**Cook East Georeference Point creation methodology, 1998:**

From “RESOLUTION OF DIGITAL ELEVATION MODELS AND TERRAIN ATTRIBUTES, APPARENT ELECTRICAL CONDUCTIVITY, AND PREDICTION OF PALOUSE SOILSCAPE Bt HORIZONS”, Masters of Science, David Uberuaga, May 2004, Washington State University.

“A 30.48-m grid was superimposed over the southern 36.4 hectares of the Washington State University (WSU) Cunningham Agronomy Farm (CAF). A random point was generated in each cell of the grid using MapInfo (RedHen Software Inc.) creating 369 geo-referenced points (Figure 2.1).”

36.4 hectares = 364000 square meters

Density of Cook East sampling points is 10.137 points per hectare, or 1 point ever 986.44 square meters

**Cook West Georeference Point creation methodology, 2017:**

Produced a 30.48 by 30.48 meter grid running in a true north south orientation and superimposed over the Cook West field boundary. The field boundary area was used to clip the grid producing a grid with many cells on the boundary smaller than the 30.48 by 30.48 meters. One point was randomly placed within each full grid cell producing 292 potential sampling points. Points were then manually removed from many of the smaller cells formed along the boundary of the area resulting in 250 points. Points were also randomly moved within grid cells when the original placement overlapped roads, excavations conducted to map tile drains, known tile drain locations or any other infrastructure.

Sampling locations 250

Area = 23.4 hectares = 234000 square meters

10.68 points per hectare, 1 point per 936 meters.

Randomized points for Cook West generated in ArcGIS Desktop version 10.4.05524:

1. Create a grid over the area of interest using the Fishnet tool in ArcGIS. The height and width of the grid should be predetermined to produce the desired density of sampling locations per area if one sampling point is randomly paced within each grid.
2. Convert the boundary for the area of interest into a polygon feature if it is a polyline feature.
3. Clip the grid using the boundary polygon feature of the area of interest, using the Clip (Analysis) tool.
4. Convert the clipped polyline grid into a polygon feature using the Feature to Polygon tool. Use both the clipped poly line grid and the polyline boarder for the same area of interest. The result should be a polygon feature with multiple polygons representing the clipped grid.
5. There are a number of ways to generate random points. For this a custom ArcToolbox script was used "Generate Random Points"<http://ianbroad.com/> , the built in Create Random Points tool can work as well.

#----------------------------------------------------------------------

# Purpose: Generates a user defined number of random points inside

# each polygon of an input Polygon feature class.

#

# Author: Ian Broad

# Website: www.ianbroad.com

#

# Created: 02/27/2015

#----------------------------------------------------------------------

import arcpy

import random

import sys

arcpy.env.overwriteOutput = True

polygon = arcpy.GetParameterAsText(0)

use\_field = arcpy.GetParameterAsText(1)

field = arcpy.GetParameterAsText(2)

number\_of\_points = abs(int(arcpy.GetParameterAsText(3)))

min\_distance = abs(float(arcpy.GetParameterAsText(4)))

attempts = abs(int(arcpy.GetParameterAsText(5)))

keep\_attempts = arcpy.GetParameterAsText(6)

output = arcpy.GetParameterAsText(7)

if "in\_memory" in output:

mem\_name = output.split("\\")[-1]

else:

mem\_name = "mem\_point"

mem\_point = arcpy.CreateFeatureclass\_management("in\_memory", mem\_name, "POINT", "", "DISABLED", "DISABLED", polygon)

arcpy.AddField\_management(mem\_point, "PolygonOID", "TEXT")

result = arcpy.GetCount\_management(polygon)

features = int(result.getOutput(0))

arcpy.SetProgressor("step", "Generating Random Points...", 0, features, 1)

fields = ["SHAPE@", "OID@"]

if use\_field == "YES":

fields.append(field)

with arcpy.da.InsertCursor(mem\_point, ("SHAPE@", "PolygonOID")) as insert:

with arcpy.da.SearchCursor(polygon, (fields)) as search:

for row in search:

try:

polygon\_geom = row[0]

oid = row[1]

count = 0

attempt = 1

distance = []

xmin, xmax= row[0].extent.XMin, row[0].extent.XMax

ymin, ymax = row[0].extent.YMin, row[0].extent.YMax

if use\_field == "YES":

number\_of\_points = int(row[2])

while count < number\_of\_points:

random\_x = random.uniform(xmin, xmax)

random\_y = random.uniform(ymin, ymax)

point = arcpy.Point(random\_x, random\_y)

point\_geom = arcpy.PointGeometry(point, polygon\_geom.spatialReference)

contains\_point = polygon\_geom.contains(point\_geom)

if keep\_attempts == "NO":

if attempt < attempts:

if contains\_point == True and count == 0:

distance.append(point\_geom)

insert.insertRow((point\_geom, oid))

count += 1

elif contains\_point == True and count > 0:

distance\_check = True

for point in distance:

if point\_geom.distanceTo(point) > min\_distance:

pass

else:

distance\_check = False

attempt += 1

if distance\_check == True:

distance.append(point\_geom)

insert.insertRow((point\_geom, oid))

count += 1

else:

arcpy.AddError("Failed attempting to generate {0} random points for Polygon OID: {1}".format(attempts, oid))

arcpy.AddError("Decrease number of random points or the minimum distance and try again.")

sys.exit(0)

if keep\_attempts == "KEEP\_ALL":

if attempt < attempts:

if contains\_point == True and count == 0:

distance.append(point\_geom)

insert.insertRow((point\_geom, oid))

count += 1

elif contains\_point == True and count > 0:

distance\_check = True

for point in distance:

if point\_geom.distanceTo(point) > min\_distance:

pass

else:

distance\_check = False

attempt += 1

if distance\_check == True:

distance.append(point\_geom)

insert.insertRow((point\_geom, oid))

count += 1

else:

arcpy.AddWarning("Only generated {0} out of {1} random points for Polygon OID: {2}".format(count, number\_of\_points, oid))

arcpy.AddWarning("Decrease number of random points or the minimum distance and try again.")

break

if keep\_attempts == "KEEP\_ONLY\_SUCCESSFUL":

if attempt < attempts:

if contains\_point == True and count == 0:

distance.append(point\_geom)

count += 1

elif contains\_point == True and count > 0:

distance\_check = True

for point in distance:

if point\_geom.distanceTo(point) > min\_distance:

pass

else:

distance\_check = False

attempt += 1

if distance\_check == True:

distance.append(point\_geom)

count += 1

if int(len(distance)) == int(number\_of\_points):

for geom in distance:

insert.insertRow((geom, oid))

else:

arcpy.AddWarning("Failed attempting to generate {0} random points for Polygon OID: {1}".format(number\_of\_points, oid))

arcpy.AddWarning("Decrease number of random points or the minimum distance and try again.")

break

arcpy.SetProgressorPosition()

except Exception as e:

arcpy.AddMessage(str(e.message))

if "in\_memory" in output:

arcpy.SetParameter(8, mem\_point)

else:

arcpy.CopyFeatures\_management(mem\_point, output)

arcpy.Delete\_management(mem\_point)

arcpy.ResetProgressor()

arcpy.GetMessages()