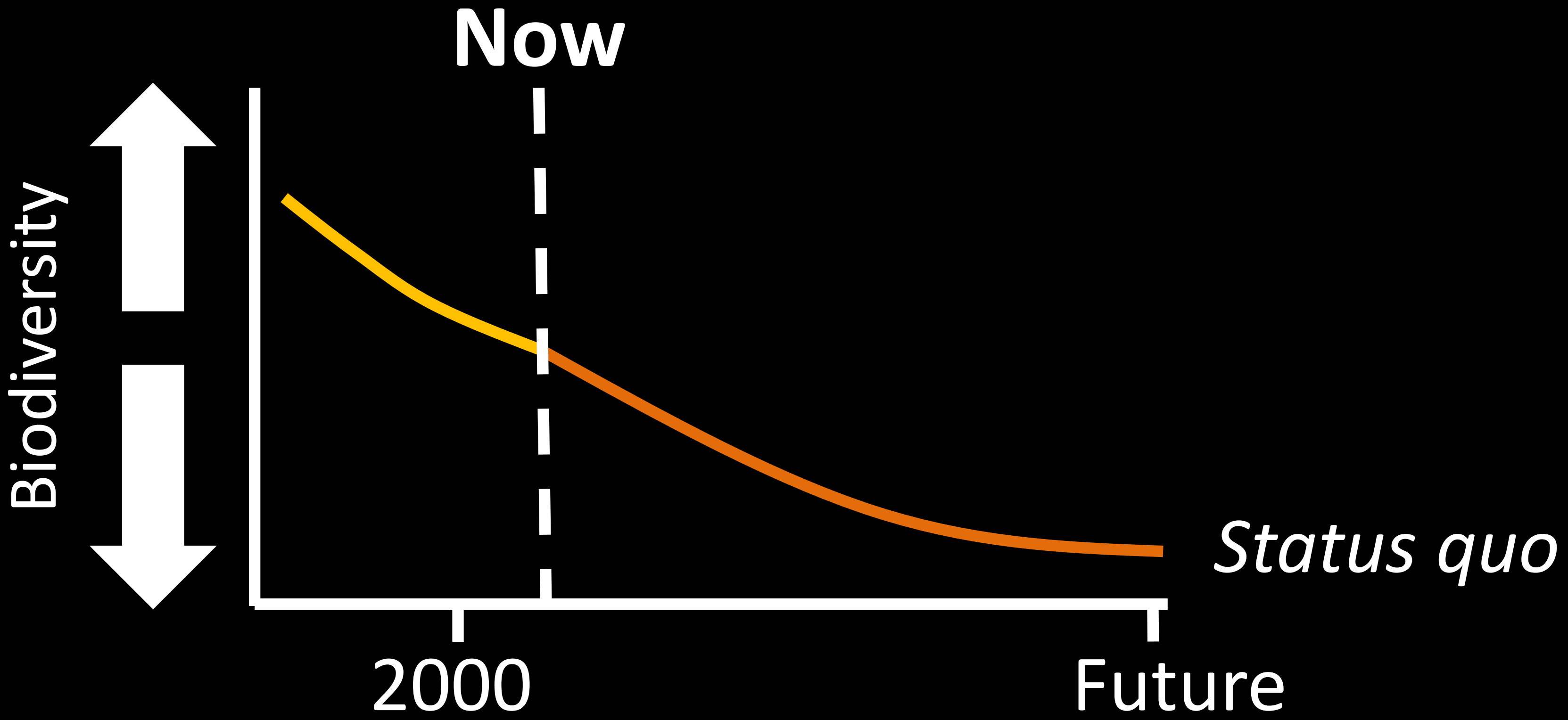
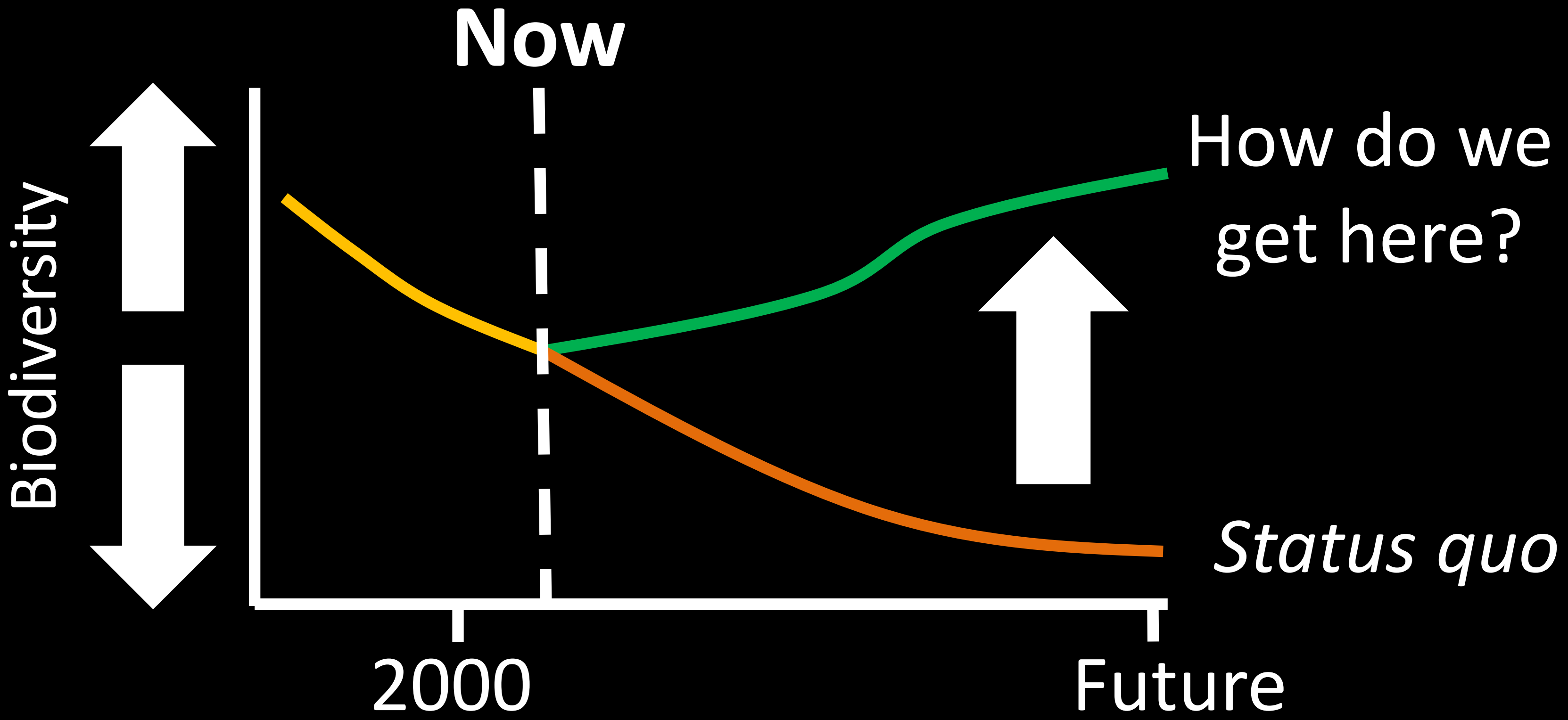
Conservation decisions with exact algorithm solvers

Jeffrey Hanson

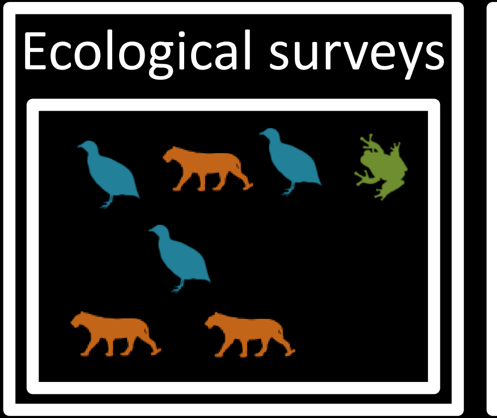
✉ jeffrey.hanson@uqconnect.edu.au

🌐 jeffrey-hanson.com

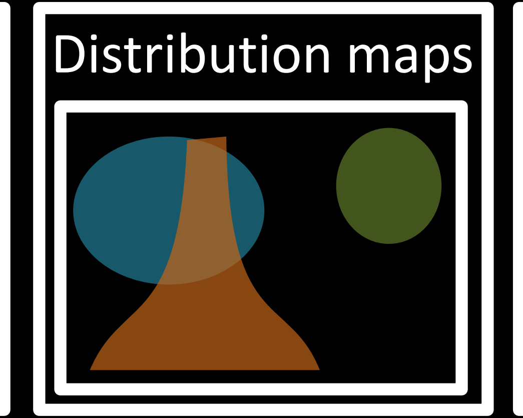

Now
Biodiversity
Status quo
2000
Future


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

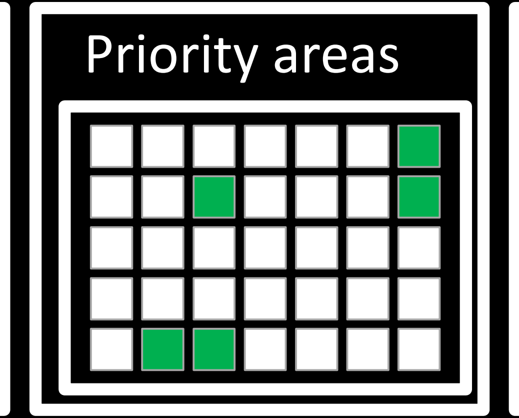
How do we bend the curve?



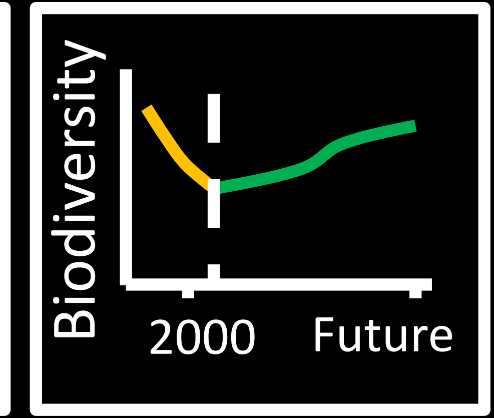
Ecological surveys



Distribution maps



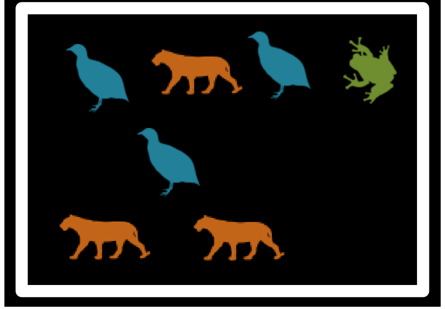
Priority areas



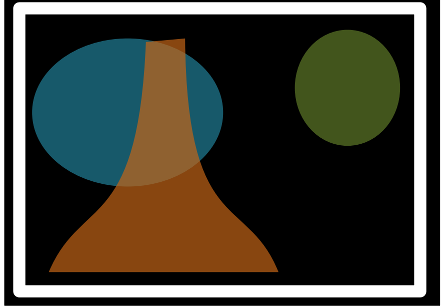
Data → Information → Plan → Outcome

How do we bend the curve?

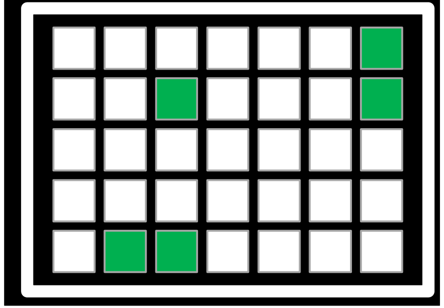
Ecological surveys

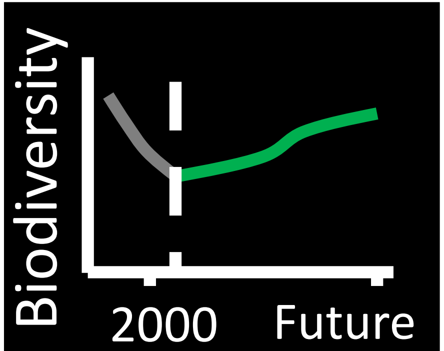


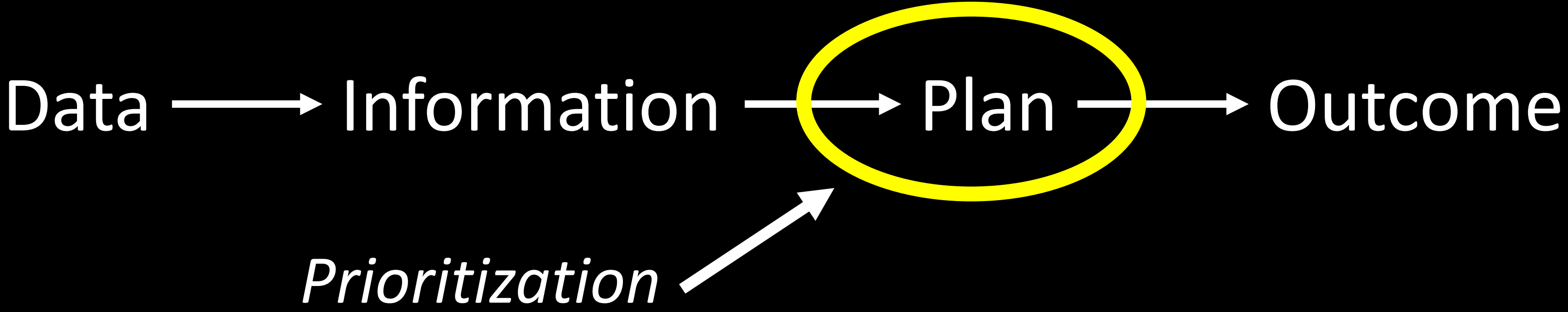
Distribution maps



Priority areas







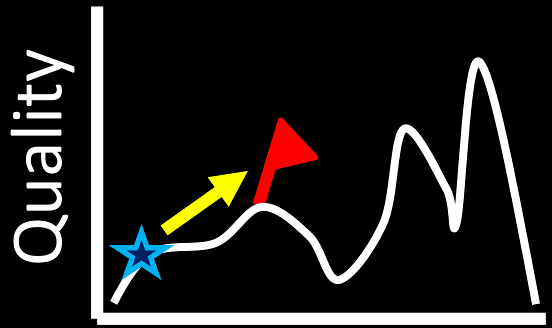
Data → Information → Plan → Outcome

Prioritization

Framing conservation as a decision science problem

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

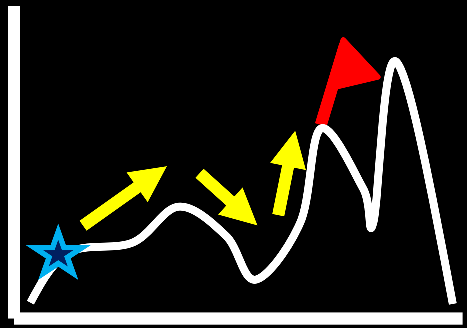
Heuristic algorithm



Different solutions

PPP

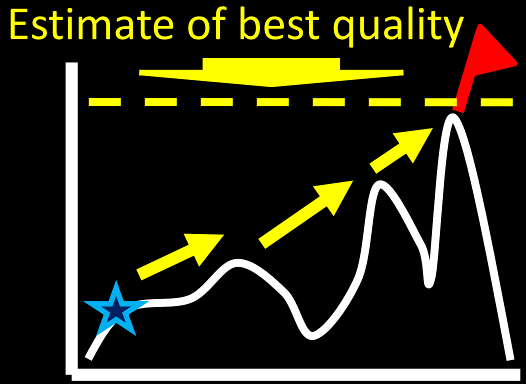
Meta-heuristic algorithms



Different solutions

MARXAN
conservation solutions

Exact algorithms



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

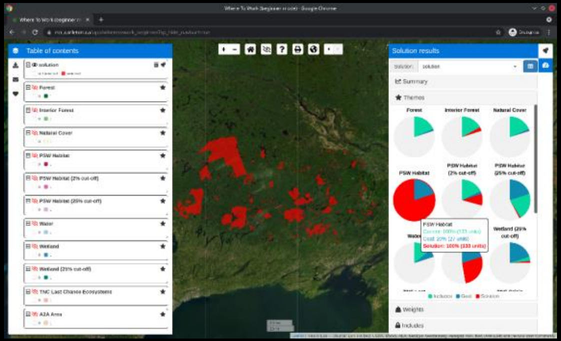
Exact algorithm solvers

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

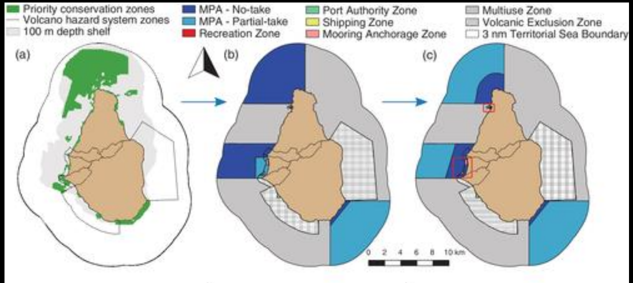
  - integer programming (IP/ILP) =
    binary/integer decision variables

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

Conservation examples



Nature Conservancy of Canada for land acquisition

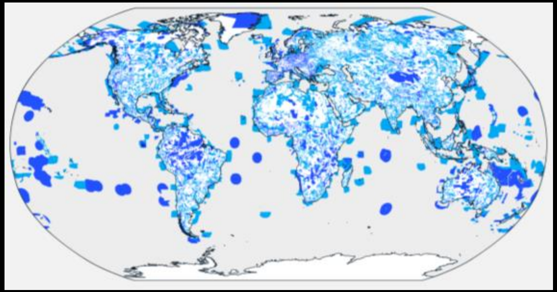


Priority conservation zones
Objectives met:
- 30% of each habitat protected
- 50% of total coral and fish species richness protected

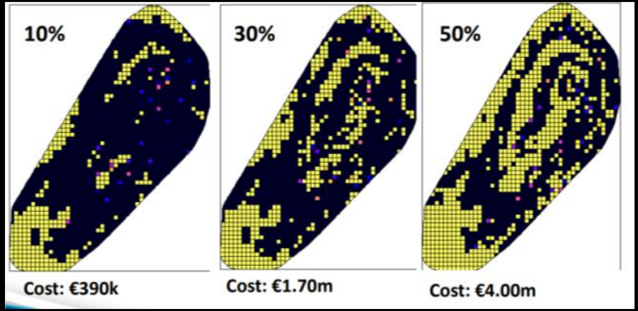
Marine Spatial Plan: Draft 1
Zones coverage:
- 4 habitats meet 30% protection goal, 3 meet 20% protection, 1 meets 10% protection
- 20% in no-take MPAs
- 13% in partial-take MPAs

Marine Spatial Plan: Draft 2 and Final Draft
Zones coverage:
- 4 habitats meet 30% protection goal, 3 meet 20% protection, 1 meets 10% protection
- 6% in no-take MPAs
- 27% in partial-take MPAs

Waitt Institute to help Government of Montserrat



McKinsey Consulting

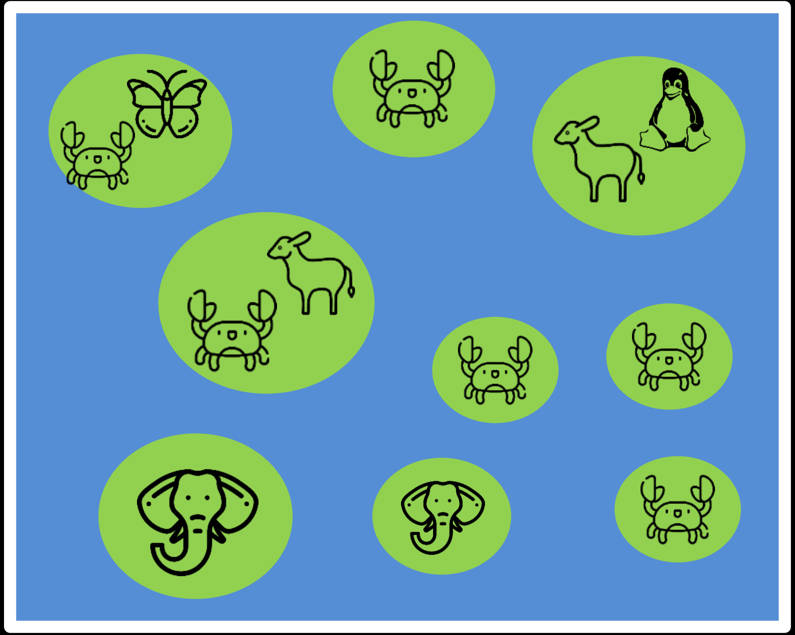


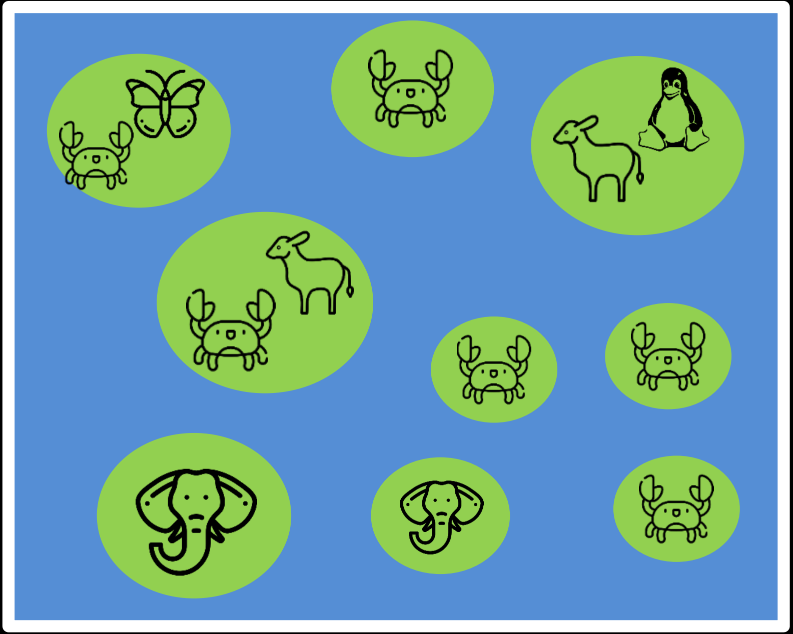
Scottish Government for marine spatial planning in Rockall Bank

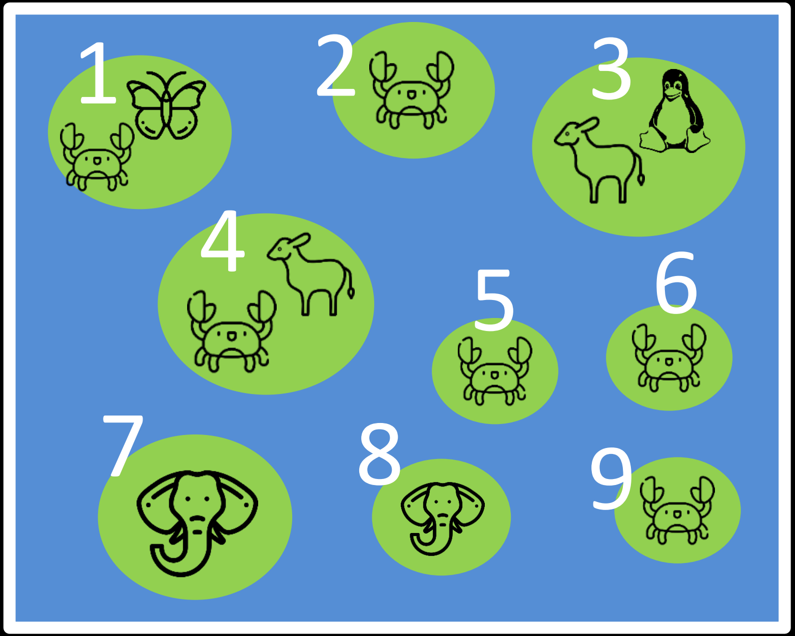
USGS to prioritize recovery areas in Hawaii

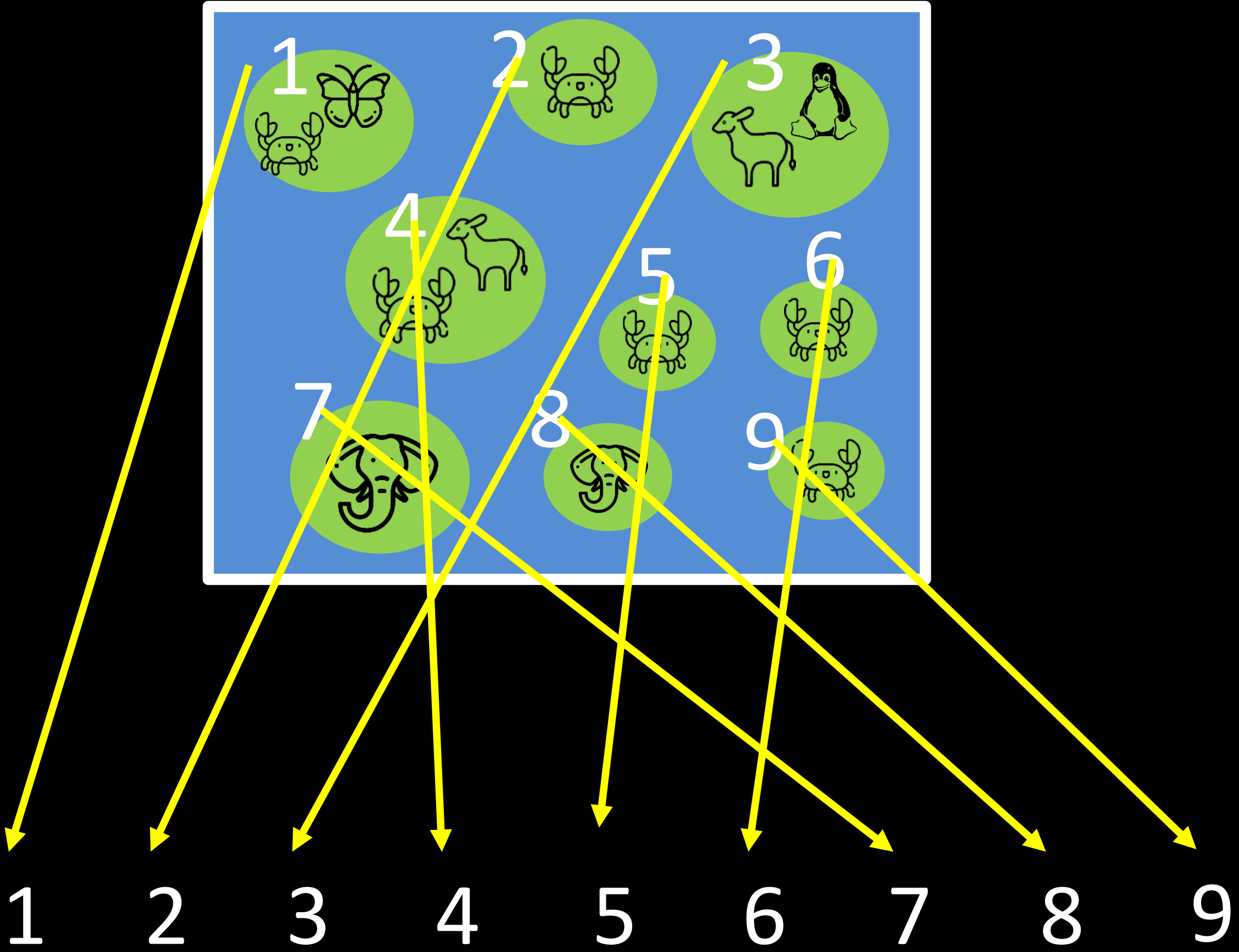
Reserve selection as optimization

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






1
2
3
4
5
6
7
8
9

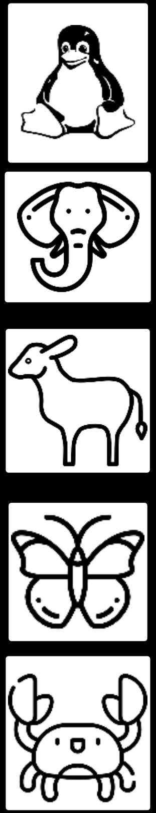

1
2
3
4
5
6
7
8
9
1
2
3
4
5
6
7
8
9

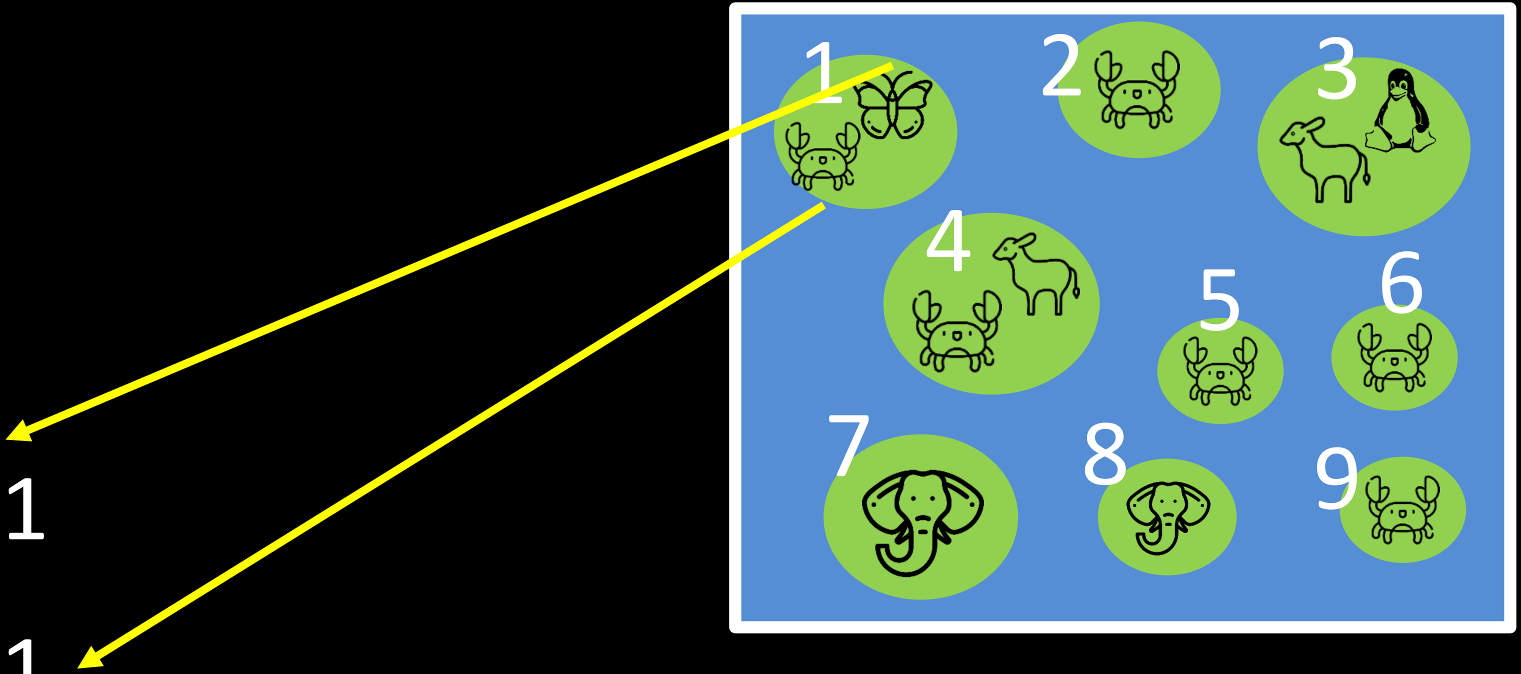
1 2 3 4 5 6 7 8 9

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

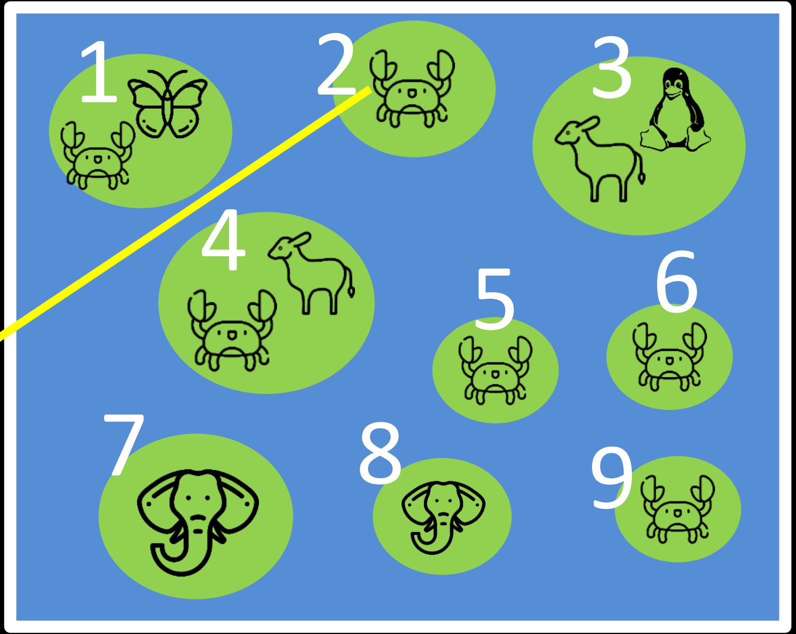
Min €: +1 +1 +1 +1 +1 +1 +1 +1 +1



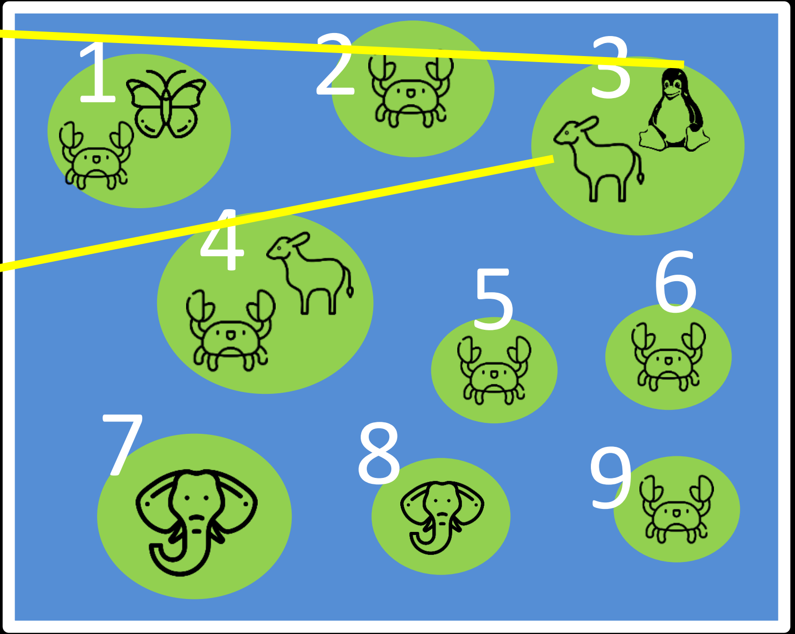


1 2 3 4 5 6 7 8 9

Min €:	+1	+1	+1	+1	+1	+1	+1	+1	+1
🐧									
🐘									
🦌									
🦋	+1								
🦀	+1	+1							
	1	2	3	4	5	6	7	8	9



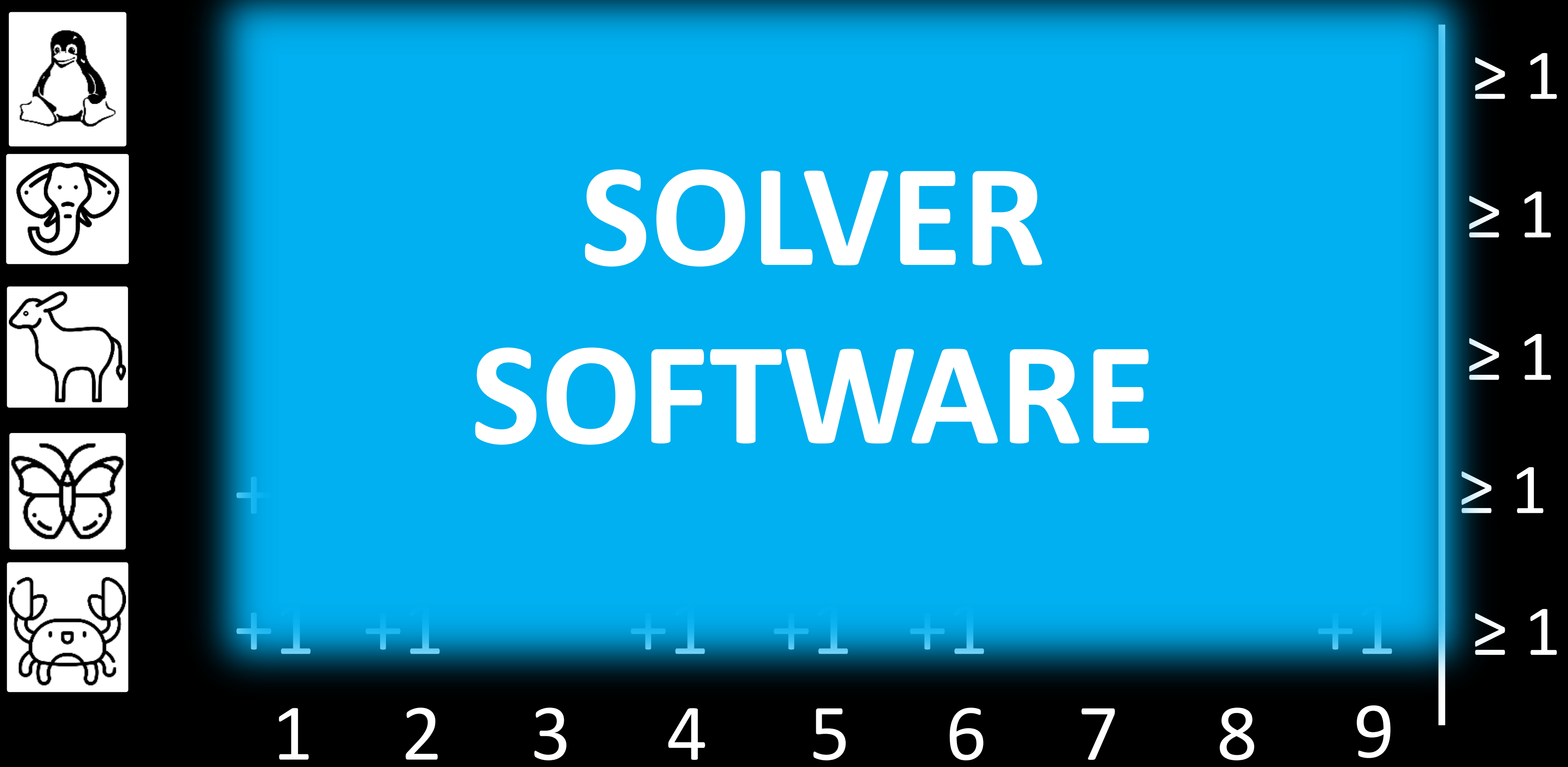
	1	2	3	4	5	6	7	8	9
Min €:	+1	+1	+1	+1	+1	+1	+1	+1	+1
🐧			+1						
🐘									
🐐			+1						
🦋	+1								
🦀	+1	+1							



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

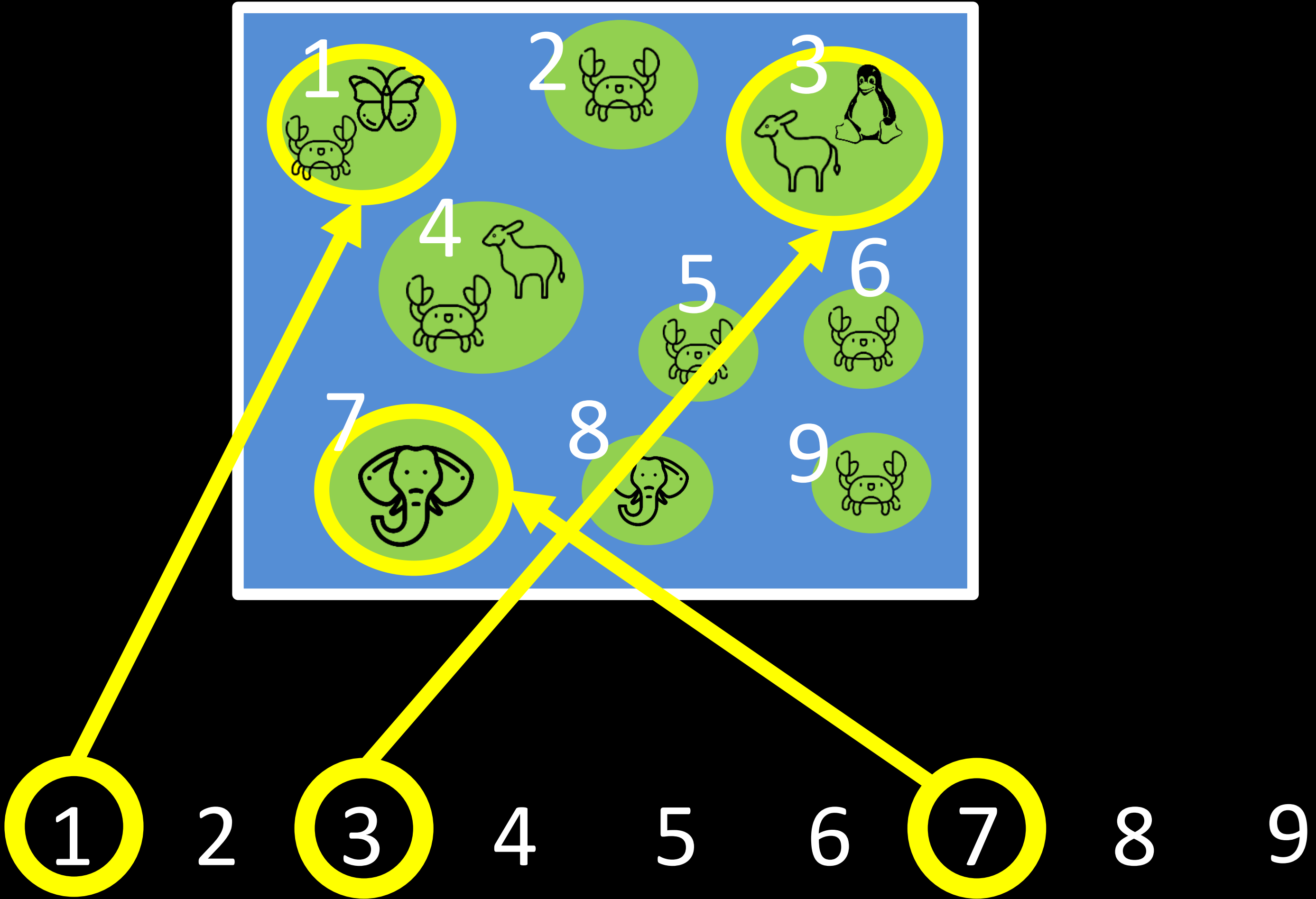
Min €:	+1	+1	+1	+1	+1	+1	+1	+1	+1	
🐧			+1							≥ 1
🐘							+1	+1		≥ 1
🐐			+1	+1						≥ 1
🦋	+1									≥ 1
🦀	+1	+1		+1	+1	+1			+1	≥ 1
	1	2	3	4	5	6	7	8	9	

Min €: +1 +1 +1 +1 +1 +1 +1 +1 +1



SOLVER SOFTWARE

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

Exact algorithm solvers

GUROBI
OPTIMIZATION

IBM
CPLEX

coin-or/
SYMPHONY

COMPUTATIONAL INFRASTRUCTURE
COIN|OR
FOR OPERATIONS RESEARCH

HiGHS

coin-or/Cbc

COIN-OR Branch-and-Cut solver

COMPUTATIONAL INFRASTRUCTURE
COIN|OR
FOR OPERATIONS RESEARCH

lp_solve

Cost and licensing

Commercial

GUROBI
OPTIMIZATION

IBM
CPLEX

Open source

coin-or/
SYMPHONY

COMPUTATIONAL INFRASTRUCTURE
COIN|OR
FOR OPERATIONS RESEARCH

HiGHS

coin-or/Cbc

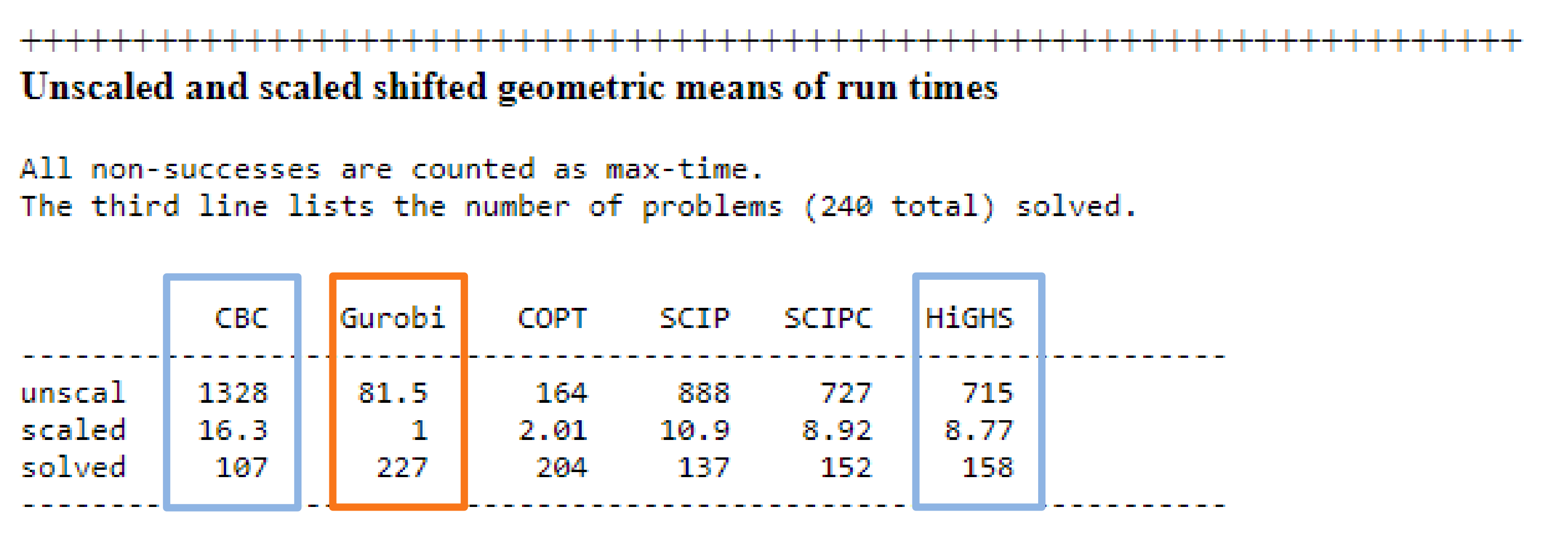
COIN-OR Branch-and-Cut solver

COMPUTATIONAL INFRASTRUCTURE
COIN|OR
FOR OPERATIONS RESEARCH

lp_solve

General performance

- Modern benchmarks (April 2023)
- CBC vs Gurobi vs HiGHS



```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

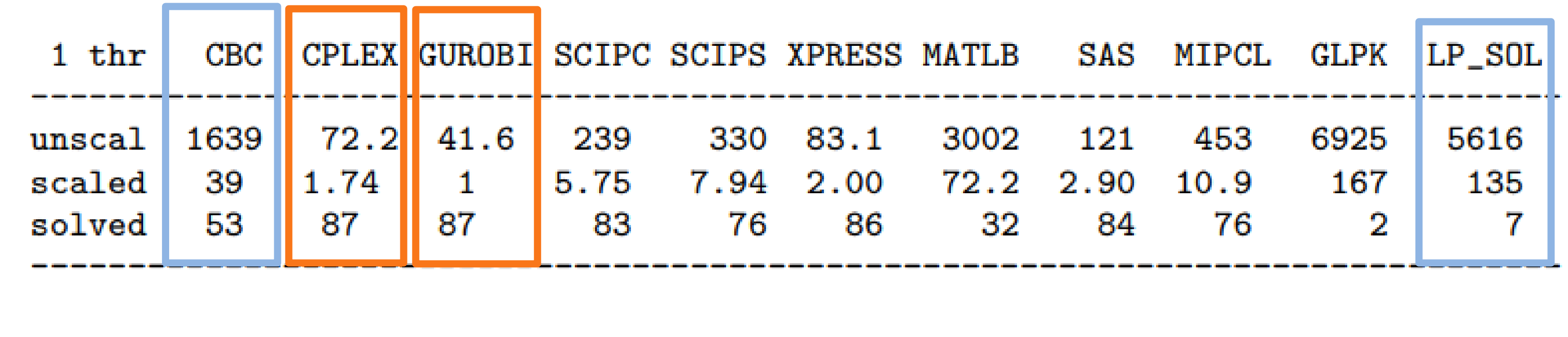
Unscaled and scaled shifted geometric means of run times

```
All non-successes are counted as max-time.
The third line lists the number of problems (240 total) solved.

              CBC     Gurobi      COPT      SCIP     SCIPC     HiGHS
-------------------------------------------------------------------------
unscal       1328       81.5       164       888       727       715
scaled       16.3          1      2.01      10.9      8.92      8.77
solved        107        227       204       137       152       158
-------------------------------------------------------------------------
```

General performance

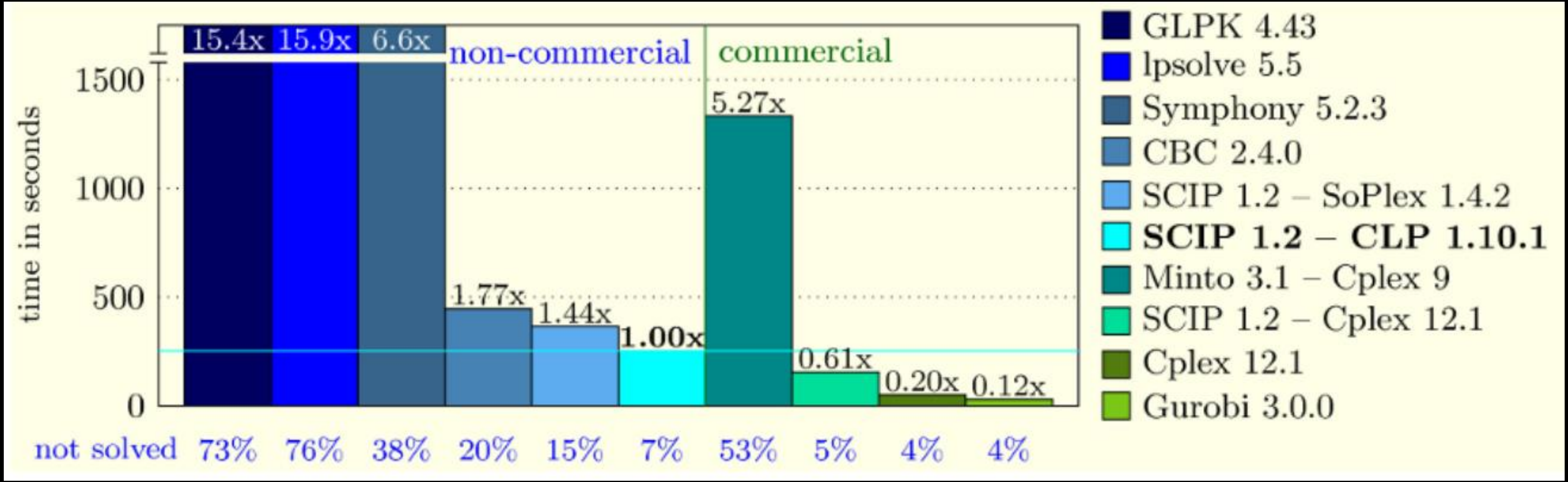
- Historical benchmarks (Oct 2018)
- See how CPLEX and lp_solve compare



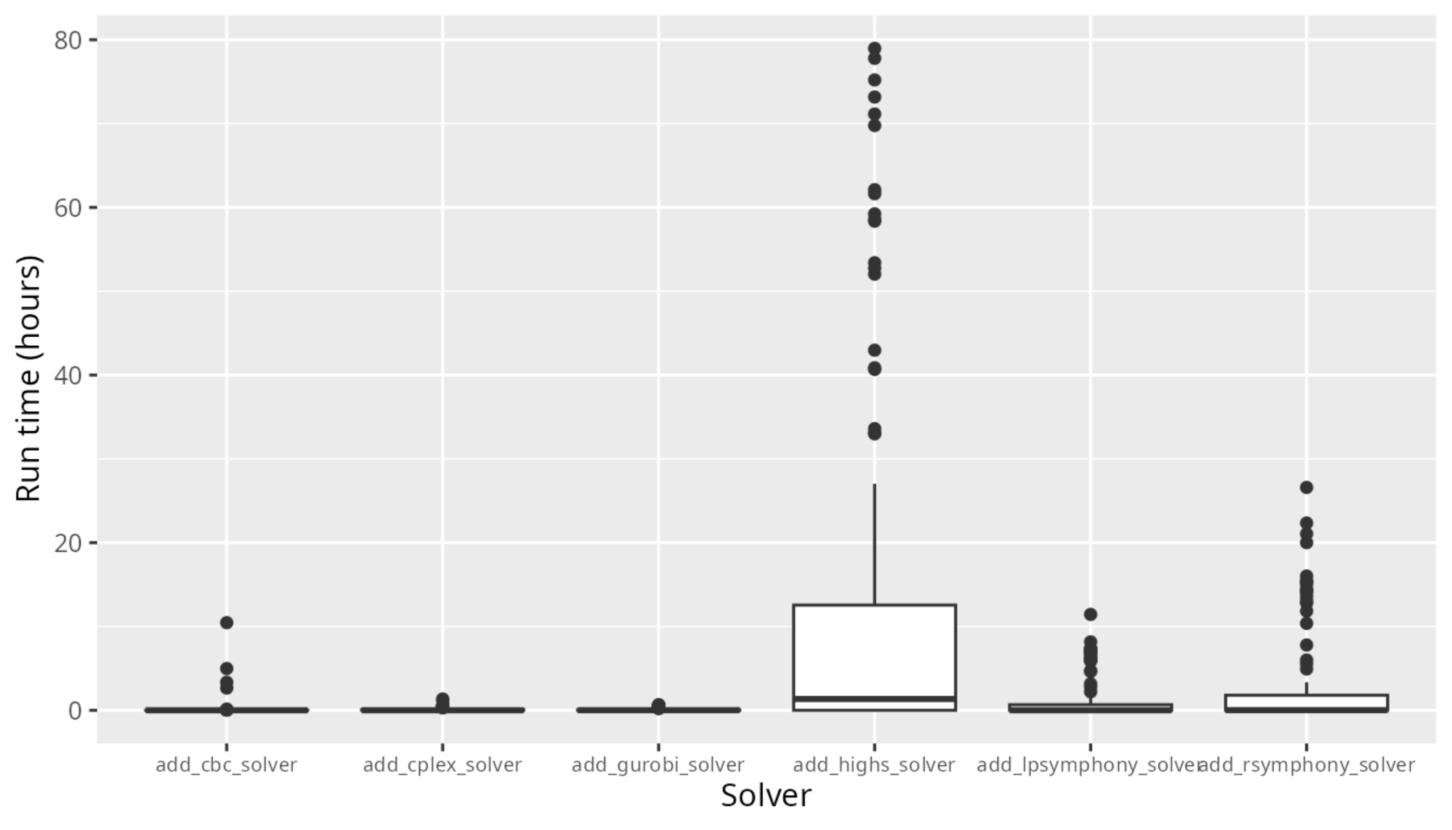
1 thr	CBC	CPLEX	GUROBI	SCIPC	SCIPS	XPRESS	MATLB	SAS	MIPCL	GLPK	LP_SOL
unscal	1639	72.2	41.6	239	330	83.1	3002	121	453	6925	5616
scaled	39	1.74	1	5.75	7.94	2.00	72.2	2.90	10.9	167	135
solved	53	87	87	83	76	86	32	84	76	2	7

General performance

- Historical benchmarks (2010)
- See how Symphony compares



Performance for a conservation problem



Running the software programmatically

- Gurobi (gurobi R package)
- IBM CPLEX (cplexAPI and Rcplex R packages
- Symphony (lpsymphony and Rsymphony R packages)
- CBC (rcbc R package)
- lp_solve (lpsolveAPI R package)
- HiGHS (highs R package)

Software interfaces are available for other programming languages too!

Recommendations

1. Gurobi is the best solver, and a special academic license is freely available

2. CBC and HiGHS are the best open source solvers

3. All three solvers have fairly similar R package interface, so running the same analyses with different solvers is relatively straight forward

jeffrey.hanson@uqconnect.edu.au

github.com/prioritizr/prioritizr

prioritizr.net

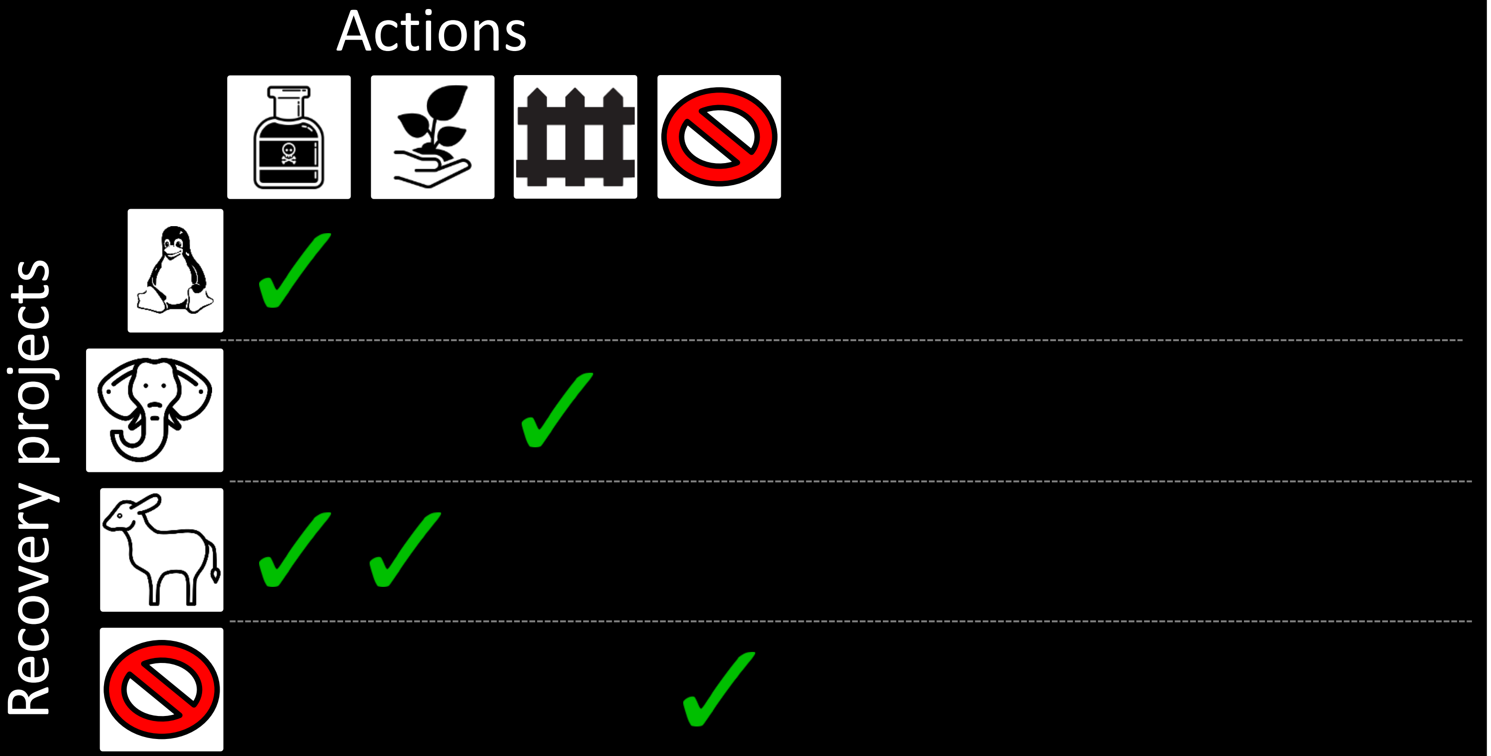
Project prioritization

Project data

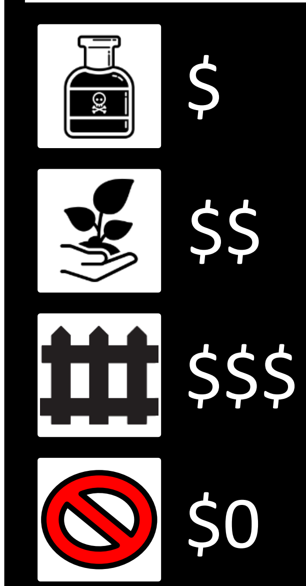
Recovery projects

Project prioritization

Project data

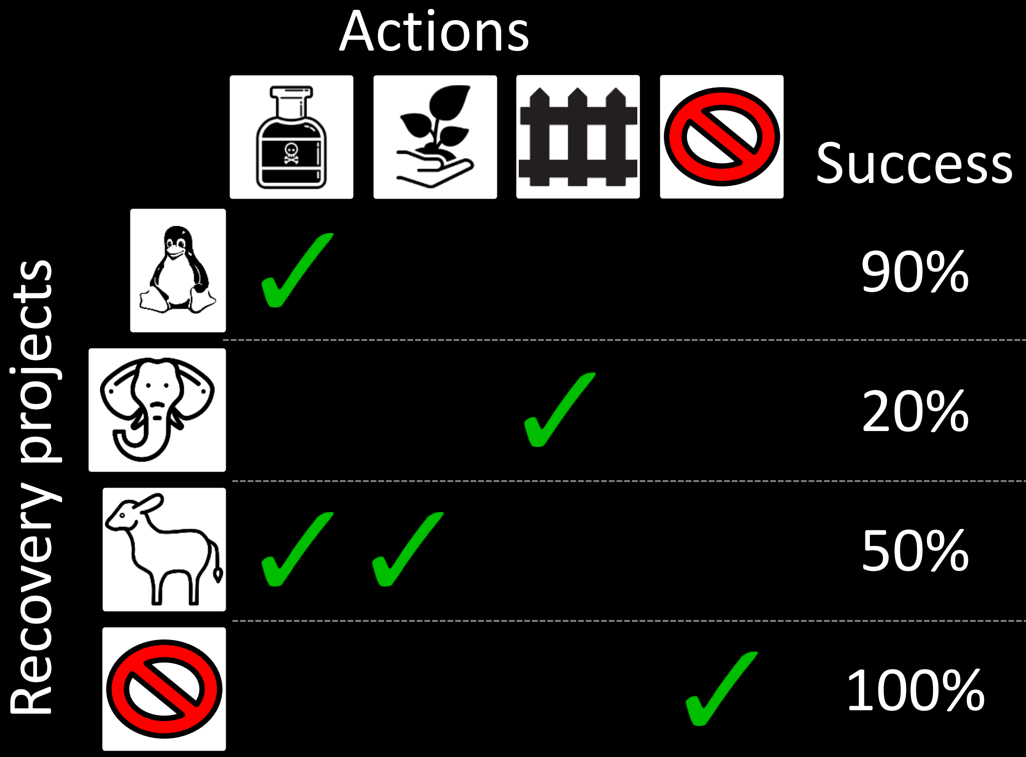


Cost data



Project prioritization

Project data

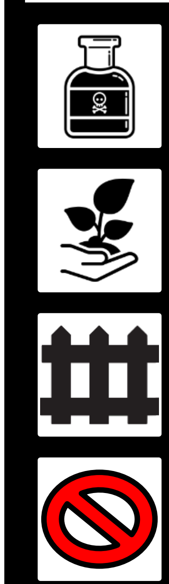


Actions

Recovery projects

				🚫	Success
	✓				90%
			✓		20%
	✓	✓			50%
🚫				✓	100%

Cost data



	$
	$$
	$$$
🚫	$0

Project prioritization

Project data

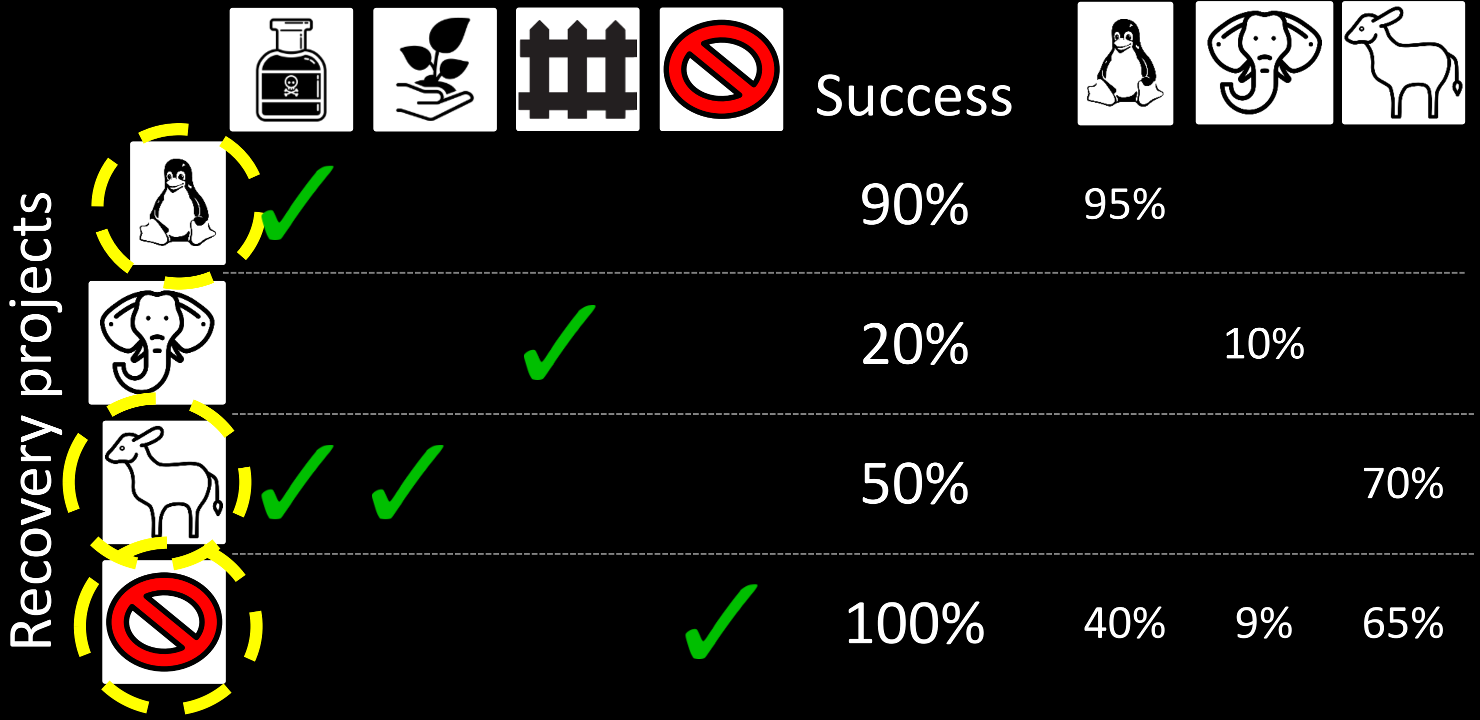
Recovery projects	Actions				Success	Persistence (%)		
	☠	🌱	▥	🚫		🐧	🐘	🐑
🐧	✓				90%	95%		
🐘			✓		20%		10%	
🐑	✓	✓			50%			70%
🚫				✓	100%	40%	9%	65%

Cost data

☠	$
🌱	$$
▥	$$$$$
🚫	$0

Project prioritization

Project data



Recovery projects

	Actions					Persistence (%)		
	☠	🌱	▥	🚫	Success	🐧	🐘	🐑
🐧	✓				90%	95%		
🐘			✓		20%		10%	
🐑	✓	✓			50%			70%
🚫				✓	100%	40%	9%	65%

Cost data

☠	$
🌱	$$
▥	$$$
🚫	$0

Budget $$$