A sampling scheme for Peirson's milk-vetch in the Algodones Dunes

FRAN EVANISKO
Lead cartographer
Bureau of Land Management
California State Office and
Adjunct faculty member
American River College
e-mail: fevanisk@ca.blm.gov

Abstract: Peirson's milk-vetch is an endangered plant species that occurs on the Algodones Dunes in the Imperial Valley of California. John Willoughby, a BLM botanist, devised a systematic ransom sampling design for monitoring the occurrence of the plant over time. The sampling is stratified by 12 sample areas within eight management areas. The design requires 135 transects with a total length of 930 km and 37,169 sample points 25 meters apart. This paper discusses the algorithm used to generate the sampling scheme, and the code required to implement the design in both AML and ArcObjects. The two methods are compared in terms of complexity and performance.

Background

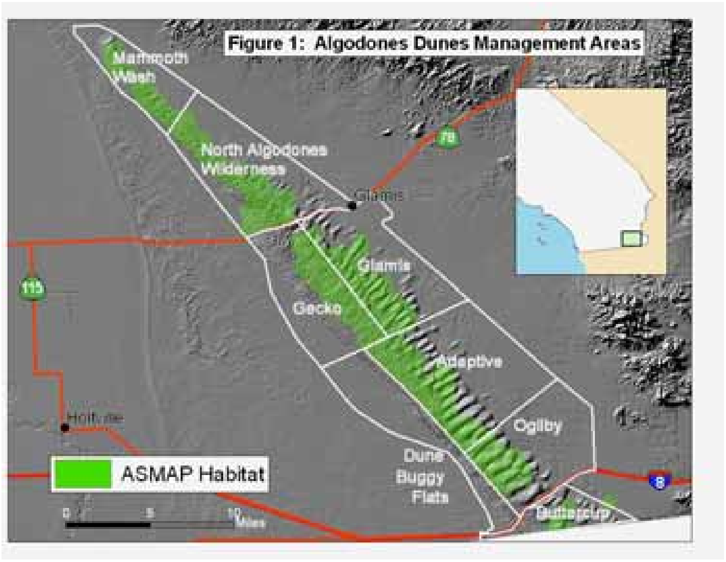
The BLM has been monitoring the density and population size of Pierson's milk-vetch (*Astragalus magdalenae* var. *peirsonii,* hereafter referred to as ASMAP), and several other plant species since 1998 (BLM, 2000). In 2004, because of potential conflicts between sensitive plant species, ASMAP, a federally-listed threatened species, and off-highway vehicle use, the Bureau of Land Management (BLM) implemented an ambitious monitoring program in the Algodones Dunes (also known as the Imperial Sand Dunes). .

Sampling was stratified by eight management units identified by the Imperial Sand Dunes Recreation Management Plan (ISDRAMP) (BLM, 2005). Monitoring has been conducted in seven of the eight management units in which ASMAP occurs. Furthermore, sampling units were placed so that the major part of the ASMAP habitat was encompassed within the sampling areas. Figure 1, depicts the relationship between the ISDRAMP management units, and the ASMAP habitat.

This paper describes the construction of the sampling schema for the monitoring program, according to requirements specified by the BLM California State Office (BLM CASO) botanist, John Willoughby. Fran Evanisko, the BLM CASO staff cartographer, constructed production datasets using Arc Macro Language (AML) procedures. Subsequently, Fran Evanisko programmed Visual Basic for Applications (VBA) procedures employing ArcObjects to construct a comparable schema in a personal geodatabase. This paper describes the AML approach and the ArcObjects approach, and tests the performance of each method on similar datasets.

Objective

The objective of this project was to create sample points along a set of transects normal to a set of provided baselines at a specified sampling interval of 25 meters. Each sample point having eastings and northings associated with them representing the UTM Zone 11, NAD 83 coordinates, and each transect will having a grid (UTM Zone 11 NAD 83) bearing associated with it. The resulting sample points had to be made available to the monitoring crews as a shapefile feature class.



Construction of sample areas and baselines

Sample areas and baselines were constructed using primarily interactive methods in ArcGIS. John Willoughby inspected the relationship between the management areas and the ASMAP habitat to determine the orientation and width of individual rectangular sample areas. Linear features representing the centerlines of sample areas were digitize into a feature class. These centerlines were buffered by a distance one-half of the width of each sample area, using the flat end option. This resulted in a set of rectangular sample area polygons situated appropriately within each management unit.

Baselines, that would serve as a starting line for parallel transects within each sampling unit, were constructed by selecting the northern bounding line of each sample area rectangle and copying them into a baseline feature class. All of the baselines were directed in a southwest to northeasterly direction.

These baselines would serve as input to procedures that constructed linear features representing transects in both the AML and VBA/ArcObjects programs. Figure 2 depicts the relationship between the management areas, sample areas and the baselines.

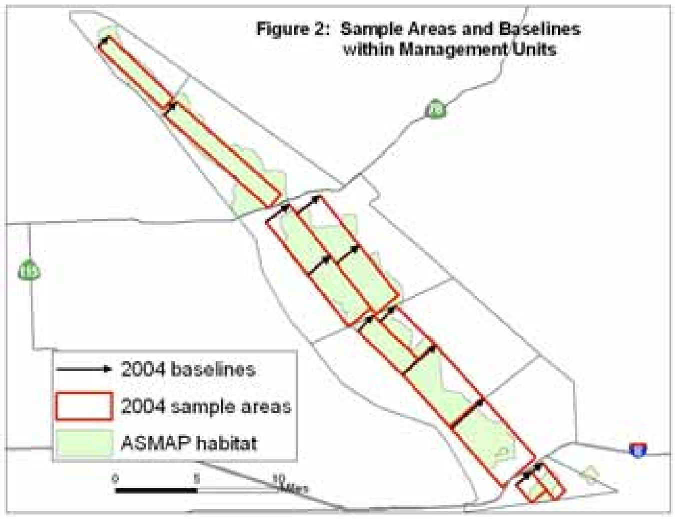


Figure 2: Sample Areas and Baselines within Management Units
78
115
8
2004 baselines
2004 sample areas
ASMAP habitat
0
5
10
Miles

Requirements for the transects and sample points

John Willoughby provided requirements for transect length and the number of transects for each sample area. Given the sample areas, transect lengths and number of transects the goal was to construct the transects and to construct sample points 25 meters apart along each transect. The coordinate system for these data would be UTM Zone 11, NAD 83.

The starting point for the first transect, within each sample area, was based on a random offset from the end of each sample area baseline.

Further requirements included:

1. An indexing system to relate sample points to transects, transects to baselines, and baselines to sample areas,
2. associating UTM coordinates with each sample point, and
3. associating bearings with each transect.

Computations required for constructing transects and sample points

1) The transect interval for each baseline was calculated as:
   $i = l/n$ Where: i is the transect interval for a baseline
   l is the baseline length
   n is the number of transects for a baseline

2) The random offset for the first transect on each baseline was calculated as:
   $o = int((i+1) * Rnd())$ Where: o is a random integer between 0 and i,
   i is the interval transect for a baseline.

3) The inclination of a transect relative to a baseline was calculated as:
   $a = b - (\pi/2)$ Where: a is the inclination of a transect (assuming the transect is turned right).
   b is the inclination of a baseline.

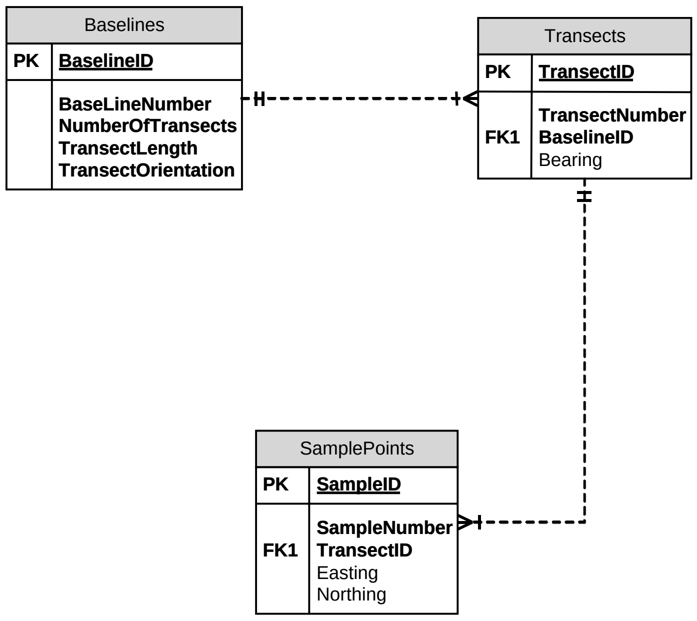
4) Coordinates for a point (x, y) at a distance t along a line from a point (x_1, y_1) with inclination a were calculated as:
   $x = x_1 + (t * Cos(a))$
   $y = y_1 + (t * Cos(a))$

5) The inclination (a) of a line trough points (x_1, y_1) and (x_2, y_2), required by the AML method only, was calculated as:
   $a = Atan((y_2 - y_1) / x_2 - x_1))$

The logical data model

Figure 3 is an entity relationship diagram illustrating the data model for the sampling schema. This model was used to develop the ArcInfo coverages as well as the personal geodatabase feature dataset. Although, in the coverage feature classes underlines "-" were used as separators in the item names, since INFO cannot support the camelback notation.

Figure 3: Logical Data Model for Constructing Sampling Schema



Arc Macro Language (AML) Procedures

The AML procedures resulted in three separate ArcInfo coverages; baselines, transects, and sample points. The program is primarily a linear procedural program using, for the most part, available ArcInfo Commands. Following is a description, in outline form, of the procedures that make up the program. ArcInfo commands are in capital letters.

Transect Construction:

- BUILD node topology on baselines
- Used ADDXY command to associate coordinates with baseline nodes
- Created RELATE between FromNodes and ToNodes and the baselines
- For each baseline feature
    - Computed the baseline inclination
    - Computed the interval between transects
    - Computed the offset for the first transect
    - For Each Transect
        - Computed x and y coordinates for the beginning and end of the transect
        - Wrote the TransectID, and beginning and endpoint coordinates to a text file (transect.gen)
        - Wrote the TransectID, TransectNum, and BaselineID to a text file (transect.att)
- Used the GENERATE command to create the transect coverage
- Used PROJECTCOPY to associate the UTM 11 coordinate system with the transect coverage
- In TABLES, created the transect attribute table and added records from transects.att
- Used JOINITM to permanently join the transects attribute table to the transects feature class
- Used ARCCOGO to add cogo attributes to the transect line features, and used DROPITEM to eliminate all items except the bearing

Sample Point Construction:

- Built node topology on the transects
- Used ADDXY to associate coordinates with the transect nodes
- Created relates between FromNodes and ToNodes and the transects
- For each transect
    - Calculated the transect inclination
    - From 0 to the transect length
        - Calculated X and Y for each sample point
        - Wrote SampleID, X, and Y to a text file (samples.gen)
        - Wrote SampleID, SampleNum, TransectID, BaselineID to a text file (samples.att)

- Used the GENERATE command to create the samples coverage
- Used PROJECTCOPY to associate the UTM 11 coordinate system with the samples coverage
- In TABLES, created the samples attribute table and added records from samples.att
- Used JOINITM to permanently join the samples attribute table to the samples feature class
- Used ADDXY to associate coordinates with the sample points.
- Clean up temporary files.

ArcObjects Procedures

The ArcObjects procedures were programmed using VBA in the ArcMap application. These procedures resulted in three feature classes; Baselines, Transects, and Sample points within a personal geodatabase called SamplingSchema. The SamplingSchema database was set up interactively in ArcCatalog. The procedures however, constructed the Transects and SamplePoints feature classes. The VBA program is modular in form. It was advantageous, if not necessary, to program a series of functions and subroutines using ArcObjects. Following are descriptions of the functions and subroutines. A routine programmed on a UIButton control's click event called the procedures in sequence to construct and populate the geodatabase feature classes.

UIBtnMakeSamples_Click: - Subroutine
Input: None
Does:
1. Deletes the transect and sample point feature classes should they exist
2. Calls CreateTransectsFeatureClass
3. Calls CreateSamplePointsFeatureClass
4. For each baseline feature
    - Calls ConstructTransects
    - Calls AddBaselineTransects
5. For each transect feature
    - Calls ConstructSamplePoints
    - Calls AddTransectSamples

CreateTransectFeatureClass - Function
Input: FeatureDataset
Output: Boolean
Does:
1. Creates a transect feature class, with fields, in the feature dataset.
2. Returns a Boolean (True if successful, False otherwise).

CreateSamplePointFeatureClass - Function
Input: FeatureDataset
Output: Boolean
Does:
1. Creates a sample point feature class, with fields, in the feature dataset.
2. Returns a Boolean (True if successful, False otherwise).

ConstructTransects - Function
Input: Baseline feature

Output: Geometry enumeration
Does:
1. Computes offset for first transect
2. Computes beginning and ending point for each transect
3. Places transect polyline in a GeometryBag
4. Returns a Geometry Enumeration

AddBaselineTransects - Subroutine
Input: Baseline number
Transect enumeration
Transect feature class
Does:
1. Creates a polyline feature for each transect
2. Assigns attributes to each transect feature (BaselineNumber, TransectNumber, and Bearing).
3. Adds transect feature to the transect feature class.

ConstructSamplePoints - Function
Input: Transect feature
Sample spacing
Output: Geometry enumeration
Does:
1. Computes coordinates for each sample point
2. Places sample points into a GeometryBag
3. Returns a Geometry Enumeration

AddTransectSamples - Subroutine
Input: Baseline number
Transect number
Sample point Enumeration
Sample point feature class
Does:
1. Creates a point feature for each sample point.
2. Assigns attributes to each transect feature (SampleNumber, BaselineNumber, TransectNumber, Easting and Northing).
3. Adds point features to the sample point feature class.

GetBearingFromInclination - Function
Input: Inclination Angle (in radians)
Output: Bearing
Does:
1. Converts inclination angle to a bearing, in the form "N32-24-15E".
2. Returns a bearing

GetLineBearing - Function
Input: Line feature
Output: Bearing
Does:
1. Gets the line's inclination angle
2. Calls GetBearingFromInclination
3. Returns a bearing for the line feature

Results Comparison

The AML program excluding comments and white space has 194 lines of code. It took about one full work day for Fran to compose this program. The VBA/ArcObjects program has 498 lines of code excluding comments and white space. It took about 5 – 6 full workdays to compose this program. However, Fran was an accomplished and experienced AML programmer, and a novice ArcObjects programmer, at the time of this work.

On May 30, 2005, Fran ran both of these programs against the 2004 baseline feature classes at the BLM California State Office. This test was carried out on a Dell computer with an INTEL, XEON, 2.20 Ghz processor with 1GB RAM. The ArcMap project containing the VBA code, the personal geodatabase, the AML program and Baseline coverage were stored locally. The AML took 12 minutes and 33 seconds to complete, and the VBA/ArcObjects program took 8 minutes and 45 seconds to complete.

About one year ago on May 16, 2004 Fran conducted a similar test at the American River College, Science and Engineering Department's Computer Lab. This task was carried out on a Gateway computer with an INTEL, PENTIUM IV, 1 Ghz processor with 512 MB RAM. The AML took 17 minutes and 50 seconds. The VBA/ArcObjects took 15 minutes and 11 seconds.

Figures 4-6 illustrate feature classes produced by both the AML and VBA/ArcObjects programs, during the 2005 performance test.

Figure 4: Test baselines and transects

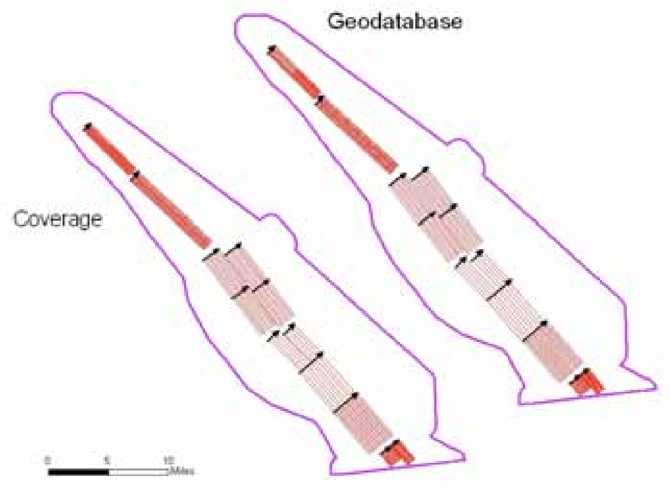


Figure 4 depicts the relationship between the Imperial Sand Dunes Recreation Management Area Boundary, the 2004 baselines and the transects generated by both programs. Figure 5 depicts a potions of transects in the Buttercup Management Area. Note that both methods resulted in bearings with some angular variability for transects turned from a single baseline. Finally, Figure 6 depicts the relationship between portions of Buttercup Management Area transects and sample point locations with associated UTM Zone 11 NAD 83 coordinates. These data are from the geodatabase version, but the coverage sample points have similar attributes.

Figure 5: Some transects in the Buttercup Management Area

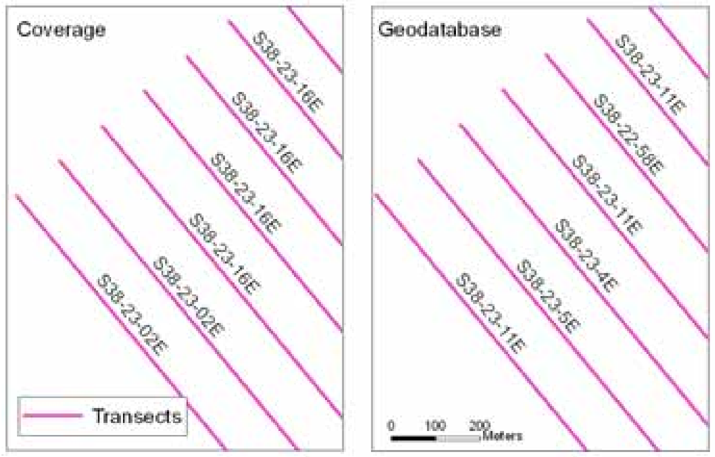
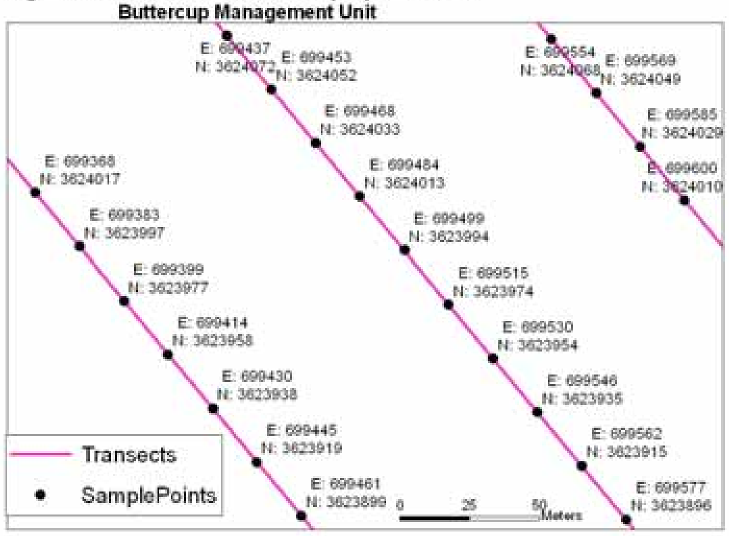


Figure 6: Some transects and sample points in the Buttercup Management Unit



Clearly for this task, the ArcObjects approach appears to be more efficient. The modular character of the ArcObjects approach is an added benefit. For example, Fran has generalized the angle conversion routines to create a DirectionConversion model that includes four functions "GetAzimuthFromInclination", "GetBearingFromInclination", "GetInclinationFromAzimuth", "GetInclinationFromBearing". These functions will accept or produce inclinations in either decimal degrees or bearings.

Literature Cited

Willoughby, J.W. 2000. Monitoring of special status plants in the Algodones Dunes, Imperial County, California: Results of 1998 monitoring and comparison with the data from WESTEC's 1977 monitoring study. Bureau of Land Management, California State Office, Sacramento, CA.

Willoughby, J.W. 2005. 2004 Monitoring of special status plants in the Algodones Dunes, Imperial County, California. Bureau of Land Management, California State Office, Sacramento, CA.