Integer programming formulation

21 February 2020

We will begin by recalling fundamental concepts in systematic conservation planning. Conservation features describe the biodiversity units (e.g. species, communities, habitat types) that are used to inform protected area establishment. Planning units describe the candidate areas for protected area establishment (e.g. cadastral units). Each planning unit contains an amount of each feature (e.g. presence/absence, number of individuals). A prioritisation describes a candidate set of planning units selected for protected establishment. Each feature has a representation target indicating the minimum amount of each feature that ideally should be held in the prioritisation (e.g. 50 presences, 200 individuals). Furthermore, prioritisations that are costly to implement are not desirable, and prioritisations that are excessively spatially fragmented are not desirable. Thus we wish to identify a prioritisation that meets the representation targets for all of the conservation features, with minimal acquisition costs and spatial fragmentation.

We will now express these concepts using mathematical notation. Let denote the set of conservation features (indexed by ), and denote the conservation target for each feature . Let denote the set of planning units (indexed by ), and denote the cost of establishing planning unit as a protected area. Let denote the amount of each feature in each planning unit (e.g. presence or absence of each feature in each planning unit). To describe the spatial arrangement of planning units, let denote the total amount of exposed boundary length of each planning unit. Also let denote the total amount of shared boundary length between each planning unit and (where and are not equal). Furthermore, to describe our aversion to spatial fragmentation, let denote a spatial fragmentation penalty value (equivalent to the “boundary length modifier” parameter in the Marxan decision support tool). Higher penalty values indicate a preference for less fragmented prioritisations.

We will consider the following example to explain the spatial and variables in further detail. Imagine three square planning units (, , ) that are each 100 100 m in size and arranged left to right in a line. These planning units each have a total amount of exposed boundary length of 400 m (i.e. , , ). Additionally, and have a shared boundary length of 100 m (i.e. , ); and have a shared boundary length of 100 m (i.e. , ); and and have a shared shared boundary length of 0 m (i.e. and ). Note that planning units do not share any boundary lengths with themselves (i.e. , , ).

We use the binary decision variables for planning units (eqn 1a), and for planning units and (eqn 1b).

$$\begin{aligned}
X\_j &=
\begin{cases}
1, \text{ if $j$ selected for prioritisation}, \tag{eqn 1a} \\
0, \text{ else }
\end{cases} \\
Y\_{jk} &=
\begin{cases}
1, \text{ if both $j$ and $k$ selected for prioritisation}, \tag{eqn 1b} \\
0, \text{ else } \\
\end{cases}\end{aligned}$$

The reserve selection problem can be formulated following:

$$\begin{aligned}
\text{minimize} & \sum\_{j \in J} X\_j C\_j + \left( \sum\_{j \in J} p E\_{j} \right) - \left( 0.5 \times \sum\_{j \in J} \sum\_{k \in J} p Y\_{jk} L\_{jk} \right) \tag{eqn 2a} \\
\text{subject to} & \sum\_j^J R\_{ij} \geq T\_i & \forall i \in I \tag{eqn 2b} \\
& Y\_{jk} - X\_j \leq 0 & \forall j \in J \tag{eqn 2c} \\
& Y\_{jk} - X\_k \leq 0 & \forall k \in J \tag{eqn 2d} \\
& Y\_{jk} - X\_j - X\_k \geq -1 & \forall j \in J, k \in K \tag{eqn 2e} \\
& X\_j \in \{ 0, 1 \} & \forall j \in J \tag{eqn 2f} \\
& Y\_{jk} \in \{ 0, 1 \} & \forall j \in J, k \in K \tag{eqn 2g} \\\end{aligned}$$

The objective function (eqn 2a) is the combined cost of establishing the selected planning units as protected areas and the penalized amount of exposed boundary length associated with the selected planning units. Constraints (eqn 2b) ensure that the conservation targets () are met for all conservation features. Additionally, constraints (eqns 2c–2e) ensure that the variables are calculated are correctly (as outlined in Beyer *et al.* 2016). Finally, constraints (eqns 2f and 2g) ensure that the decision variables and contain zeros or ones.