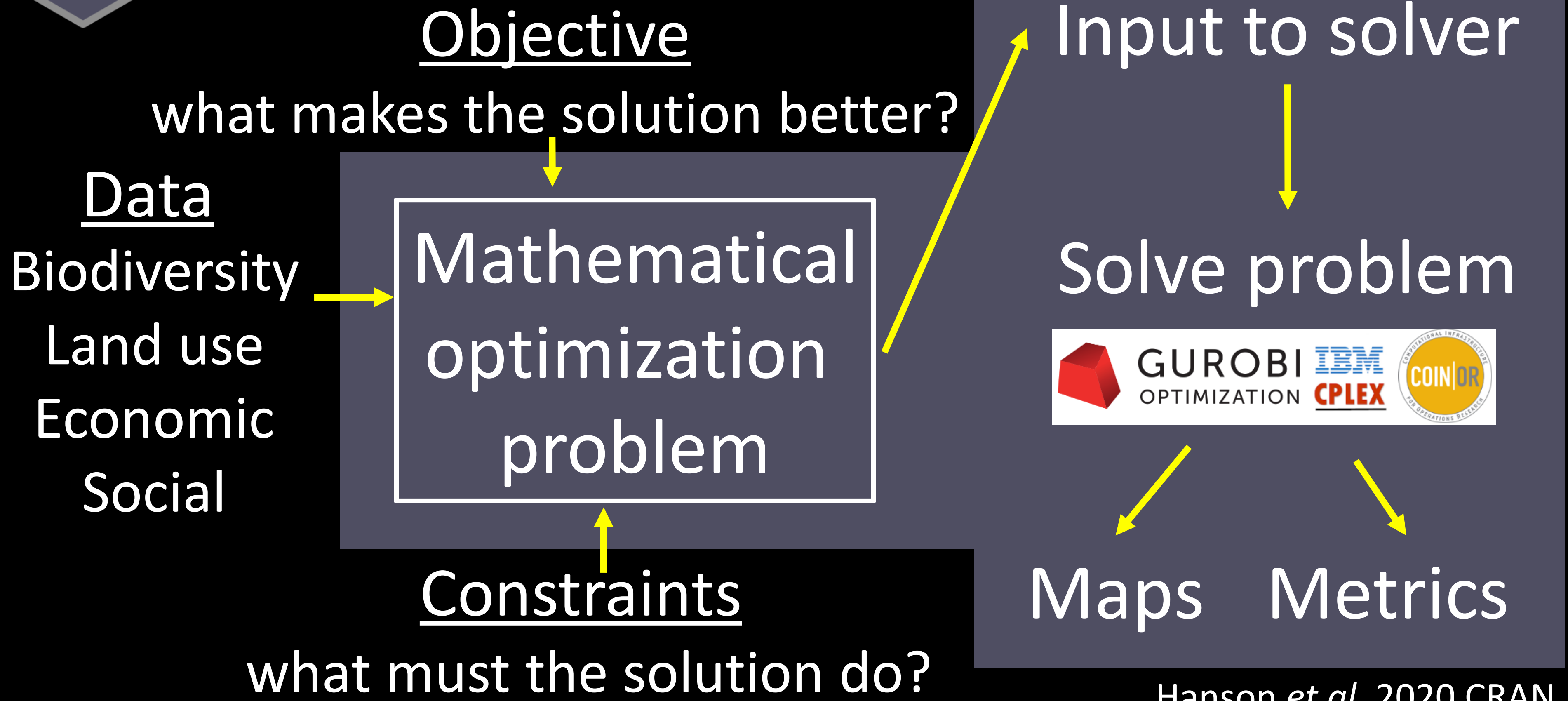

1
2
3
4
5
6
7
8
9
1 2 3 4 5 6 7 8 9

Reserve selection as optimization

- <u>Goal</u>: conserve biodiversity
- <u>Objective</u>: min. # of islands
- <u>Constraints</u>: sufficient habitat for each species
- <u>Decisions</u>: create a reserve on an island or not?



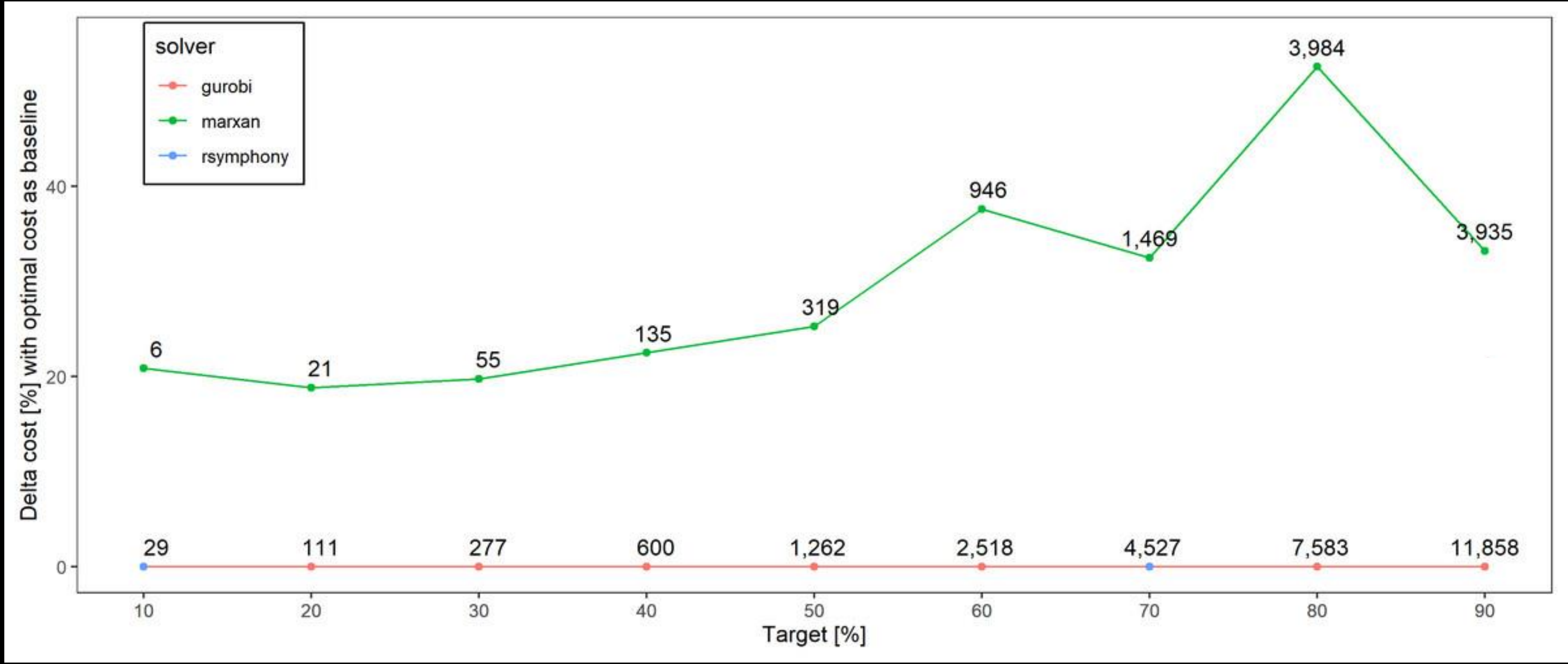
prioritizr R package



Hanson *et al.* 2020 CRAN

Better solutions

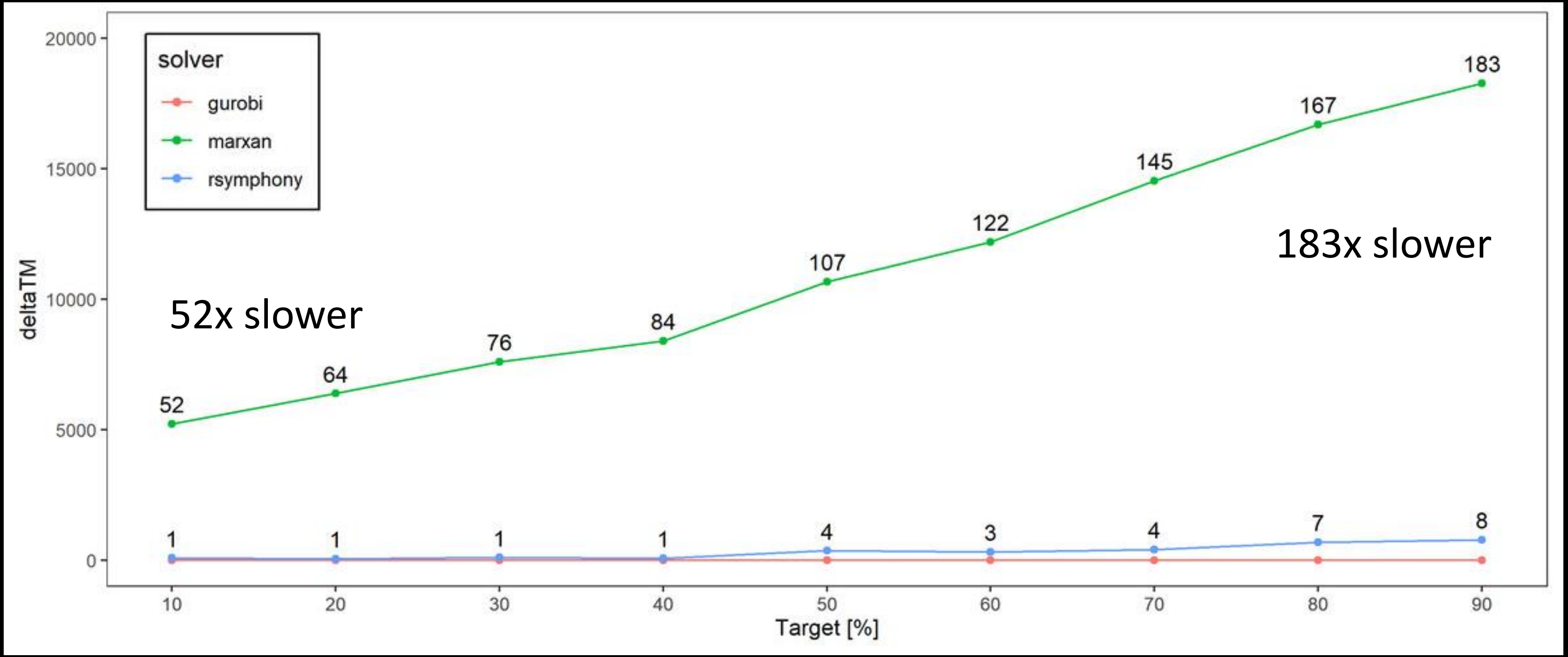
Better solution quality



Schuster *et al.* 2020 PeerJ

Faster too!

Slower run time



Schuster *et al.* 2020 PeerJ

Project prioritization

Project data

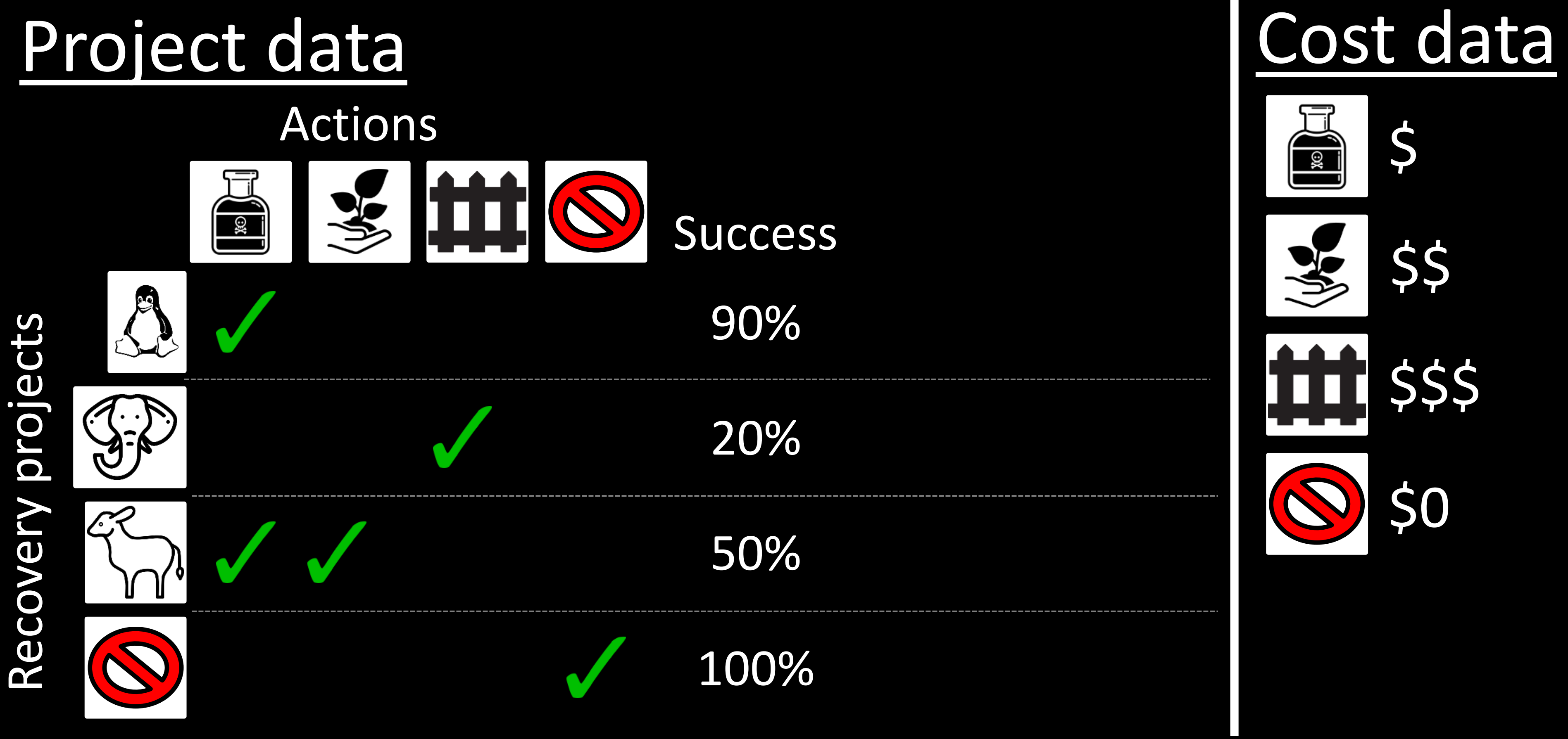
Recovery projects

Project prioritization

Project data

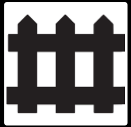
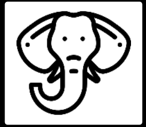
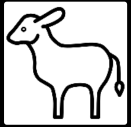
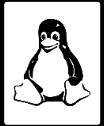
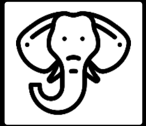
Cost data

Project prioritization

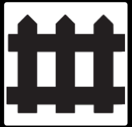


Project prioritization

Project data

Recovery projects	Actions				Success	Persistence (%)		
								
	✓				90%	95%		
			✓		20%		10%	
	✓	✓			50%			70%
				✓	100%	40%	9%	65%

Cost data

	$
	$$
	$$$$$
	$0

Project prioritization

Project data

Recovery projects

	Actions				Success	Persistence (%)		
	✓				90%	95%		
			✓		20%		10%	
	✓	✓			50%			70%
				✓	100%	40%	9%	65%

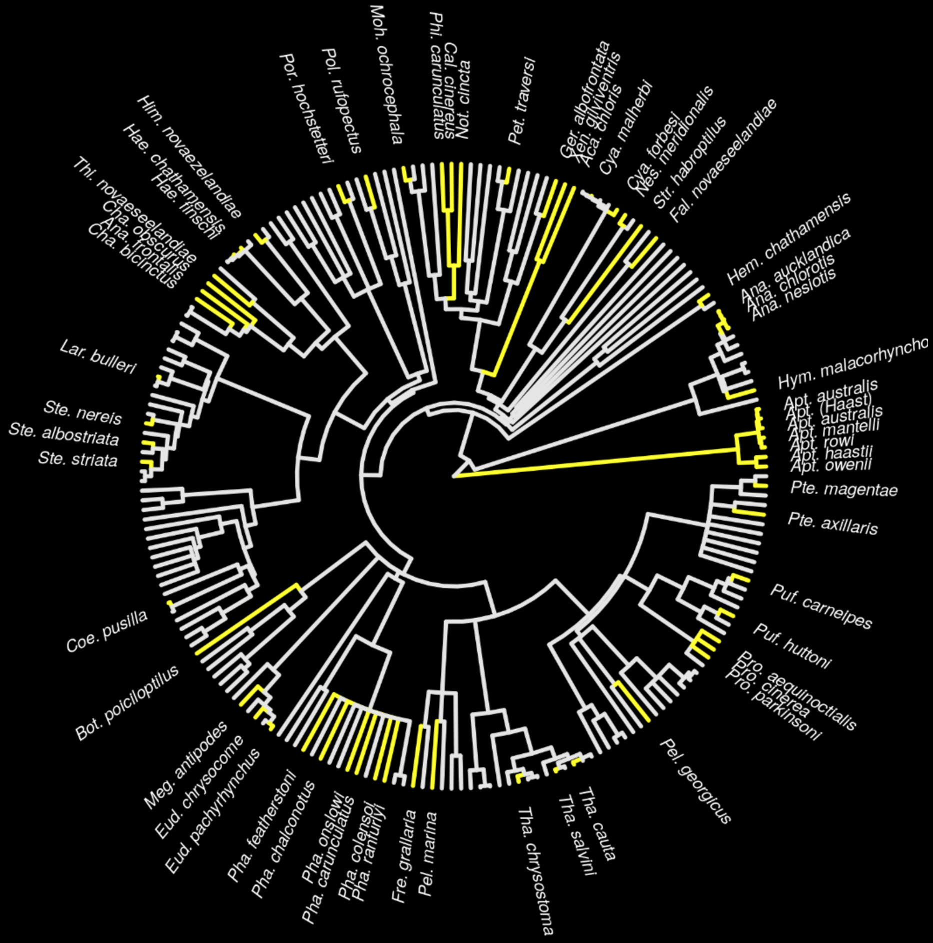
Cost data

	$
	$$
	$$$
	$0

Budget $$$

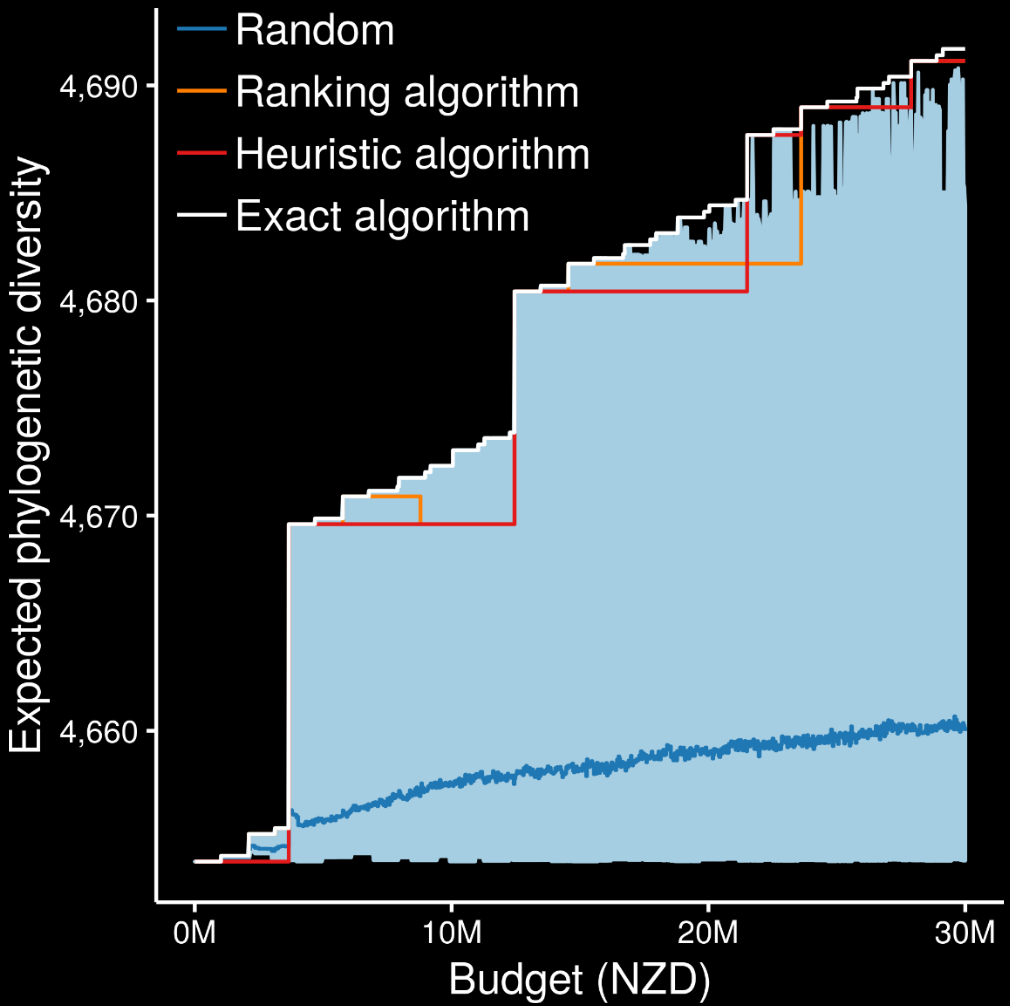
New Zealand case study

- Projects for 62 imperilled bird species
- 1,218 different actions
- Many actions shared between projects for different species
- oppr R package on CRAN



Hanson *et al.* (2019) Methods Ecol Evol , DOI:10.1111/2041-210X.13264

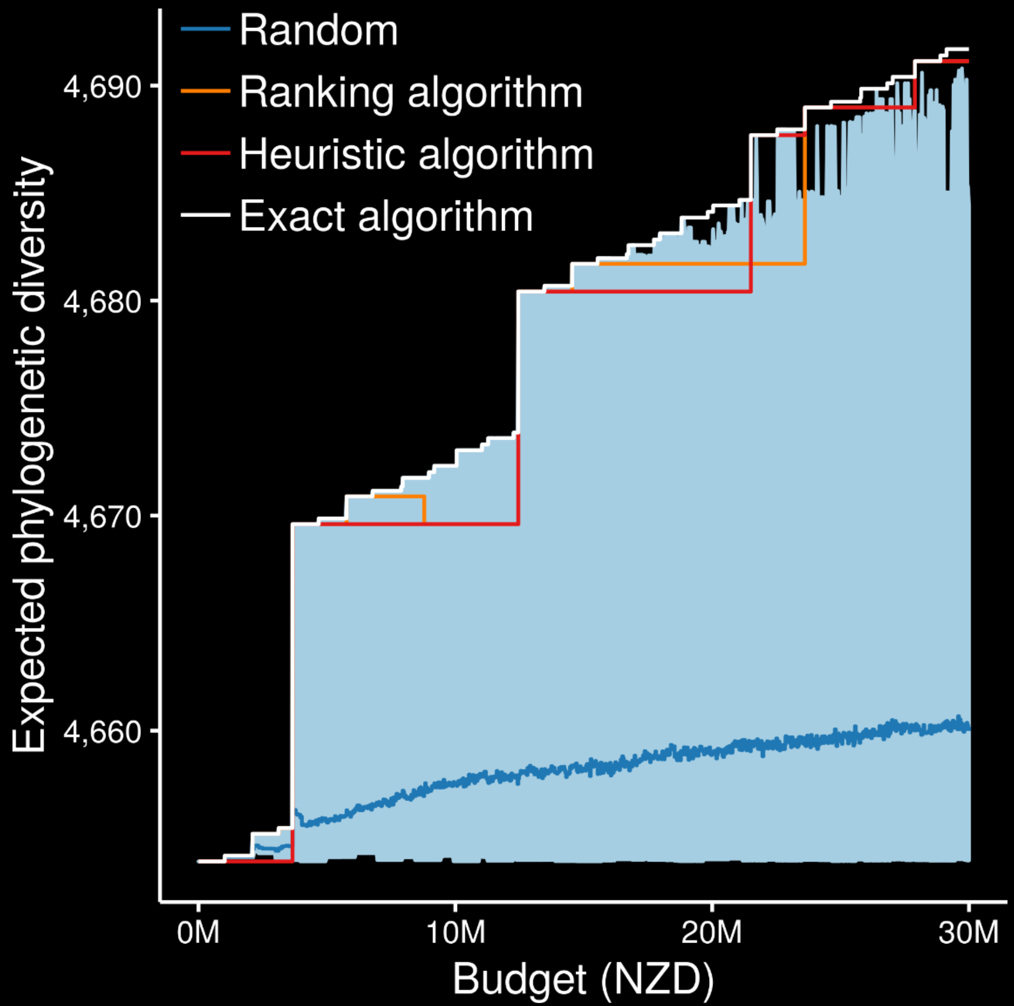
- Exact algorithms always generated the best prioritizations



Hanson *et al.* (2019) Methods Ecol Evol , DOI:10.1111/2041-210X.13264

- Ranking and heuristic algorithms sometimes produced optimal prioritizations

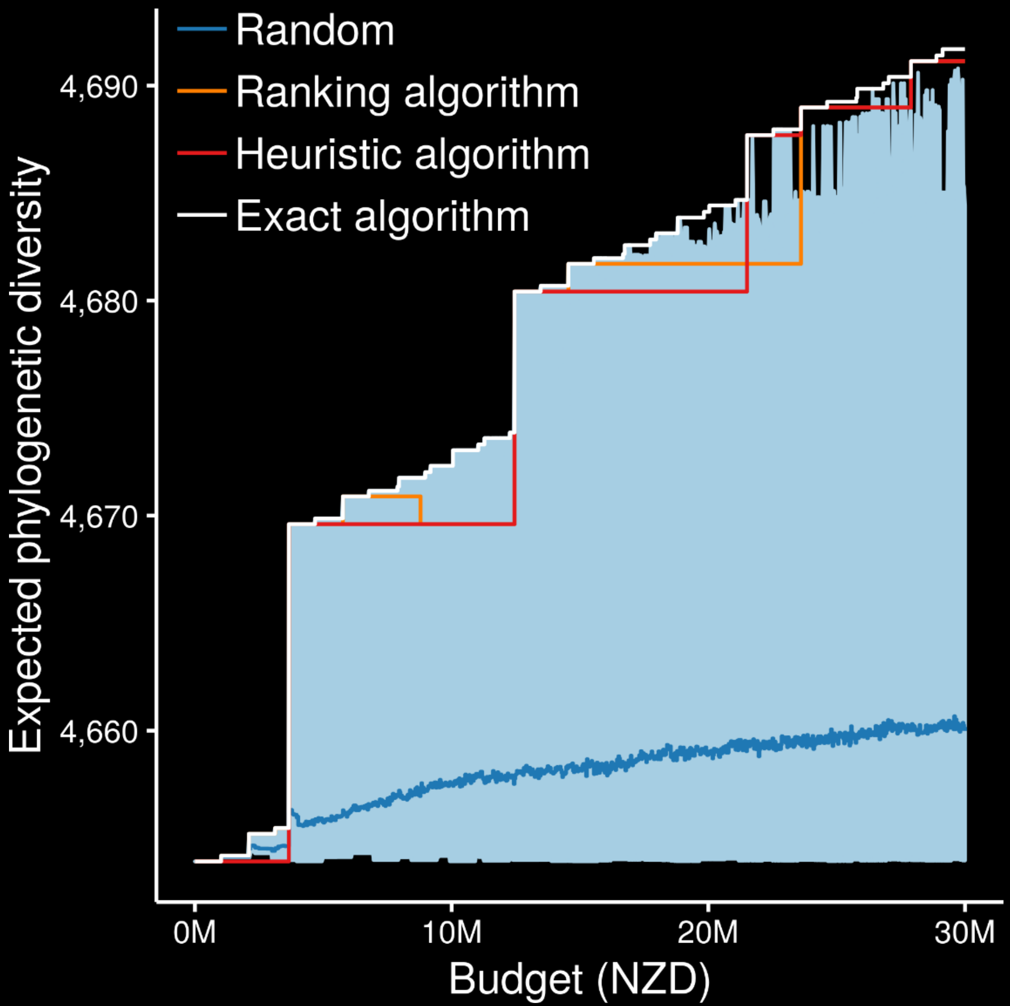
- Ranking and heuristic algorithm sometimes produced sub-optimal prioritizations



Hanson *et al.* (2019) Methods Ecol Evol , DOI:10.1111/2041-210X.13264

- Ranking and heuristic algorithms sometimes produced worse prioritizations than randomly allocating funds

- Ranking and heuristic algorithms often had large amounts of unspent funding, meaning less guidance for decision making



Hanson *et al.* (2019) Methods Ecol Evol , DOI:10.1111/2041-210X.13264

jeffrey.hanson@uqconnect.edu.au

github.com/jeffreyhanson

jeffrey-hanson.com

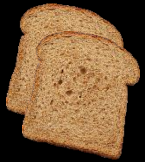
Case-study: Sandwich

- Goal: Best sandwich experience
- Objective: Maximize taste
- Constraints:
  - must have 2 slices of bread,
  - total calories for the meal must not be too high,
  - total cost of ingredients must not exceed budget
- Decisions: which ingredients shall I use?


Blue Plate
REAL
MAYONNAISE

Maximize taste:

+1 +50 +20 +40 +3 +1 +8 -1





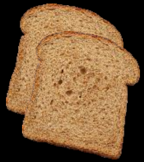


Maximize taste:

+1	+50	+20	+40	+3	+1	+8	-1		

Constraint 1: must have 2 bread slices

+1	+0	+0	+0	+0	+0	+0	+0	|	=1







Maximize taste:

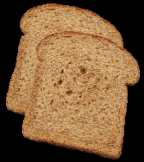
+1	+50	+20	+40	+3	+1	+8	-1	

Constraint 1: must 2 bread slices

+1	+0	+0	+0	+0	+0	+0	+0	=1

Constraint 2: calories (C per serving)

+70	+145	+380	+299	+15	+680	+24	43	≤235







Maximize taste:

+1	+50	+20	+40	+3	+1	+8	-1	
Constraint 1: must 2 bread slices								
+1	+0	+0	+0	+0	+0	+0	+0	=1
Constraint 2: calories (C per serving)								
+70	+145	+380	+299	+15	+680	+24	43	≤235
Constraint 3: budget (€ per serving)								
+0.21	+2.5	+0.12	+0.84	+0.04	+0.04	+0.08	+0.07	≤2.75

Maximize taste:

+1	+50	+20	+40	+3	+1	+8	[illegible]	

Constraint 1: must 2 bread slices

+1	+0	+0	+0	+0	+0	+0	+[illegible]	=1

Constraint 2: calories (C per serving)

+70	+145	+380	+299	+15	+680	+24	[illegible]	≤235

Constraint 3: budget (€ per serving)

+0.21	+2.5	+0.12	+0.84	+0.04	+0.04	+0.08	+[illegible]7	≤2.75

Maximize taste:

Constraint 1: must 2 bread slices

Constraint 2: calories (C per serving)

Constraint 3: budget (€ per serving)

Taste	+1	+50	+	+	+3		+8		
Constraint 1	+1	+0			+0	-	+0	-	=1
Constraint 2	+70	+145	+3 0	+2 9	+15	+6 0	+24	2	≤235
Constraint 3	+0.21	+2.5	+0 2	+0 4	+0.04	+0 4	+0.08	+0 7	≤2.75

Maximize taste:

+1 +50 [illegible] [illegible] +3 [illegible] +8 [illegible]

Constraint 1: must 2 bread slices

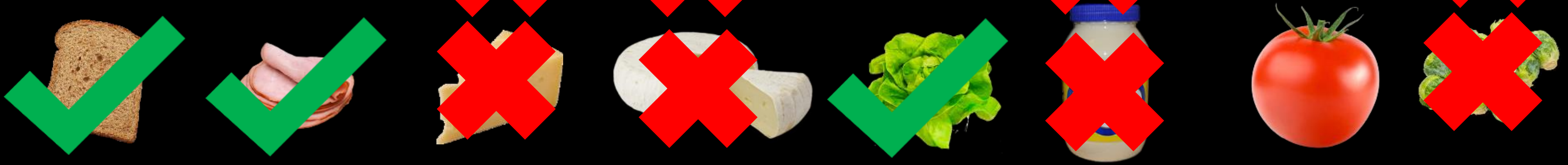
+1 +0 [illegible] [illegible] +0 [illegible] +0 [illegible] =1

Constraint 2: calories (C per serving)

+70 +145 [illegible] [illegible] +15 [illegible] +24 [illegible] ≤235

Constraint 3: budget (€ per serving)

+0.21 +2.5 [illegible] [illegible] +0.04 [illegible] +0.08 [illegible] ≤2.75



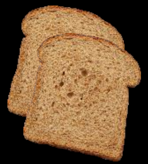
Maximize taste:

+1	+50	+20	+40	+3	+1	+8	+20

Constraints 2--3

Constraint 4: mayo + lettuce combination

+0	+0	+0	+0	+0	+1	+0	-1	≥ 0
+0	+0	+0	+0	+1	+0	+0	-1	≥ 0







Performance for phylogenetic diversity

- Exact algorithms generated prioritizations within a feasible period of time
- Heuristic and ranking algorithms are faster
- It might be worth waiting worth ~15 seconds if it means you could potentially save millions of $$$

Table: Average run time for generating prioritizations under different budgets

Algorithm	Average run time (seconds)
Exact	16.2
Heuristic	1.19
Ranking	1.5

NB. Heuristic and ranking algorithms coded in C++ for performance, so if you're using R, these timings would be much higher

Hanson *et al.* (2019) Methods Ecol Evol , DOI:10.1111/2041-210X.13264

Performance for total expected persistence

- Simulated dataset
  - 40 conservation projects
  - 80 management actions
  - 50 species
- Performance of exact algorithm solver is pretty much the same for simple problems

Table: Average run time for generating prioritizations under different budgets

Algorithm	Average run time (seconds)
Exact	1.6
Heuristic	1.48
Ranking	1.18

NB. Heuristic and ranking algorithms coded in C++ for performance, so if you're using R, these timings would be much higher

Hanson *et al.* (2019) Methods Ecol Evol , DOI:10.1111/2041-210X.13264