An introduction to Systematic conservation planning with prioritzr

Brooke Williams & Caitie Kuempel (adapted from https://iiasa.github.io/eccb2024/An-introduction-to-Systematic-conservation-planning-with-prioritzr.pdf)

Location: Forgan Smith Building 01-E107 (90mins)

Preface

Welcome to the training course in systematic conservation planning with the prioritzr. This training course was originally held at the 2025 International Congress for Conservation Biology in Brisbane, although the materials found here will be preserved even after the conference and be openly available to everyone. Slides to the presentation can be found here.

What you will learn

- The basic concepts of Systematic conservation planning (SCP) and Integer Linear Programming (ILP) in particular
- How to prepare your input data for a Conservation planning project
- How to setup and run your first prioritization
- How outputs can be analysed and interpreted

Completing all course materials will take you on average 90 minutes, although people who have been exposed to similar methods or introduction before might take less.

In this training course a number of different terms will be used. Whenever there are uncertainties with regards to definitions, see the Glossary.

If you have already heard before about the basic concepts of SCP and ILP then feel free to jump to Part II and data preparation in Chapter 3.

Part I: Introduction to SCP

Preface

1.1 Systematic conservation planning

The classical definition of Systematic conservation planning (SCP) is that of a structured, scientific approach to identifying and prioritizing areas for conservation (Margules & Pressey (2000)). Its goal is to ensure that biodiversity is maintained and ecosystems are protected in a way that maximizes ecological, economic, and social benefits. Although SCP has been conceived specifically for creating and expanding reserve networks (usually protected areas), it can be used for much more including for example the identification of restoration, land-use planning or monitoring options. It is also a common misconception that a project implementing SCP is only about prioritization (the algorithm part). Rather, it describes a whole framework typically ranging from

1. Defining Conservation goals and objectives
2. Eliciting pathways to impact and theory of change with stakeholders
3. Compiling and preparing data
4. Identifying targets, constraints and costs
5. Formulating a planning problem and identifying priorities for it
6. Evaluating said priorities through robust performance metrics
7. Implementing the priorities in exchange with stakeholders
8. Monitoring the performance and adapting plans where necessary

1.2 Exact algorithms and integer programming

Exact algorithms in spatial planning are computational methods designed to find optimal solutions to spatial planning problems, where spatial planning involves the organization, management, and allocation of land and resources within a given area. These algorithms guarantee to find the best possible solution based on the defined criteria, constraints, and objectives of the problem.

Exact algorithms enable the solving of SCP problems as a mathematical model, such as a mixed (MILP) or integer linear programming (ILP) typically. Linear in this context refers to this common formulation of a planning problem, although non-linear problem formulations (e.g. quadratic or even more complex functions) are also possible. All LP problems have in common a specific objective function such as the maximum coverage or minimum set problem. See Hanson et al. (2019) for additional discussion of optimality in linear programming.

1.3 Tools and software

There are a range of tools and software for creating prioritizations in a SCP framework. Typical other well-known complementarity-based spatial conservation prioritization software are for example Zonation and Marxan, both of which use heuristic approaches for identifying priorities. For ILP problems the prioritizr R-package is the easiest and most comprehensive package currently available, although other options exist as well. It should be stressed that in principle any mathematical or programming language can be used to solve ILP problems. The prioritizr package simply provides a convenience wrapper.

Part II: Problem Creation

2 Obtaining data for the course

To get started, please download the data here. Save it in a folder named "ICCB_workshop" somewhere (for example your desktop). Unzip the data into a folder called data to follow along with the tutorial. Alternatively you can also fork or clone the entire tutorial from github if you feel comfortable with git (A version control system).

2.1 Systematic conservation planning

Specifically this folder contains:

1. Species Distributions: koala distributions modeled under current climatic conditions and future climate scenario SSP370 (2090) which represents a medium to high emissions scenario. Note: all spatial raster data is at ~5x5 km resolution (GDA_1994_Geoscience_Australia_Lambert) and harmonized spatially.
2. other data, including:
    a. PlanningUnits.tif - planning units are spatially defined areas or parcels used as the basic decision-making blocks in systematic conservation or land-use planning.
    b. Protected_areas.tif
    c. urban_centers.tif
    d. cost_hfp2013.tif

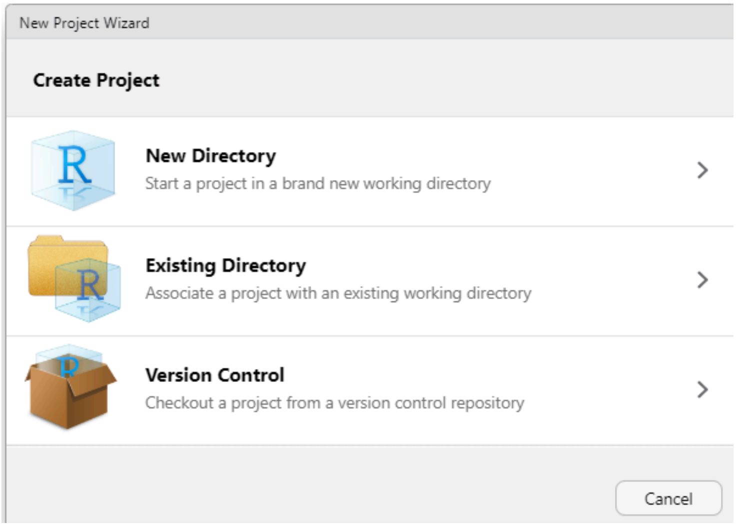
> **Steps for data preparation**
>
> For the use in prioritizr features, and all other spatial data, need to be perfectly harmonized with the planning units data and (same extent, resolution, number of grid cells). This step must be done prior to the prioritisation, as part of the data preparation.
> For this training workshop, the data is already prepared, but bear in mind that data preparation is an essential step in the conservation planning process. Note that in practice for any SCP project the data preparation might take considerable time for extraction and harmonization.

3 Loading prepared input data

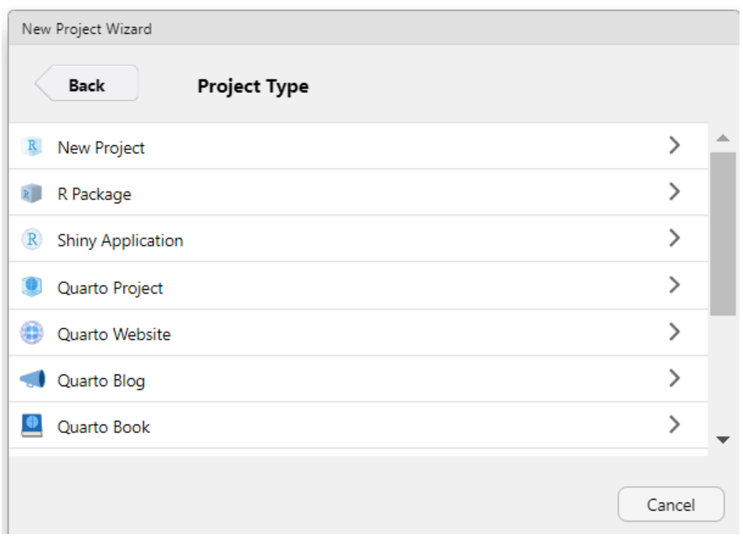
In the following sections we will load and explore the various data sources used for the planning in this course.

Open RStudio, select File and then New Project…

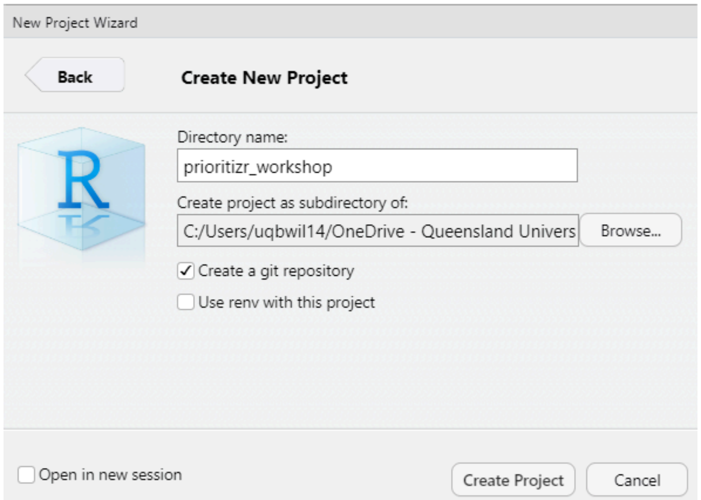
Select New Directory



Then New Project



Navigate to the folder you just created called “ICCB_workshop” and name your directory “prioritzr_workshop” and click Create Project.



Then click File>>New File>>R Script

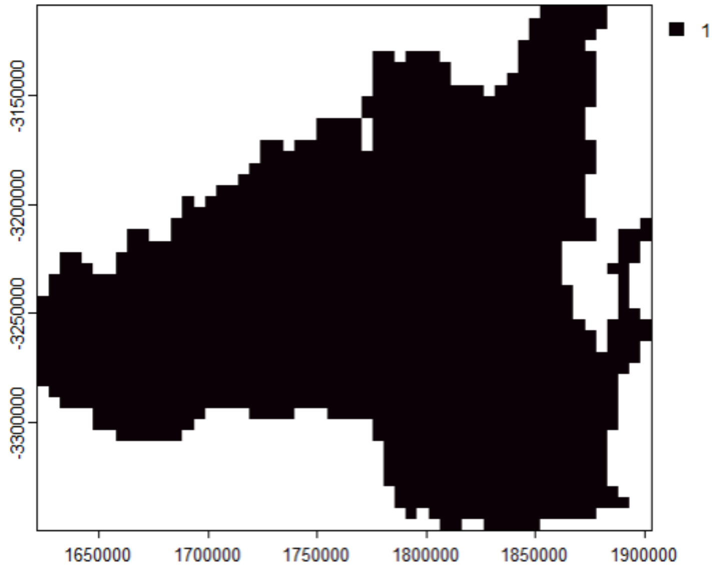
Finally, move your saved data folder inside of the prioritizr_workshop folder

3.1 Planning units

Planning units (PU) contains the spatial data of the study area. Although a range of different data formats are theoretically possible in prioritizr. PU are generally defined in SCP as the spatial units at which decisions are realised. For this tutorial we primarily rely on a raster format, specifically 10x10 km grid cells in Europe.

Let's read and plot the planning units raster:

```
# Required packages (install if necessary, eg. install.packages("terra"))
library(terra)
library(viridisLite)
library(prioritizr)

# Since you've created a project the working director should already be set to the folder you created. If you did not create a project, you may need to set the working directory to the correct folder. You can also consider using the here() package.

# Load the Planning unit
PU <- rast("data/otherdata/PlanningUnits.tif")
plot(PU, col = viridisLite::mako(n = 1))
```



Our study region is right here is Brisbane:

The value of the planning units can determine the cost of each planning unit in the prioritisation. In our case, we often want to reach 30% area coverage (out of total area) for example, based on international conservation agreements like the Global Biodiversity Framework. The cost for achieving this is the amount of land value in the Planning Units raster, here specified as equal value of 1 (so that the budget will be expressed in the number of grid cells in prioritizr).

> **! Important**
>
> Note especially when planning over larger extents the amount of area within a PU might differ depending on the geographic projection used. For this tutorial and simplicity, we rely on a longitude-latitude projection, which does not reflect area accurately (It is not an equal-area projection). In other words: PU in the north of Europe might contain less area than PU in the south of Europe despite having the same cost.
>
> When planning your own SCP project use a geographic projection appropriate for your case study!

3.2 Features

A feature is spatial data on the distribution of a biodiversity entity, typically a species, habitat, ecosystem service or similar.

Here, we consider the SDMs that you made in our previous workshop as features. We will focus on current distributions but we will also consider the projected distributions under a future climate scenario (SSP370 (2090)) that you created. By aiming to conserve areas that support koalas not only today, but also into the future, we are creating climate-sensitive protected areas.

Let's read the current SDM as a raster stack and plot one species as an example:

```
# Get the file names of the testing data

spp.list <- list.files(path = "data/SpeciesDistributions/", full.names = T, recursive = T, pattern =
".tif$")


# Load all files and rename them

spp <- rast(spp.list[grep("current", spp.list)])

# Get just the filenames (without full paths and extensions)

new_names <- tools::file_path_sans_ext(basename(spp.list[grep("current", spp.list)]))

# Load and assign names

spp <- rast(spp.list[grep("current", spp.list)])

names(spp) <- new_names

# Plot species distributions

plot(spp, axes = F,col = viridisLite::mako(n = 100, direction = -1), main = c(names(spp)))
```

```
# Do the same for "future" rasters

spp <- rast(spp.list[grep("future", spp.list)])

# Get just the filenames (without full paths and extensions)

new_names <- tools::file_path_sans_ext(basename(spp.list[grep("future", spp.list)]))

# Load and assign names

spp <- rast(spp.list[grep("future", spp.list)])

names(spp) <- new_names

# Plot first four species distributions

plot(spp, axes = F,col = viridisLite::mako(n = 100, direction = -1), main = c(names(spp)))
```

Note that we rename feature layers by species names. This enables linking the features rasters to a table of feature characteristics, weights, targets, taxonomy.

3.3 Existing protected areas

Often, we do not start from scratch: we often want to identify top priorities that complement and expand on existing protected areas. This helps meet the CARE principles of being representative. Protected areas used in this analysis represent IUCN categories Ia (most strict - science only), Ib (very strict - minimal recreation), and II (strict - recreation & education, no extractive use), downloaded from the World Database of Protected Areas (UNEP-WCMC and IUCN 2020).

```
# load protected areas data

PA <- rast("data/otherdata/protected_areas.tif")

# plot them

plot(c(PA),

    axes = FALSE,

    col = viridisLite::mako(n = 100, direction = -1),

    main = c("Protected Areas (I & II)"))
```

3.4 Areas under constrained use (locked-out or no-go areas)

Some areas are usually unavailable for SCP. Here we use urban areas as a proxy, as urban areas are highly developed and may not be available for protection. Our urban area dataset is derived from the Digital Atlas of Australia (Geoscience Australia 2025). We lock out (i.e., exclude from selection) the planning units that have high intensity urbanization. In doing so, we assume that, in these high-intensity areas, conservation would likely conflict with economic interests.

```
# load urban centers data

urban <- rast("data/otherdata/urban_centers.tif")

# plot them

plot(c(urban),

    axes = FALSE,

    col = viridisLite::mako(n = 100, direction = -1),

    main = c("Urban Centers"))
```

3.5 Costs

In the context of SCP Costs are typically spatially-explicit socio-economic data that can be factored into a prioritization to account for the feasibility of implementing conservation in a planning unit. In the planning they are typically used as a constraint to penalize or limit the allocation of PU to a solution. There are different types of costs that commonly used:

- Acquisition cost = price of land/water area
- Opportunity cost = lost revenue to other land use types
- Transaction cost = e.g. cost of negotiating protection
- Management cost = maintenance and management of the PA

In reality, we rarely have this information and need to use proxies. Here, we use global human footprint (hfp) as a proxy for socio-economic costs (Williams et al. 2020). Including the hfp as a cost layer would assume that highly human-dominated landscapes would be more costly to protect, than others.

```
# load cost layer

hfp <- rast("data/otherdata/cost_hfp2013.tif")

# plot the result
```

```
plot(hfp,
    axes = FALSE,
    col = viridisLite::mako(n = 10, direction = -1),
    main = "Global human footprint")
```

> **ℹ About costs**
>
> Generally be mindful about the use of costs in SCP as chosing any specific costing estimate can be quite impactful in driving final solutions.
> For some further background reading we recommend McCreless *et al.* (2013), Kujala *et al.* (2018) and Armsworth (2014)

3.8 Targets

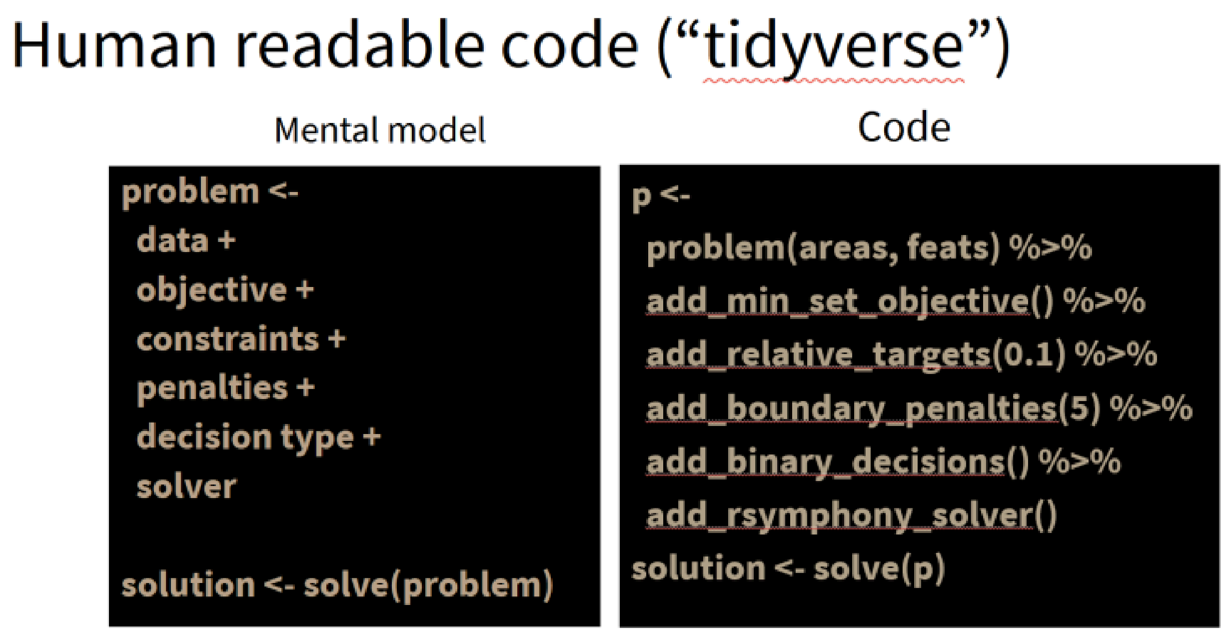
Another important aspect of planning are area-based targets, which define the amount of the distribution of each feature that is deemed sufficient to protect. Although one could set flat targets if there is a valid reasoning (e.g. 10% of all features), the most typical approach for targets is to use log-linear targets (Rodrigues et al. 2004). Another is to use the IUCN criteria to set targets based on the minimizing extinction risk (Jung et al. 2021).

Note that targets, similar as costs (see Section 3.5) can substantially drive the solution. Thus care should be taken how such targets are defined and used in SCP.

> **! Necessity of targets**
>
> Not every objective function in *prioritizr* requires targets. However the specification of targets is usually recommended as it forces the planner to think about the critical question of "How much do we want and need to conserve or manage". If such decisions are not taken by the analyst, it is usually taken by the algorithmic approach.

4 Create and understand planning problems

In the previous section (Chapter 3) we loaded a range of already prepared datasets including PU and features. Now we are ready to create our first conservation planning problem. Prioritizr makes use of a 'tidyverse' informed and human-readable syntax where a problem is defined by adding data, features and constraints sequentially to an object. This is thus quite similar to the use of 'dplyr' in R.



Pipe

We will use in this workshop the pipe symbol to chain different R functions (such as those from prioritizr) together. Useable are both the classical pipe from the magrittr package (%>%) and the pipe used by default since R version 4.0 (|>). We use both pipes often interchangeably in this workshop.

4.1 Our first planning problem

For our first problem we will create a problem that finds the best areas for 30% protected area coverage within our study region. Go to the next section (Chapter 5) to learn how to solve and interpret the outputs from a prioritizr problem.

```
budget.area <- round(0.3 * length(cells(PU))) #1

# Scenario 1
p <- problem(PU, spp) %>% #2
  add_min_shortfall_objective(budget = budget.area) %>% #3
  add_relative_targets(targets = 1) %>% #4
  add_default_solver() %>% #5
  add_proportion_decisions() #6
```

1. This effectively defines the total budget as 30% of the length of all PU. This works since the length is identical to the sum (cost =1).

2. Here we define a problem using the planning unit layer and the different species layer. Internally this will create an intersection of both. Other possible inputs to this function could be zones.
3. An objective function is added here. In this case we use the minimum shortfall objective.
4. For simplicity we define target as 100% for all species distributions
5. Here we add a solver. We rely here on CBC which in tests hasthe best performance among open-source solvers.
6. Here we add proportional decisions that means that proportions of planning units can be selected in the solution. This typically solves faster than binary decisions.

4.2 Understanding the problem object

Now that we have created a problem, let's have a look at the object.

```
# Simply run
p
```

```
> p
A conservation problem (<ConservationProblem>)
├•data
│├•features:    "future_koala_glm1", "future_koala_glm2", and "future_koala_RF" (3 total)
│└•planning units:
│ ├•data:       <SpatRaster> (1464 total)
│ ├•costs:      constant values (all equal to 1)
│ ├•extent:     1621552.0771, -3349593.6508, 1903369.5736, -3108767.7901 (xmin, ymin, xmax, ymax)
│ └•CRS:        GDA94 / Geoscience Australia Lambert (projected)
├•formulation
│├•objective:   minimum shortfall objective (`budget` = 439)
│├•penalties:   none specified
│├•targets:     relative targets (between 1 and 1)
│├•constraints: none specified
│└•decisions:   proportion decision
└•optimization
 ├•portfolio:   shuffle portfolio (`number_solutions` = 1, …)
 └•solver:      gurobi solver (`gap` = 0.1, `time_limit` = 2147483647, `first_feasible` = FALSE, …)
# i Use `summary(...)` to see complete formulation.
```

As visible the object contains information about the Planning units, including the spatial extent and geographic projection, as well as any features and complexity factors related to the formulation of the problem.

Running p$summary() will provide a summary with more detail.

> **Object**
>
> The prioritizr planning objects contain a range of different functions that can be queried and executed, for example to obtain summaries or specific datasets and parameters contained within. For example object$data will return: (a) features, (b) planning units, (c) an intersection call rij_matrix and more information.

4.3 Different datasets for planning

In this tutorial we use gridded datasets. However it should be noted that - internally - prioritizr does not operate on spatial files but on tabular data. This is also true when not gridded but vector data are provided.

Why is this relevant to know? When creating a problem with spatial data (gridded or vector files), at the time of problem creation these data are internally converted into large tabular data. When planning over many species or planning units it can be computational efficient to not have prioritizr, but to do this conversion directly and then supply tabular data.

Typical steps involved here are:

1. Converting gridded planning unit data to a table in long form (row) containing both the cell id, the planning unit id and the cost
2. Convert the features into a long table containing the planning unit id, the feature id (and name) and the amount stored.
3. Intersecting the tables created in step 1) and 2)
4. Preparing any other tables for weights or targets, aligned with the planning unit id.

> **Note**
>
> Another context might be when data has already been formatted for use in another software such as Marxan. Prioritizr is able to directly use the formatted tables prepared for a typical Marxan application, thus making it easy to switch between software.

Part III: Solving a problem

5 Solving and interpreting solutions

In this section we will create a scenario analysis by expanding on the previously set up planning problem (object **p**, see Chapter 4). Specifically we will four problems, analyse its their outputs and calculate a range of metrics and indicators describing it.

5.1 Find a solution for a conservation problem

In the previous section we defined a conservation problem based on a planning unit file, features, the specification of an objective function and decision variable.

We can use the solve function to solve the previously defined problem. We can look at the solved optimisation problem by plotting it. This shows the planning units that were selected (value of 1) versus those that were not selected (value of 0) to meet the defined planning objectives.

```
s1 <- solve(p)
plot(s1)
```

Now let's look at another scenario where we lock out, or exclude, high density urban areas as they are unsuitable for conservation.

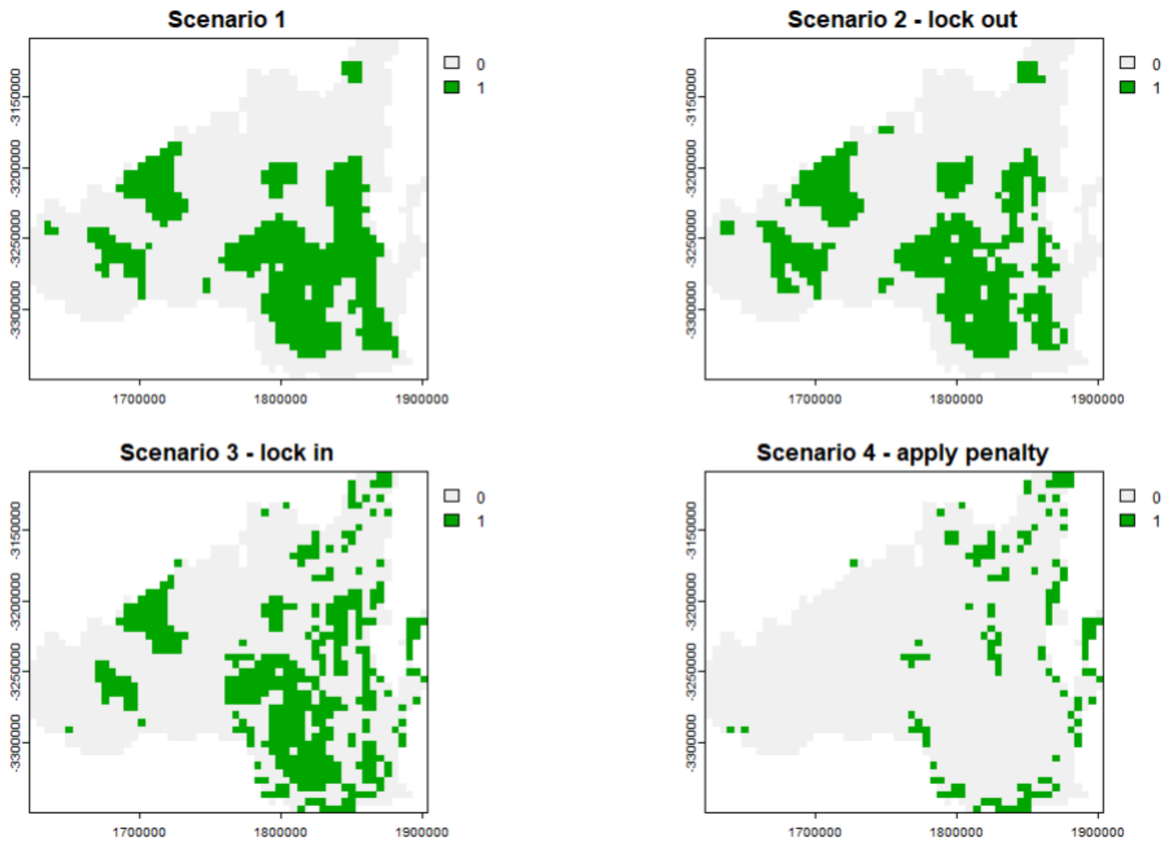
```
# Scenario 2 - lock out urban areas
p <- problem(PU, spp) %>%
  add_min_shortfall_objective(budget = budget.area) %>% # Budget in # of cells or area units
  add_relative_targets(targets = 1) %>%
  add_proportion_decisions() %>%
  add_locked_out_constraints(urban) %>%  # <- Lock out unsuitable areas
  add_default_solver()

s2 <- solve(p)
plot(s2)

# Scenario 3 - lock in protected areas
p <- problem(PU, spp) %>%
  add_min_shortfall_objective(budget = budget.area) %>% # Budget in # of cells or area units
  add_relative_targets(targets = 1) %>%
  add_proportion_decisions() %>%
  add_locked_in_constraints(PA) %>%  # <- <- Lock in protected areas
  add_locked_out_constraints(urban) %>%  # <- Lock out unsuitable areas
  add_default_solver()

s3 <- solve(p)
```

```
plot(s3)

# Scenario 4 - hfp is a cost/penalty
# Build the problem
p <- problem(PU, spp) %>%
  add_min_shortfall_objective(budget = budget.area) %>%  # Budget in # of cells or area units
  add_relative_targets(targets = 1) %>%
  add_linear_penalties(penalty = 1, data = hfp) %>%  # Penalize high HFP values
  add_proportion_decisions() %>%
  add_locked_in_constraints(PA) %>%  # <- Lock in protected areas
  add_locked_out_constraints(urban) %>%  # <- Lock out unsuitable areas
  add_default_solver()

s4 <- solve(p)
plot(s4)
```

5.2 Plot the solutions

The output of the solved problems from above are essentially spatial rasters that can be plotted.

```
# Plot all scenarios side by side
# Set plotting area to 1 row, 4 columns
par(mfrow = c(2, 2), mar = c(3, 3, 3, 1))  # 2 rows, 2 columns
```



With the specified objective functions, can you summarize the amount of area contained in the solution? How much would you expect?

5.3 Calculate performance evaluation metrics

Now that we have created a solution and visualized it, the next obvious question would be: How good is it? What are we conserving and what maybe not? These are critical questions for any SCP application and different problem formulations will achieve different levels of representation. Performance metrics are usually used to answer such questions in any SCP workflow, and they can assess a solution based on its spatial distribution and/or the features conserved within.

During the problem setup we defined a set of targets for each feature, so naturally a question could be for how many species we reach the target and also how far are we off (see also (Jantke et al. 2019)). We used a minimum shortfall objective, thus our objective is to minimize the shortfall (e.g. distance) between the amount covered by the feature as constrained by the budget. Thus we can most feasibly assess the performance of this solution for the species by assessing their representation and their target shortfall.

```
# In Prioritizr there are convenience functions that can summarize the
# coverage of species in terms of amount held
```

```
#Calculate metrics
#Scenario 1
rpz_target_spp_s1 <- eval_target_coverage_summary(p, s1) #1
# mean representation across all species
mean(rpz_target_spp_s1$relative_held) #2
# mean target shortfall across all species
mean(rpz_target_spp_s1$relative_shortfall) #3
#Scenario 2
rpz_target_spp_s2 <- eval_target_coverage_summary(p, s2)
mean(rpz_target_spp_s2$relative_held)
mean(rpz_target_spp_s2$relative_shortfall)
#Scenario 3
rpz_target_spp_s3 <- eval_target_coverage_summary(p, s3)
mean(rpz_target_spp_s3$relative_held)
mean(rpz_target_spp_s3$relative_shortfall)
#Scenario 4
rpz_target_spp_s4 <- eval_target_coverage_summary(p, s4)
mean(rpz_target_spp_s4$relative_held)
mean(rpz_target_spp_s4$relative_shortfall)
```

1. This calculates the coverage of the features (taken from the problem) over the solution, also providing the initial amount.
2. Here we calculate mean representation, e.g. how much habitat is held by the solution across all features.
3. This calculates the average shortfall, so the difference between held amount and target across features.

Which scenarios performed the best and worst? What does this tell you about planning in the context of competing objectives?

References

Geoscience Australia. 2025. “Built Up Areas.” https://digital.atlas.gov.au/.

Jantke, Kerstin, Caitlin D. Kuempel, Jennifer McGowan, Alienor L. M. Chauvenet, and Hugh P. Possingham. 2019. “Metrics for Evaluating Representation Target Achievement in Protected Area Networks.” *Diversity & Distributions* 25 (2): 170–75.

Jung, Martin, Andy Arnell, Xavier de Lamo, Shaenandhoa García-Rangel, Matthew Lewis, Jennifer Mark, Cory Merow, et al. 2021. “Areas of Global Importance for Conserving Terrestrial Biodiversity, Carbon and Water.” *Nature Ecology & Evolution* 5 (11): 1499–1509.

Rodrigues, Ana S. L., Sandy J. Andelman, Mohamed I. Bakarr, Luigi Boitani, Thomas M. Brooks, Richard M. Cowling, Lincoln D. C. Fishpool, et al. 2004. “Effectiveness of the Global Protected Area Network in Representing Species Diversity.” *Nature* 428 (6983): 640–43.

UNEP-WCMC, and IUCN. 2020. “Protected Planet: The World Database on Protected Areas (WDPA) Version 1.6.” 2020. https://www.protectedplanet.net/en.

Williams, Brooke A., Oscar Venter, James R. Allan, Scott C. Atkinson, Jose A. Rehbein, Michelle Ward, Moreno Di Marco, et al. 2020. “Change in Terrestrial Human Footprint Drives Continued Loss of Intact Ecosystems.” *One Earth (Cambridge, Mass.)* 3 (3): 371–82.

Glossary

A glossary of key terms used in this Training course

Term	Abbreviation if any	Definition
Allocation		Synonym use for process taken by the optimization and the decision variable per zone within a PU. Example: Binary decision is to identify expansion or not. The solver allocates PU to a solution based on complementarity.
Boundary Length Modifier	BLM	A penalty constant added to a conservation problem that penalizes selecting isolated patches. Results in overall more compact solutions.
CARE	CARE	A often used abbreviation that stands for ***C**onnectivity*, ***A**dequacy*, ***R**epresentation*, and ***E**ffectiveness* which key principles that should be considered when designing a conservation network. See the Marxan website for more information.
Conservation Prioritization		The computational process of identifying (spatial) priorities for a given conservation objective (such as for identifying protected areas). Usually comes in in the form of a map.
Constraint		A (often linear) constant or parameter that limits the selection of certain PU as part of the solution.
Cost	c	A single or multiple constant typically used in SCP to penalize solutions and any allocation of land to PU.
Exact algorithm		A method to solve mathematical optimization problems using a solver.
Feature		Spatial data representing the distribution of an individual biodiversity unit, such as a species, habitat, ecosystem service, etc.
Integer		In programmatic terms a full number (e.g. -1, 1, 2, 3, ...)
Integer Linear Programming	ILP	Mathematical problem formulation using Linear Programming (ILP) where the variables are integer values and the objective function and equations are linear.
Penalty	p	In the context of SCP commonly referring to a constant parameter used to penalize solutions. For example a costing or connectivity matrix.
Planning unit	PU	The fundamental unit at which decisions in SCP are realized. Can be of multiple formats such as grid cells or farms
Solver		An algorithm to identify 'solutions' to a mathematical problem. Often available as open- or closed-source software.
Systematic Conservation Planning	SCP	A framework and step-wise approach towards mapping conservation areas. Usually involves multiple steps such as the identification of a problem and the theory of change, data collection and preparation, conservation prioritization, evaluation and finally implementation. See Margules & Pressey (2000)
Zonation		A SCP software for creating conservation priority rankings and priority maps and tradeoffs. Uses a meta-heuristic approach and benefit functions for ranking. The latest version is Zonation 5.
Zone	Z	Zones are spatial or thematic management units over which decisions are made in a SCP problem. Examples: Core zone, Buffer zone and sustainable management zone in a national park.