

Appendix 4. Explanation of the SITES Model

Adapted from *SPOT: The Spatial Portfolio Optimization Tool, User Guide*. Dan Shoutis 2003. SPOT is a successor to the SITES model, and is being developed by The Nature Conservancy, using the same algorithm for selection.

1	Overview	1
1.1	Introduction	1
1.2	How it works	1
1.3	Limitations	2
2	Methodology: The SITES Cost Function	2
2.1	Overview.	2
2.2	Analysis units	3
2.3	Targets and goals	4
2.4	Target Distribution	5
2.5	Target representation	7
2.6	Example cost function	10
3	Methodology: Simulated Annealing	12
3.1	Overview.	12
3.2	The Annealing of function.	12
3.3	Hill descent.	13
3.4	Simulated annealing.	14
3.5	Considerations.	16

1. Overview

1.1 Introduction

The Spatial Portfolio Optimization Tool (SITES) is a generalized tool for conservation portfolio selection, using a flexible approach to automatically design an efficient portfolio around specified conservation goals.

About this documentation:

This guide is intended to explain the methodology behind the SITES model and the assumptions necessary for its application. To learn the application of the SITES model refer to the documentation that accompanies the application. To learn more about simulated annealing refer to the bibliography at the end of this appendix.

1.2 How it works

SITES analyzes a region by dividing it into small parcels called analysis units, then forming a portfolio by marking individual units as included or excluded from a portfolio.

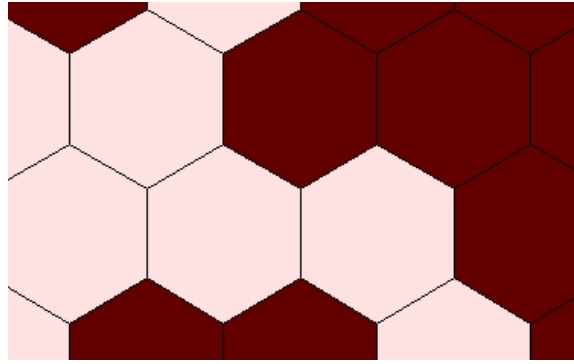


Figure 1.1: Detail of hexagonal analysis units in a SITES portfolio; dark units are in the portfolio and light units were not included.

Analysis unit: Analysis units form the portfolio; they are small areas that are marked as in or out to create a portfolio.

During a process known as simulated annealing, SITES forms and analyzes millions of portfolios while searching for the most efficient portfolio. Each is evaluated according to three criteria:

- How well it meets conservation goals
- The area included
- The fragmentation of the portfolio
- The portfolio that does the best job of minimizing the area and fragmentation while meeting conservation goals is considered the most optimal, and is output as the final result.

Simulated annealing: A general technique for finding the lowest value of a function through many trial runs and repeated adjustment to input values.

1.3 Limitations

- SITES is prone, just as any other automated tool, to the “Garbage In – Garbage Out” syndrome. Any results will only be as good as the input datasets. Additionally, although SITES is very flexible, the final portfolios it produces will be the best according only to its internal criteria, which may differ from what planners have in mind.
- SITES creates and evaluates an entire portfolio at a time, which means that it never makes a decision about any individual area. Thus, there is no information available as to why specific areas were included or excluded from resulting portfolio.
- Although SITES’s algorithm is statistically likely to find the most efficient portfolio given enough iterations, just how many iterations are necessary is variable and can change drastically with the nature of the region. Some experimentation will prove necessary.
- SITES’s portfolio assembly algorithm is not deterministic. Running the tool multiple times on the exact same data will produce many, slightly differing, result portfolios. To overcome this limitation, planners should be performing many runs on the same data to ensure that SITES is giving comparable results for each.

2. Methodology: The SITES Cost Function

2.1 Overview

SITES attempts to assemble a portfolio with the minimal possible value of a cost function that encapsulates desirable characteristics for an ecoregional plan. The SITES cost function is derived from the following goals for a portfolio:

- The portfolio should minimize the area required to adequately represent targets.
- The portfolio should meet conservation target goals set for the region.
- Fragmentation should be avoided; when choosing between a scattered area and a contiguous one with similar representation and size, the contiguous one is preferable.

Cost function: The cost function for SITES calculates a single cost value for a given portfolio that represents its effectiveness.

In order to search for a portfolio that meets these principles, the region is first broken into small units known as analysis units. SITES forms a portfolio by marking analysis units as included from the portfolio by marking analysis units as included or excluded from the portfolio. Conservation goals are specified on a per-target basis, and each unit is attributed with the amounts of each target that it contains. To take into account the three principles, the cost function is a sum of:

- A base cost for each analysis unit included in the portfolio. This will increase the value of the cost function as more analysis units are added, encouraging SITES to find solutions that use less units.
- A shortfall cost, penalizing the portfolio for failures to meet goals. Every unmet target will increase the cost function's value.
- A boundary cost the boundary of the portfolio. The way SITES measures a portfolio's fragmentation is by the length of its boundary, and longer boundaries mean a higher value of the cost function.

More formally:

$$\text{cost}(x) = \text{base}(x) + \text{boundary}(x) + \text{short fall}(x)$$

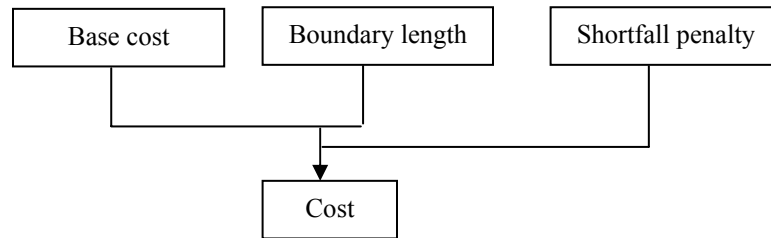


Figure 2.1: The cost function

Base cost: A component of the SITES cost function that encourages SITES to minimize the area of the portfolio. It is the sum of the cost specified for each analysis unit included in the portfolio.

Boundary cost: A component of the SITES cost function, aimed toward minimizing a portfolio's fragmentation by minimizing the length of its boundary.

Shortfall cost: A component of the SITES cost function that penalizes portfolios that don't meet conservation goals

2.2 Analysis units

Analysis units can be arbitrarily shaped and sized, depending on the needs of planners. In The Nature Conservancy's ecoregional plans, small hexagons have been used most often.

The prime consideration when choosing analysis units is size:

- If the units are too small, there will be so many of them that portfolio assembly will be unacceptably slow or fail to produce a robust answer.
- If the units are too large, the analysis will be too coarse and fail to adequately represent reality.

It is important to remember that everything in SITES is based around analysis units: target distributions or other spatial information, no matter how fine-grained, is rounded up to the nearest analysis unit, much like information in digital images is rounded up to the nearest pixel. See figure 2.5 for an example.

Analysis units consist of:

- An ID
- A base cost

When a unit is included in the portfolio, its cost is added to the total value of the cost function. This cost can represent simple area, or planners can use more sophisticated values to make some

units preferred over others. Often, measures of GIS suitability are integrated into the basic unit cost.

The total base cost is thus:

n

$$Base(x) = \sum_{k=1} BaseCost_k \quad (2.2)$$

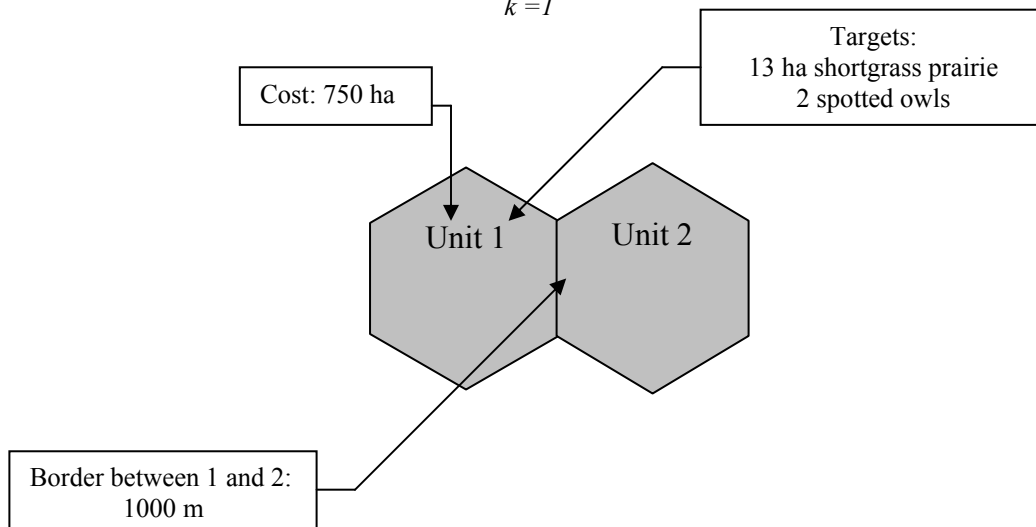


Figure 2.2: Analysis units, boundary information, and target distribution information for each unit k included in the portfolio.

2.3 Targets and goals

SITES represents planners' goals by a list of targets, each representing a separate biological value that needs to be preserved. A target consists of:

- A numeric ID
- A name
- A goal
- A penalty factor
- A minimum representative area

Target: a biological feature with a conservation goal that SITES attempts to meet during assembly.

For example, if a portfolio goal is to preserve at least 10,000 ha of a shortgrass prairie system type, then typical settings in SITES's target table might be an ID of 1001, a name of "prairie", a goal of 10,000, a penalty factor of 1.0 and a minimum representative area of 0 ha.

Minimum area: The minimum contiguous amount of a target required for it to contribute to a conservation goal.

SITES will attempt to find portfolios that contain enough of a target to meet its goal. Portfolios that cannot fully represent a target will be penalized with a shortfall cost. (This calculation is discussed in more detail in Section 2.7.)

The minimum area requirement prevents SITES from counting an occurrence of a target unless its contiguous size is greater than the specified amount. For example, if the type of prairie in the above sample target is only viable in occurrences of 1,000 ha or greater, then a minimum area requirement of 1,000 ha will force SITES to collect connected analysis units that represent more than this amount before they can contribute to this target's goal. See Section 2.6 for more information on how this is calculated.

Target patch: An occurrence of a target that can spread over several neighboring analysis units.

SITES also allows planners to give a target a variable degree of impact on the portfolio via the penalty factor. Any shortfall penalties generated by a target are multiplied by this factor before being added to the total portfolio cost, so planners can weigh individual targets appropriately.

Penalty factor: Sets the importance of representing a target, relative to other targets and the base and boundary costs.

It is important to note that this weighting is relative to both the overall base and boundary components of the total cost, as well as other targets. (See Section 3.5.2.)

2.4 Target distribution

SITES uses a table of target distributions to represent the spread of a target through a region, identifying each analysis unit with the targets that occur there and the amount of each:

- The analysis unit ID
- The target ID
- The amount of the target contained within the analysis unit

For the target distribution (with 1=Spotted Owl and 2=Shortgrass Prairie) given in Figure 2.2, SITES's target distribution table will look like:

Target ID	Unit ID	Amount
1	1	13
2	1	2

The portfolio boundary

In order to discourage portfolio fragmentation, SITES takes into account the portfolio's perimeter. (A fragmented portfolio will have a much longer boundary than a well-connected portfolio.) SITES represents boundaries with a table that contains the spatial relationship between neighboring analysis units.

Boundaries consist of:

- Two neighboring analysis unit Ids
- The length of the boundary shared between the units

To calculate the length of the portfolio boundary, SITES looks at every boundary between two units. If both units are in the portfolio, then that boundary is interior to the portfolio and not exposed. If one unit is in the portfolio, and the other out, the boundary is exposed and the indicated length is added to the total. Figure 2.3 illustrates the process.

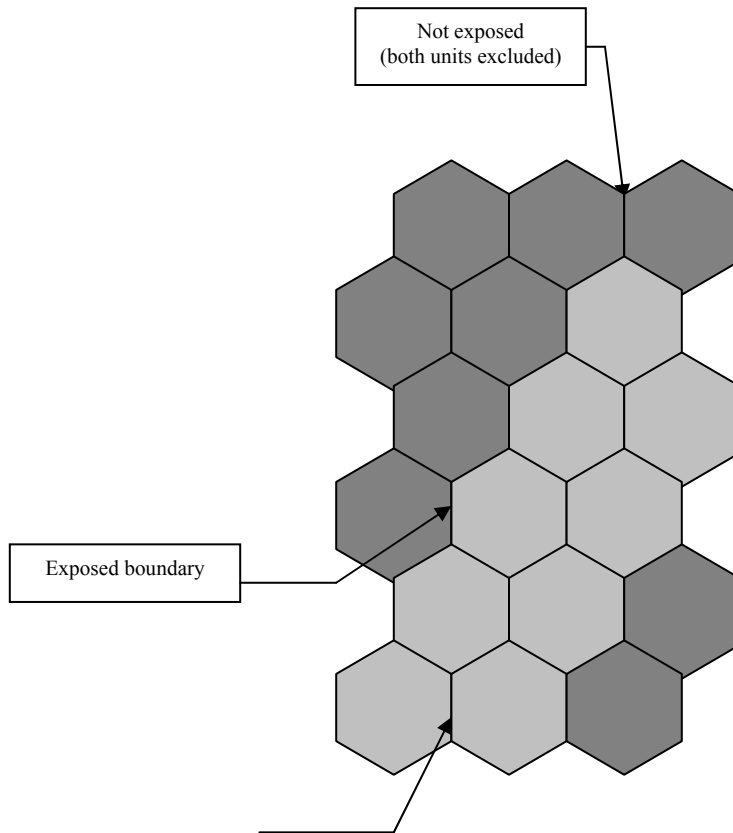


Figure 2.3: Finding exposed boundaries to calculate the boundary length of the portfolio.

Additionally, a unit can be specified as having a boundary with itself, which is useful for units that will always have a boundary when included in the portfolio (e.g., a unit at the edge of the region). Before combining the calculated boundary length with the rest of the cost function, it is multiplied by a coefficient called the boundary length modifier, or the BLM.

Boundary length modifier: A multiplier that converts and scales the boundary length of a portfolio before adding it to that portfolio's cost.

2.4.1 The boundary length modifier

Before the boundary length of a portfolio is added to the SITES cost function (as the boundary cost), it is scaled by a factor called the boundary length modifier (BLM). The BLM serves several purposes, and is thus somewhat confusing.

- To specify the relative importance of fragmentation in the cost function. Smaller values will make fragmentation less important than meeting goals and minimizing area.
- To convert units. If the base analysis unit cost is specified as hectares (or even more confusingly, GIS Suitability indexes), and boundary length as kilometers, the BLM must serve the purpose of converting the boundary into comparable units.
- To make “area” and “length” comparable. The least fragmented shape possible is a circle, and the area to circumference ratio can serve as a guide for this. Because of the many conflicting factors inherent in the BLM, the best way to arrive at a good number is via experimentation.

The boundary portion of the cost function is:

$$Boundary(x) = (BLM)(\sum_{k=1}^n boundary_k) \quad (2.2)$$

for every exposed boundary k .

2.5 Target representation

SITES calculates the representation of a target in a portfolio in the following way:
If the target has no minimum representation requirements:

- For every unit that is marked as being in the portfolio and contains the target, the amount is added to come up with a total. If the target has a minimum representation, SITES uses the following procedure to take into account the size of target patches before adding them to the total representation (Figures 2.4 and 2.5 illustrate this process):
- SITES begins by finding a unit that is in the portfolio with the target present.
- SITES adds the amount to a temporary running total.
- For every unit that: shares a boundary with the current unit, is in the portfolio, contains some of the target, and has not already been examined; SITES adds the amount to the running total.
- SITES then repeats the process with the neighbors' neighbors, then those units' neighbors, and so on, until it runs out of connected units that contain the target.
- If the running total of this connected patch is greater than the target's minimum area requirement, then SITES adds it to the target representation amount. Otherwise, it is dropped.
- SITES continues finding patches in the way described above until the entire portfolio has been examined.

It is important to note that the accuracy of this procedure depends to a large extent on analysis unit size: If two neighboring units have presence records for the same target, it is assumed that they are part of a larger presence that covers both. This may not be the case, since neighboring units may have two independent occurrences that will be erroneously added together during the minimum area assessment (Figure 2.5). One way to avoid this pitfall is to drop all occurrences that fall below the minimum area (such as the small patch in unit 6 in the figure) as a preprocessing step, before entering them into SITES's target distribution table.

SITES uses the target representation to calculate a shortfall penalty, as well as to report back to planners on goal performance.

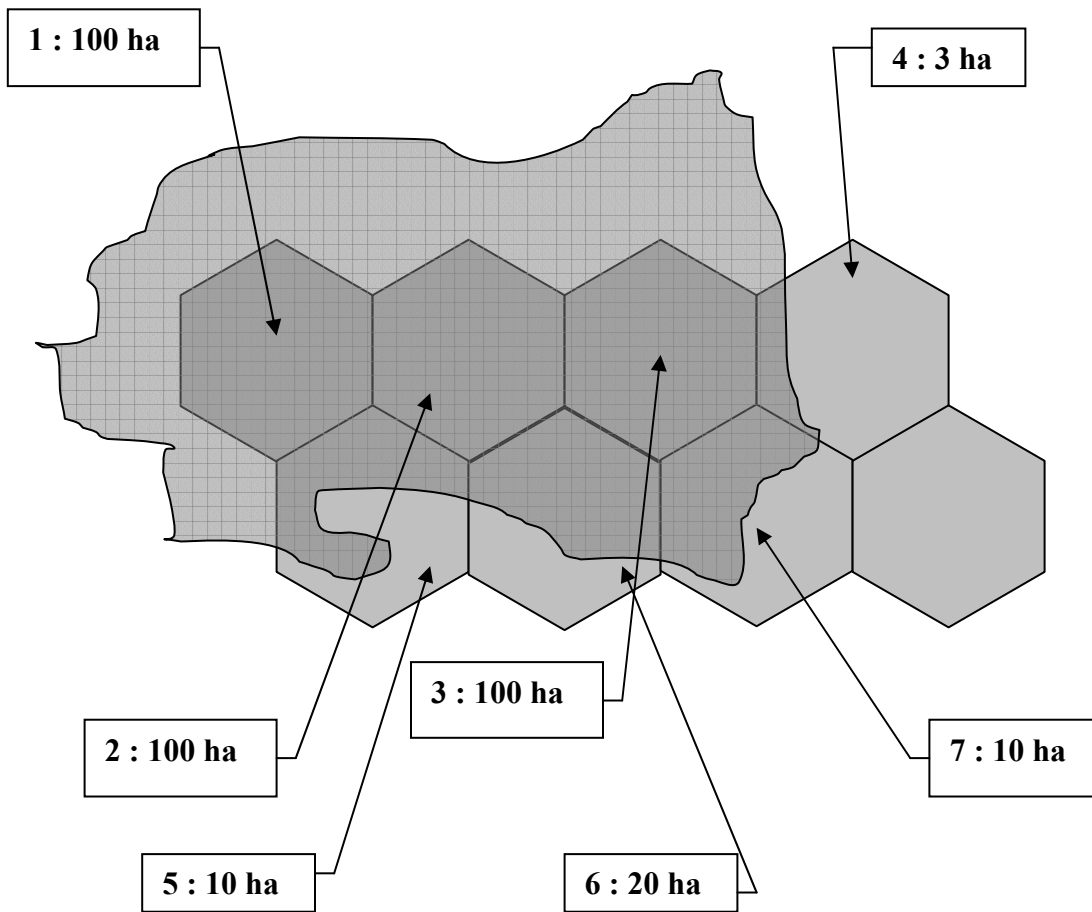


Figure 2.4: Minimum Representation in action: The distribution of a target with minimum representation. If the minimum representation for this target was 110 ha, then a portfolio that included units 1 and 2 would meet the requirement and contribute 200 ha toward the goal. A portfolio that included only units 1 and 6 would not have a connected target presence sufficient to meet the representation requirement, and so there nothing would be contributed toward the target goal.

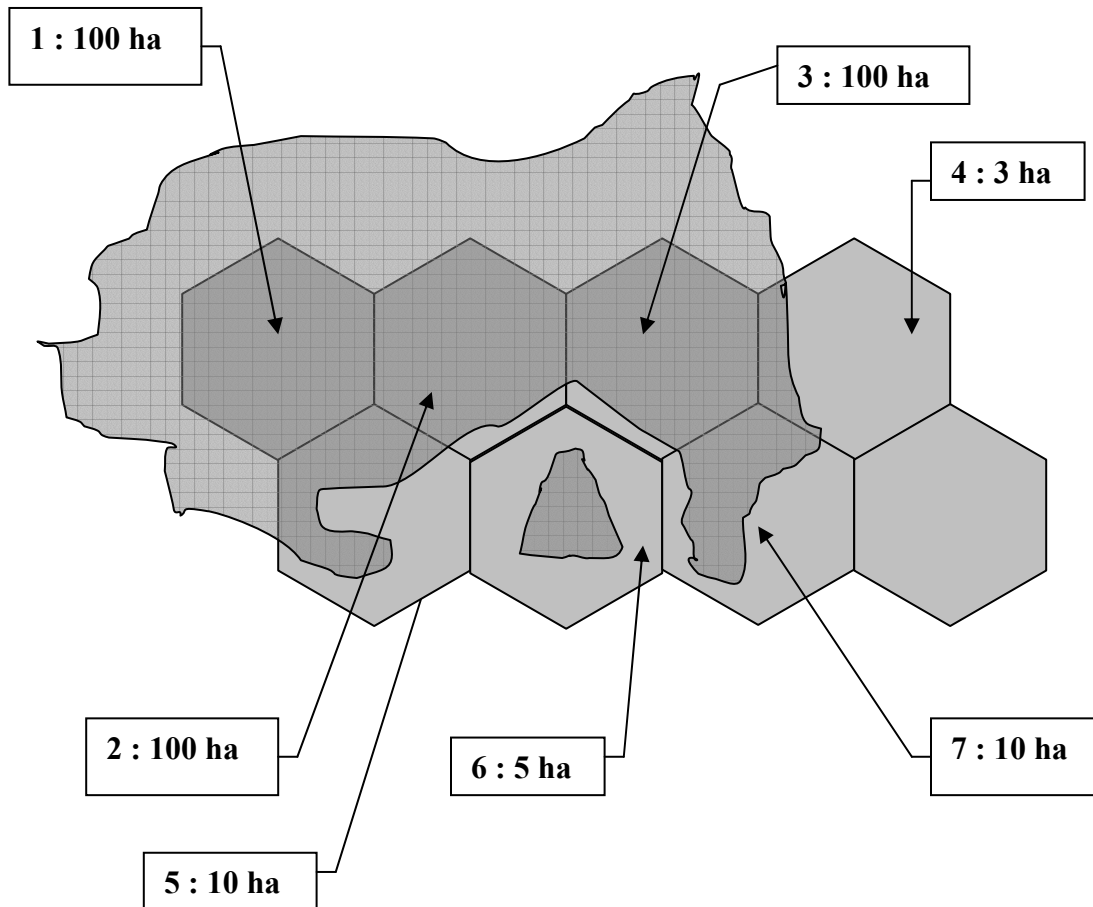


Figure 2.5: Issues with minimum representation and analysis unit size: With a minimum representation requirement of 110, a portfolio that includes units 1, 5, and 6 will meet the minimum necessary area. Even though the actual target distribution is not connected, SITES will assume that the 5 ha in unit 6 are connected to those in 5 and 1 because the analysis units are touching.

Shortfall penalties

In order to calculate a penalty for portfolios that fail to represent targets, SITES first pre-calculates an initial penalty amount for each target, designed so that the penalty imposed for a shortfall approximates, and is slightly greater than, the cost required (in terms of base cost + boundary length) to fully represent the target. Appendix B describes this in more detail.

The practical effect of this is that the cost of making up a shortfall will be slightly less than the penalty imposed by the shortfall; this way the simulated annealing process will favor portfolios with more complete target representation.

Each penalty is calculated in an initial phase where SITES builds a mini-portfolio for each target. The cost of this mini-portfolio, which is a good approximation of the cost to fully represent the target, is then stored as a penalty cost. Shortfall penalties are calculated by multiplying this cost by the proportion of any shortfall, as well as a target's penalty factor.

$$\text{Shortfall Penalty} = (\text{Penalty Factor}) \left(\frac{\text{Shortfall Amount}}{\text{Goal Amount}} \right) (\text{Penalty Cost}) \quad (2.4)$$

For example, if a goal is 90% met, the penalty cost will be 10% of the calculated amount to represent the full target. Additionally, a target's penalty factor is multiplied by the initial shortfall cost to arrive at a final shortfall penalty for that target.

2.6 Example cost function

This section ties everything together with a simplified portfolio assembly situation, calculating the cost function for a portfolio by hand.

2.6.1 Analysis units

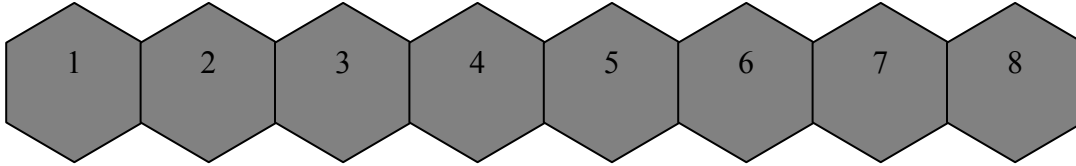


Figure 2.6: The analysis units for a simplified cost portfolio assembly situation.

The analysis units for this portfolio are specified in Figure 2.6, and the corresponding table of Ids and costs is:

Unit ID	Unit Cost
1	100
2	100
3	150
4	100
5	100
6	175
7	200
8	100

2.6.2 Boundaries

Each unit shares boundaries with one or two neighbors. If the side of each hexagon was 100 units long, the boundary definition table will be:

Unit A	Unit B	Length
1	2	100
2	3	100
3	4	100
4	5	100
5	6	100
6	7	100
7	8	100

2.6.3 Targets and distributions

This simple portfolio has targets “Trees and Grass” specified as follows:

Target Name	Goal	Penalty Factor	Penalty Amount
Trees	50	1	400
Grass	1000	1.5	200

The target distributions are:

	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6	Unit 7	Unit 8
Trees	30	30	30	30	0	0	0	0
Grass	0	600	0	0	800	0	120	0

2.6.4 Calculating the cost function

For this example, assume that units number 1, 3, and 5 are included in the portfolio. What would the value of the cost function be with a BLM of 0.5?

The base cost:

The base cost for the portfolio is the sum of the cost of each analysis unit:

$$basecost(x) = 100 + 150 + 100 = 350$$

The boundary cost:

The boundary cost is the BLM, times the length of all exposed boundaries. The boundaries between 1 and 2, 2 and 3, 3 and 4, 4 and 5, 5 and 6, are all exposed. This gives rise to:

$$boundcost(x) = 0.5 * (100 + 100 + 100 + 100 + 100) = 250$$

The shortfall cost:

The shortfall depends on the target representation. Since neither target has a minimum representative area, the amount in the portfolio is the sum of all distributions in selected units.

Target	Amount
Trees	$Amount(x) = 30 + 30 + 0 = 60$
Grass	$Amount(x) = 0 + 0 + 800 = 800$

The shortfall cost imposed by each target is calculated as follows:

$$Shortfall\ penalty = (Penalty\ Factor) \left(\frac{Shortfall\ Amount}{Goal\ Amount} \right) (Penalty\ Cost)$$

Target	Shortfall Cost
Trees	$Shortfall\ Penalty = (1.0)(0/50)(400) = 0$
Grass	$Shortfall\ Penalty = (1.5)(200/1000)(200) = 60$

$$Shortfall(x) = 60$$

The overall cost:

$$Cost(x) = base(x) + boundary(x) + shortfall(x) = 350 + 250 + 60 = 660$$

There is an interactive Excel worksheet distributed with SITES, called **SITES-CostMockup.xls** that allows you to experiment with the cost function in this situation.

3. Methodology: Simulated Annealing

3.1 Overview

Simulated annealing is the name for a general algorithm to find the general minimum value of a “mystery function” (Figure 3.1). Simulated annealing has proven to be an effective way of approaching many computationally difficult problems, including ecoregional portfolio assembly. In SITES’s case, this algorithm is used to search for the portfolio that produces the lowest value of the cost function described in Chapter 2.

3.1.1 Annealing

The name is derived from the process of a slowly changing state in materials such as water freezing from a liquid into a solid. When the speed of the temperature drop is carefully controlled in order to arrive at a near-ideal final crystalline state, it is known as annealing.

Annealing: The technique of slowly cooling a liquid into a solid such that its final form is a near-optimal crystal.

3.2 The Annealing function

This metaphor is extended to the SITES cost function, $f(x)$. If we treat x as a potential portfolio, then $f(x)$ is the value of the cost function for that portfolio. Because it is extremely difficult to predict anything about the impact of any given unit on the total portfolio cost, the SITES cost function is used as the mystery function for simulated annealing.

Although SITES uses simulated annealing with its own cost function, $f(x)$ could be any function that takes in a “state” and returns a single value we can refer to as “cost,” making annealing a very flexible technique for computationally daunting problems.

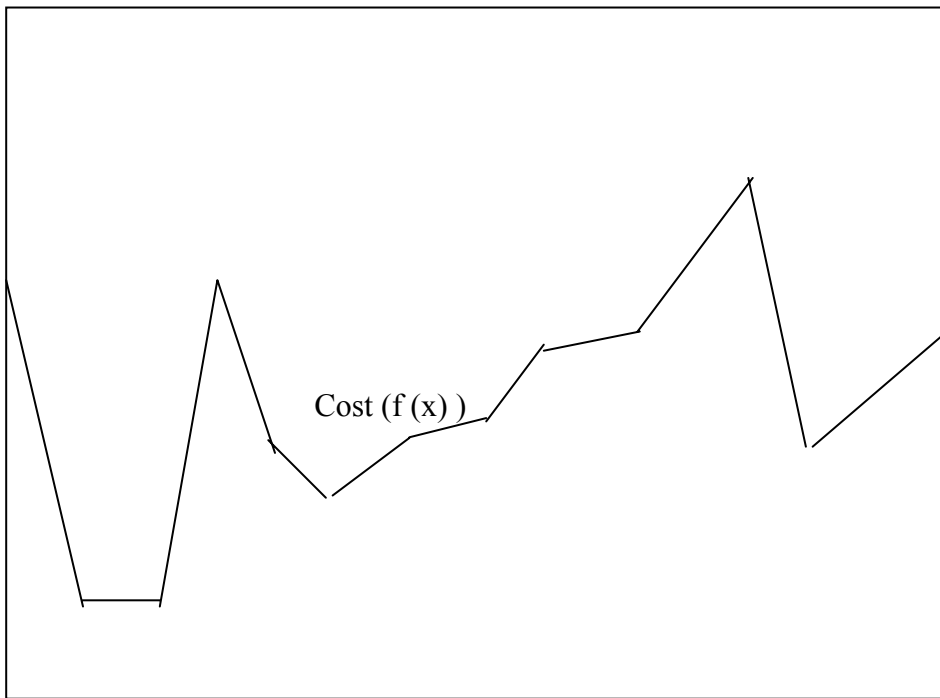


Figure 3.1: Annealing function

3.3 Hill descent

It would be impossible to evaluate every possible portfolio to determine the best configuration, since even a simple SITES portfolio assembly problem with 300 analysis units would have combinations, which is more than most estimates of the number of atoms in the entire universe. Instead, we can generate a starting portfolio randomly and evaluate $f(x)$ to determine the cost. From there, a simpleminded approach would be to generate another portfolio by making small random changes to the current one and accept them only if the changes made an improvement to the cost. This would continue for many iterations until no further improvements are possible.

This technique is called hill descent¹, and is closely related to simulated annealing. However, this algorithm will get stuck when it reaches a point where all small changes will result in higher cost, but a better solution exists elsewhere. This is called a local minimum. In Figure 3.2, B is a local minimum, while 4 is the absolute lowest minimum.

Iteration: A single change to the current state and re-evaluation of the cost function.

Local minimum: A low point in the cost function, but not the absolute lowest point.

¹ This is usually called hill *climbing* in the literature, but the name has been changed to reflect our search for minimums rather than maximums.

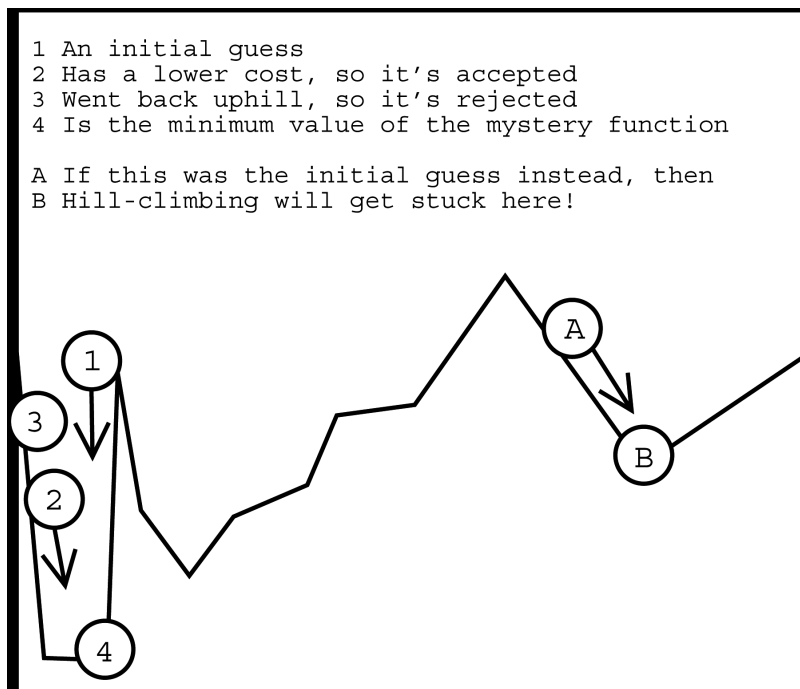


Figure 3.2

One way to visualize the process of hill descent is to imagine the cost function as a landscape. Each position represents a different portfolio, and the elevation at that point represents the cost. The current portfolio is like a ball that can only roll downhill, which means that it will quickly seek the lowest nearby value. However, if it starts in the wrong place, it will end up in a local minimum instead of the absolute minimum.

3.4 Simulated annealing

Simulated annealing is designed to avoid being stuck at a local minimum at the expense of increased runtime.

The simulated annealing process can also be imagined as a ball (the current solution) exploring the landscape, but rather than simply rolling downhill it bounces randomly. How the current portfolio changes (where the ball bounces) is driven by a factor called the temperature, which is analogous to the real-world temperature during physical (non-simulated) process of annealing.

At first, when the temperature is high, the ball will cover large distances and cross over hills easily. (Figure 3.3) As the algorithm progresses, the temperature is lowered and the annealing process is less likely to accept large cost increases – the ball is less likely to make large jumps uphill. (Figure 3.4)

Temperature: In simulated annealing, this defines the maximum allowable change in the cost function.

By lowering the temperature at an appropriately slow rate, depending on the size and steepness of the landscape, the solution has a much better chance of settling into the lowest possible point – much as slowly cooling a material will lead to an optimal crystalline structure.

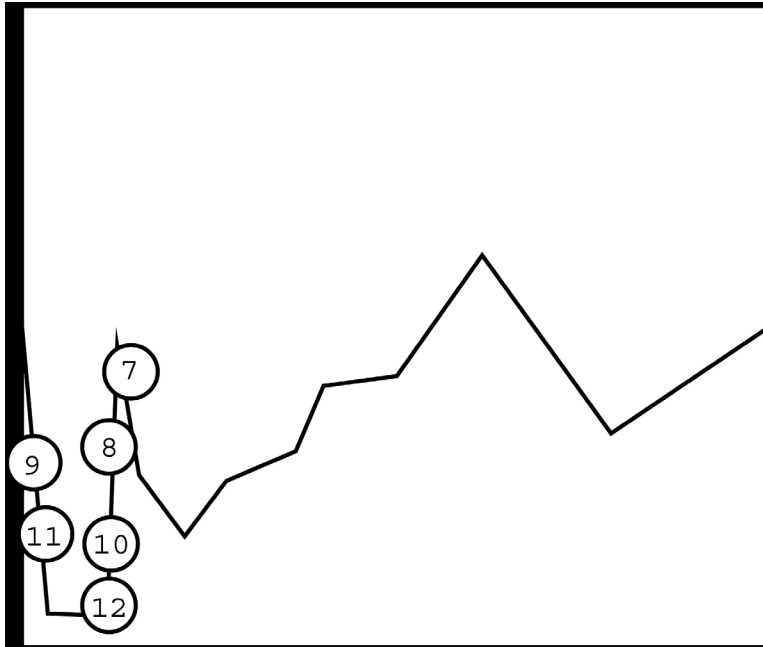


Figure 3.3

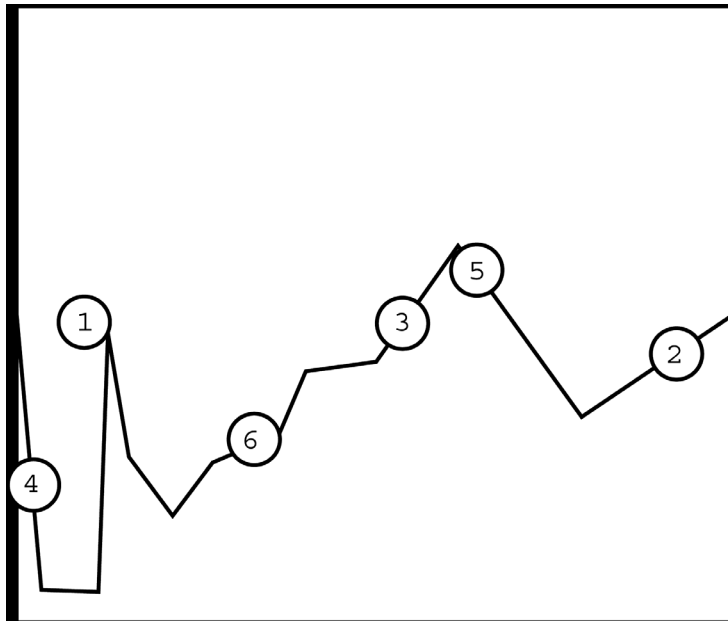


Figure 3.4

3.5 Considerations

3.5.1 Iterations

With this process of simulated annealing, the main concern is that enough iterations are specified, that the cost function adequately explores possible portfolios, and doesn't get stuck in a local minimum. Defining "enough" can be tricky. Much depends on the complexity of the cost function: A function with very steep and tall features (Figure 3.5) where the cost often changes drastically with only a small change to the portfolio will take more iterations than one that changes smoothly and has a broad, well-defined minimum (Figure 3.6).

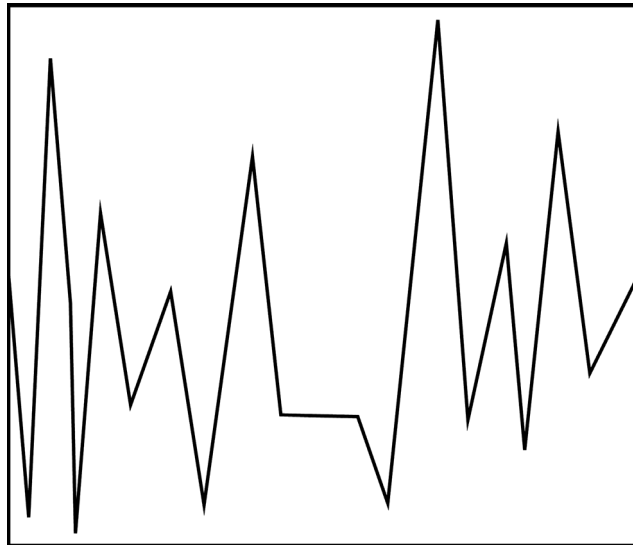


Figure 3.5

Balance

If all three terms in the SITES cost function are important to the final result, it's a good idea to be careful that their contribution to the portfolio cost is relatively equal. If any one term is weighted too heavily, the portfolios generated by SITES may find a reasonable way of satisfying that term but never reach a solution that also performs well on the less significant factors. (This is also closely related to the difficulty of the cost function, since factors that are over-weighted will increase the function's sensitivity to small changes.)

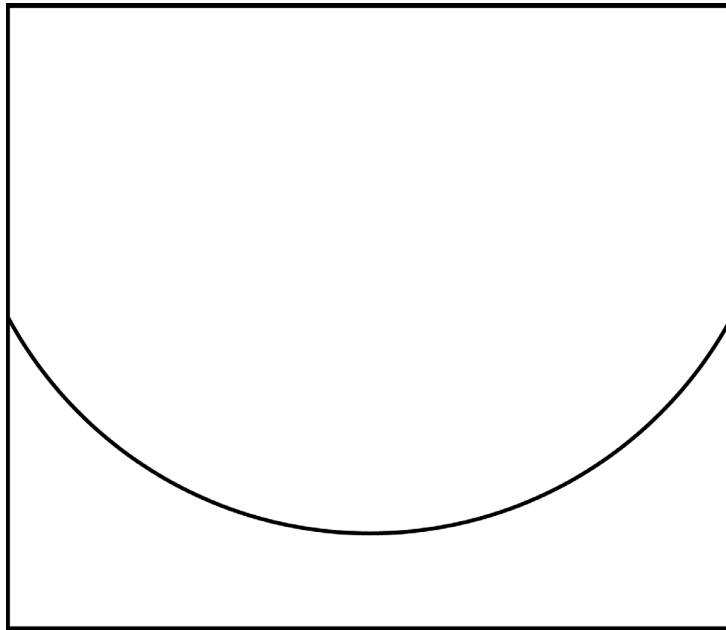


Figure 3.6

Terms Used:

- Analysis unit: analysis units form the portfolio; they are small areas that are marked as in or out to create portfolio.
- Annealing: the technique of slowly cooling a liquid into a solid such that its final form is a near optimal crystal.
- Base cost: a component of the SITES cost function that encourages SITES to minimize the area of the portfolio. It is the sum of the cost specified for each analysis unit included in the portfolio.
- Boundary cost: a component of the SITES cost function, aimed toward minimizing a portfolio's fragmentation by minimizing the length of its boundary
- Boundary length: The length of a portfolio's perimeter.
- Boundary length modifier: a multiplier that converts and scales the boundary length of a portfolio before adding it to that portfolio's cost.
- Core: the SITES module that performs the annealing process to create a portfolio.
- Cost function: the cost function for SITES calculates a single cost value for a given portfolio that represents its effectiveness.
- Iteration: a single change to the current state and re-evaluation of the cost function.
- Local minimum: a low point in the cost function, but not the absolute lowest point.
- Minimum area: the minimum contiguous amount of a target required for it to contribute to a conservation goal.

Penalty factor: sets the importance of representing a target, relative to other targets and the base and boundary costs.

Shortfall cost: a component of the SITES cost function that penalizes portfolios that don't meet conservation goals.

Simulated annealing: A general technique for finding the lowest value of a function through many trial runs and repeated adjustment to input values.

Target: a biological feature with a conservation goal that SITES attempts to meet during assembly.

Target patch: an occurrence of a target that can spread over several neighboring analysis units.

Temperature: in simulated annealing, this defines the maximum allowable change in the cost function.