# Pivot Finder for ArcGIS

*A Tool for Optimally Placing Center Pivots on Agricultural Tracts*

Justin Dietrich
A00381468
05 December 2014

CEE 6440 GIS in Water Resources
Dr. Tony Castronova
Utah State University

# Pivot Finder for ArcGIS

*A Tool for Optimally Placing Center Pivots on Agricultural Tracts*

## Introduction

Smart agriculture is largely a question of land use optimization and water use optimization. Innovations in farming over the last couple decades have led to the adoption of center pivots for watering. Center pivots are the large sprinkler systems on wheels that turn in circles to water crops. Because of this circular motion, center pivots do not tend to fit well on most agricultural parcels. Corners of fields are often unreachable by the center pivot and are usually abandoned to weeds. There are nearly always barns, roads, and other obstructions which must be considered when placing a pivot on a tract of agricultural land. These complications usually lead planners to arbitrarily decide on the pivot's placement. This decision represents a lack of sufficient analysis that can lead to preventable inefficiencies in the pivot design.

Preventing these inefficiencies traditionally requires planners to spend lots of time with scaled maps considering many different options for center pivot size and placement. This huge time commitment is the reason most planners simply pick the size and placement for a pivot based on their intuition.

This is where the Pivot Finder for ArcGIS tool package comes in. This tool package is designed to assist planners in optimally placing pivots on agricultural tracts. Utilizing the tool package should allow users to complete a basic analysis of center pivot location in a matter of a few minutes. This tool package was created as my term project for my GIS in Water Resources class at Utah State University.

## Objectives

In order to be practical, the Pivot Finder tool package needed to be simple to use, efficient in its execution, and flexible for many situation. The tool package was developed with these requirements in mind. The tool was also designed in such a way that only the basic ArcGIS license is required. There is no need to purchase ArcGIS extensions to run the Pivot Finder tools.

Another requirement of practicality is that the tool be accessible to the professionals doing this kind of work. Accessibility is one major reason that the ArcGIS platform was chosen. While it is true that few farmers will have access to an ArcGIS license, the irrigation companies and government agencies involved in agriculture likely will.

## Functionality and Methodology

To date, the Pivot Finder for ArcGIS tool package includes a tool to assist in preparing the necessary input data and also includes two analysis tools. Because this report's purpose is to report on the term project, I will explain the methodology employed to make the tools work in addition to their functionality. This information will be presented within the context of learning to use the tools. The three tools are the Setup Tool, the Full Circle Pivot Tool, and the Land Quality Pivot Tool.

### *Accessing the Tool Package*

The first step for any professional would be to simply access the tool package. Fortunately, ArcGIS Tool Packages are easily transportable from user to user. Once saved to disk, ArcGIS for desktop can access the tool package through the catalog window (Figure 1). The tool package currently contains three scripts, each one representing a tool. The scripts were coded in python and heavily rely on arcpy tools as the scripts building blocks.

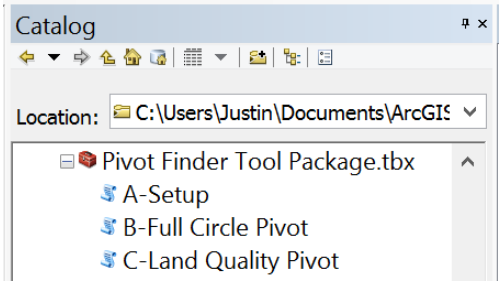



*Figure 1. Toolbox Access Point*

*Preparing the Data for Analysis*

The Setup Tool is designed to be run first and assists the user in preparing the data required for analysis. See Appendix B to view the Setup Tool's Code. This tool automates three main operations. First, the tool creates a geodatabase to store all related analysis files. Second, it automates the creation of three feature dataset folders and defines for them a coordinate system. ArcGIS tools work best when coordinate systems of the several input and output files are consistent. Using the Setup Tool ensures that the user will not run into any difficult errors or inaccuracies as a result of coordinate system problems. Third, the tool creates empty feature classes (which are analogous to shape files), in which the user will prepare the input data.

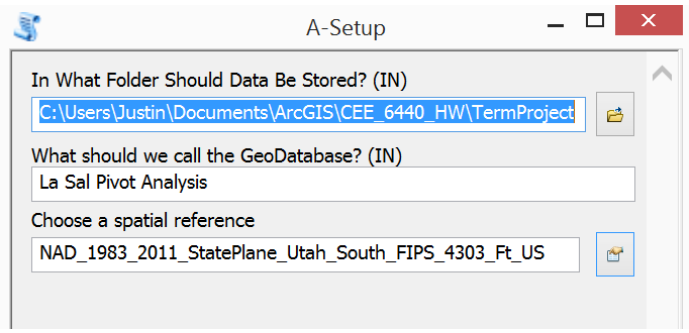



*Figure 2. Setup Tool Inputs*

The Setup Tool requires the user to provide three inputs. These inputs are: the folder on disk in which the user would like the tool to operate; a name for the geodatabase; and a spatial reference. Once these three inputs are provided, the user merely needs to select ok at the bottom of the dialog box.

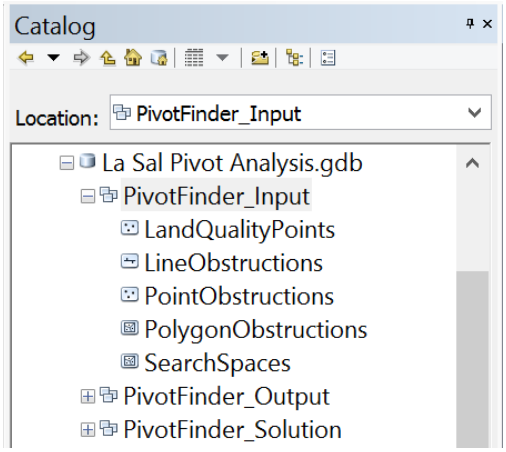
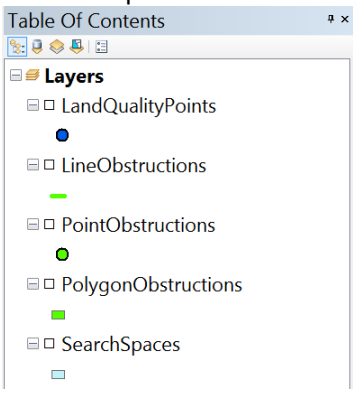



*Figure 3. Setup Tool Outputs*

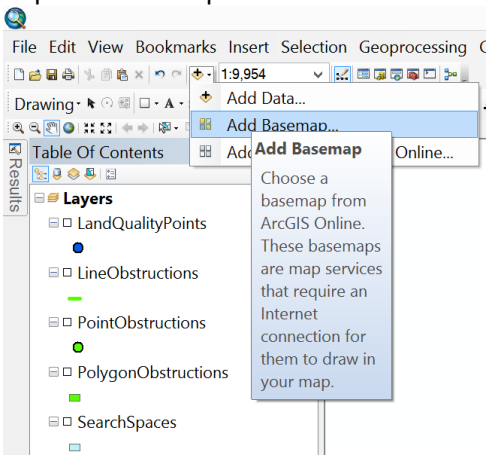
For the user, the most involved step of performing an analysis comes after running the Setup Tool and before running any of the analysis tools. This is because the step must be performed manually. It is not necessarily difficult (especially for regular ArcGIS users), but it may represent a small learning curve for some. The user will need to edit up to five feature classes, three of which represent obstructions, one represents land quality data, and the other indicates where the tool should search for the optimal pivot.

An obstruction defined in the tool as anything which would prevent a pivot from irrigating in a particular location. Examples of obstructions could be roads, buildings, trees, power poles, fences, or property boundaries. To complete this task, the user would need to execute this procedure:

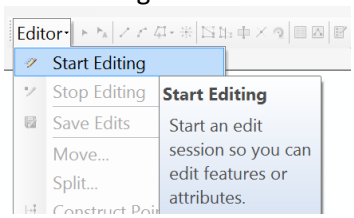
1. Add the input feature classes to the map.



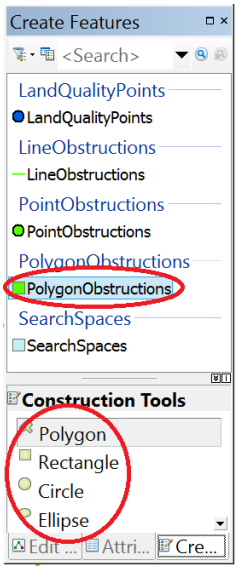

2. Import a basemap.



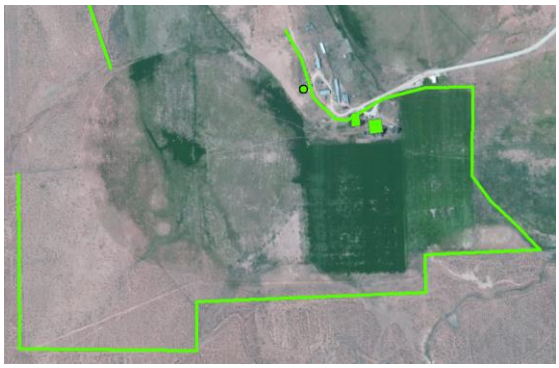
3. Start Editing.



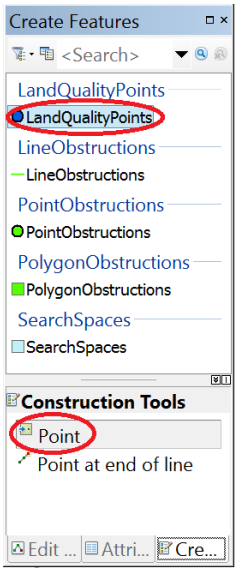

4. Select the appropriate obstruction feature class and the desired feature shape from the Create Features Window.



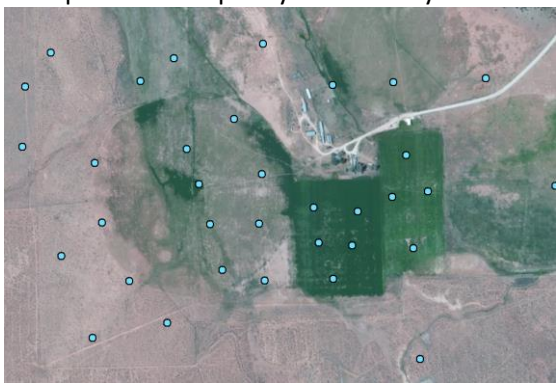

5. Draw Features over or around obstruction in the vicinity of the agricultural tract.
6. Repeat steps 3 and 4 all three obstruction feature classes if needed.



7. Select the Land Quality Points feature class from the create features window.




8. Place several points on and around the agricultural tract where land quality is known. The purpose of this is to incorporate the fact that all land on an agricultural tract does not necessarily have the same production value. One part of the land could be made from rich organic soil, while the other part could have a much higher clay content (and therefore yield less). See Figure A2 for an illustration of the need to incorporate land quality in the analysis.



9. Edit the LandValue attributes of the Land Quality Points feature class to reflect relative production value. Users might think of this as the value of the land around the point in dollars per square foot. Other users may prefer to simply value the land in relative terms (i.e. some land may be considered to be of full value and given a value of 1.0, while other land may be

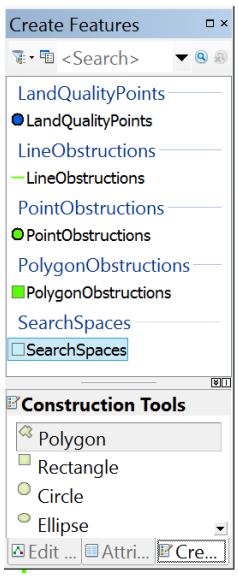
considered to be half value and given the value of 0.5). It does not matter what scheme is adopted for assigning value as long as the numbers chosen are relative to one another.

Table

LandQualityPoints

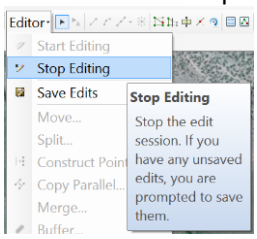
| | OBJECTID * | SHAPE * | LandValue |
|---|---|---|---|
| | 1 | Point | 1 |
| | 2 | Point | 1 |
| | 3 | Point | 1 |
| | 4 | Point | 0.2 |
| | 5 | Point | 0.2 |
| | 6 | Point | 0.4 |
| ▸ | 7 | Point | 0.4 |
| | 8 | Point | <Null> |
| | 9 | Point | <Null> |

10. Select the Search Spaces feature class and the desired feature shape from the Create Features Window.




11. Draw a polygon which represents the users "rough guess" at where the center of the optimal pivot is likely to be located. This does not need to follow any particular rules, but it helps the tool run with maximum efficiency by not needed to look for an optimal pivot in unlikely locations.

12. Save Edits and Stop Editing.




*Note: This set of instructions is not meant to be exhaustive. If more detailed instruction is needed for editing feature classes, seek help at ArcGIS online. A useful link has been provided in the reference section of this report*

Once the obstructions, land quality points, and search space are defined and saved, the user is ready to run the analysis tools on the data.

*Finding the Optimal Pivot by Maximizing Area*

The first analysis tool in the package is called Full Circle Pivot. It operates under the premise that the optimal pivot will be the pivot which can water the most land. In other words, this tool maximizes pivot coverage area. The user is required to provide eight inputs to the tool. This may seem like a lot of inputs, but they mostly consist of information previously defined or are the feature classes that were prepared previously. All that is required is for the user to navigate to the required input folder and select the proper feature.

Two new pieces of information requested on this input window are the Horizontal and Vertical Search Density. These inputs correlate to the degree of accuracy required by the analysis. For example, if the user would like the optimal center pivot location at an accuracy of plus or minus 50 feet in both directions, 50 would be placed in both input fields. The smaller the numbers provided in that field, the more accurate the solution, but the longer the tool will take to run. The Full Circle Pivot Tool is very efficient and has been tested at densities of 2 ft, and

analyzed over 23000 possible solutions in less than 25 seconds on an average laptop computer. This contrasts with the Land Quality Tool, which can only handle about 300 possible solutions at a time on an average laptop computer. Higher end computers with more RAM can process more possible solutions.

It is important for the user to choose the same spatial reference when running the tool as was chosen when running the Setup Tool.

Once the tool has finished, two solution feature classes will be written to the disk and located in the PivotFinder_Solution feature dataset folder. These solution feature classes are the Solution Center point feature class and the Solution Coverage polygon feature class. These can be added to the map for viewing. The attribute tables for these feature classes contain useful information for the planner such as pivot length and pivot coverage area. The attribute table even indicates the distance and direction to the nearest obstruction which may be valuable in defining the pivot's location on construction plans.

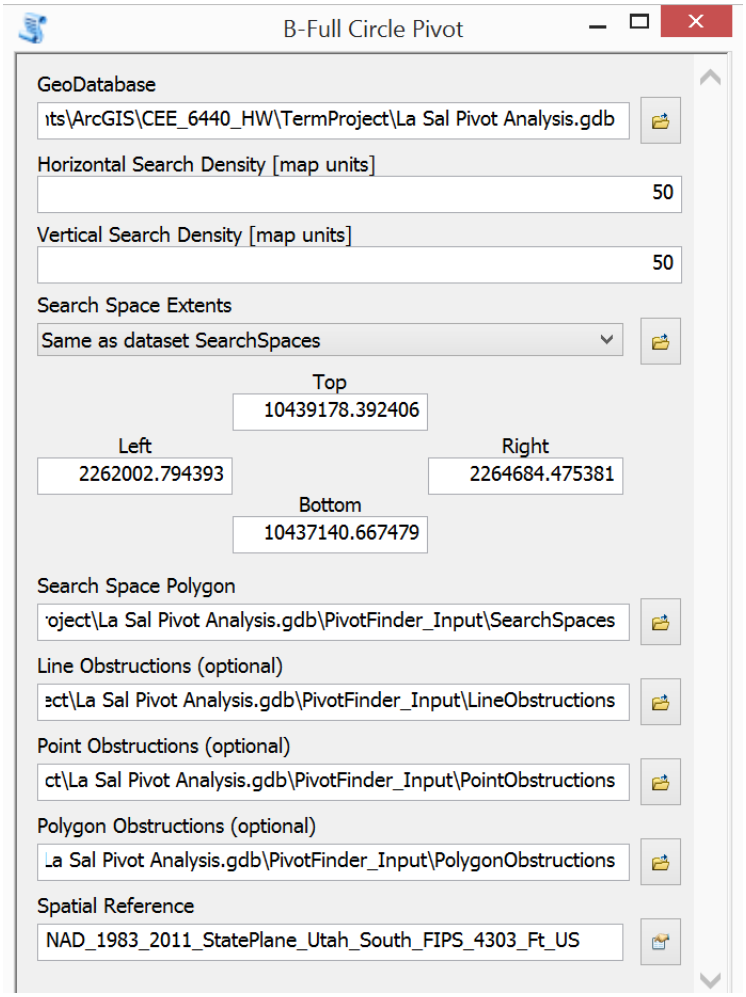



*Figure 4. Full Circle Pivot Tool Inputs*

In addition to the solutions, there are some output files which contain the tool's intermediate values. These files will have little value to most users, but may be utilized by a more experienced users for more advanced analysis.

Appendix C shows the Full Circle Tool's code.

*Finding the Optimal Pivot by Maximizing Production Value*

The second analysis tool in the package is called the Land Quality Pivot Tool. This tool maximizes pivot production value instead of simply maximizing gross area. Using this tool is ideal when there are large differences observed in land quality on the agricultural tracts in question. For example, it may be less valuable to a farmer or rancher to irrigate 300 acres of sandy, rocky soil than irrigate 100 acres of nutrient rich, organic soil. In that situation, maximizing area does not produce the optimum pivot location. However combining area with production value, optimal production can be obtained.

The inputs to the Land Quality Tool are identical to those of the Full Circle Pivot Tool with one exception. That exception is the Land Quality Points feature class. It can be added to the input dialog box in the same way that the other feature classes are added.

This tool operates under a certain limitation that is important to understand. Because the tool utilizes a complex ArcGIS function called Tabulate Intersection, it cannot handle a large amount of possible solutions at a time. The user is directed within the tool, that if the Land Quality Tool stalls, the user should cancel the operation and increase the search density and run the tool again. This easily allows the user to make an adjustments without the annoyance of re-entering all the inputs. The higher search density does cause the solution to be less accurate. However, the user can then adjust the search space to center around the solution obtained, then increase the search density and re-run the tool. After 2 or three repetitions, the user can still have a highly accurate, land quality adjusted solution.

Appendix C shows the Land Quality Tool's code.

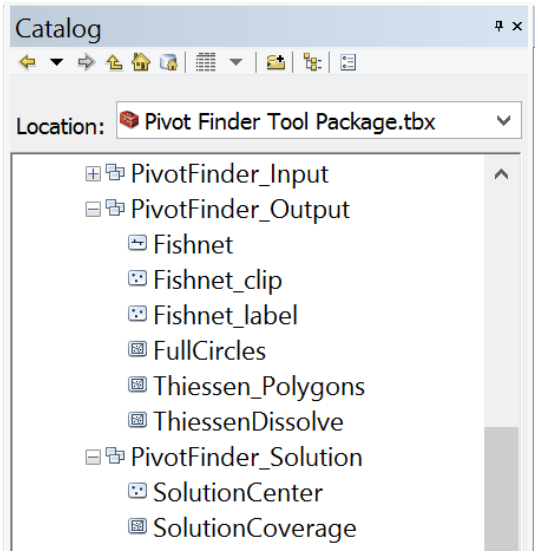



*Figure 5. Analysis Tool Outputs*

Results

The aggricultural plot we have been observing in this rport is located in LaSal, UT. The ranch is called Rattlesnake Ranch and produces alfalfa or pasture grass for beef cattle in the area. The rancher would like to replace the irrigation system in the area shown with a new center pivot. The results obtained from the two analysi tools is interesting. When considering only the gross area, the optimal pivot tends to drift west into the hilly sagebrush area. However, when land value is considered (as symbolized by the size of the green dots), the pivot moves back east to catch as much of the excellent land as possible without getting too small.

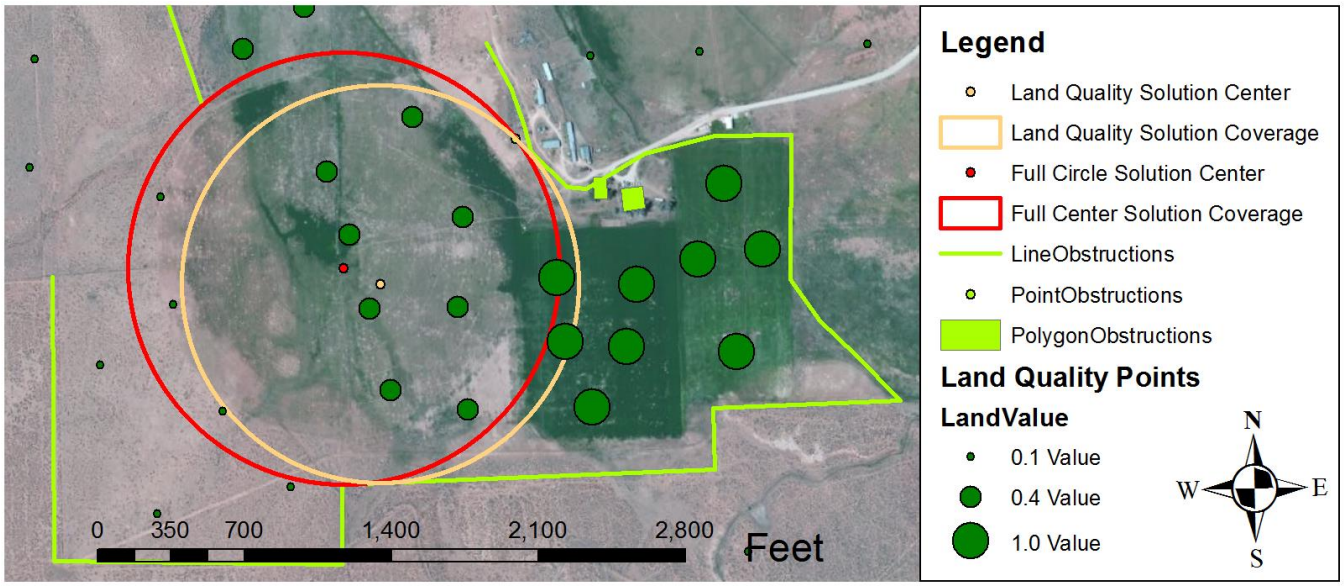



*Figure 6. Example Results Showing Contrast Between Full Circle and Land Quality Results*

Consider these results and their potential effect on agricultural efficiency. The Full Circle Pivot maximizes the available agricultural area and cuts down on un-used land. This represents a gain in efficiency over simply picking a pivot location at random. However, efficiencies improve further when considering the Land Quality Solution. The Land Quality Tool shows that smaller pivot shifted east a little will produce more crops than the larger pivot. This represents a gain in hydraulic efficiency since it will water a smaller area. Additionally, the smaller pivot will be cheaper to install and thus represent an economic efficiency for the farmer as well.

Discussion/Reflection

After many hours and hundreds of lines of code, the major question is whether these tools provide enough functionality for use in the real world. The answer to that question is that at least two more tools are needed for a full real-world application. Also, a few inefficiencies in the existing functionality need to be remedied.

The next tools which are needed are the Partially Swinging Pivot Tool (Figure A3) and the Multiple Pivots Tool (Figure A4). These tools would cover two major aspects of pivot design which have thus far remained unaddressed. However, once developed, these tools would allow users to execute a complete analysis of pivot placement in a matter of minutes. As the tool package sits now, users only have the ability to perform a *partial* analysis of pivot placement in a matter of minutes.

The inefficiencies that can be improved are manual input entry and the land quality density limitation. Data entry would be much enhanced if the script were to automatically pre-select the appropriate feature classes as the default inputs to the analysis tools. This would allow the user to simply select the analysis tool and define the search density instead of tediously copying over seven or eight input feature classes (that are always the same). The land quality density limitation, as mentioned, is caused solely by the Tabulate Intersection ArcGIS tool, which is used within the Land Quality Tool. It would be more efficient to loop over the tool once for each potential solution pivot instead of processing every potential solution pivot simultaneously.

Conclusion

Having considered the need for further development and the need for some upgrades to existing functionality, this tool still has acceptable performance. Any user with access to this design report, the Pivot Finder for ArcGIS Toolbox, and a basic knowledge of ArcGIS, should be able run the tools and interpret the results.

Smart agriculture requires that we utilize our land and our water as efficiently as possible. Optimizing the placement of center pivots on agricultural tracts is a key part of doing just that.

References

Esri. *A quick tour of editing*. ArcGIS Help 10.1. Accessed at http://resources.arcgis.com/en/help/main/10.1/index.html#/A_quick_tour_of_editing/01m5000000020000000 on 12/05/2014.

Appendix A: Conceptual Figures

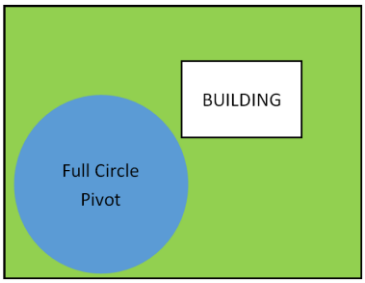



*Figure A1: Full Circle Pivot*

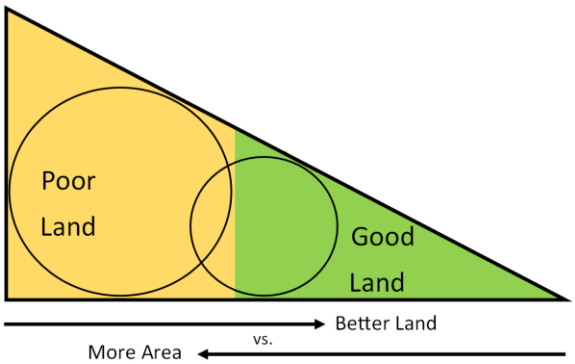



*Figure A2: More Area vs. Better Land*

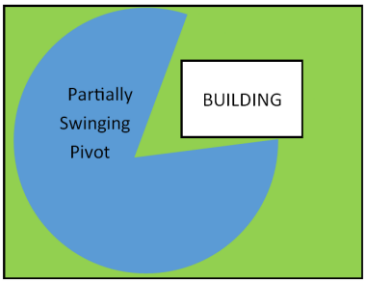



*Figure A3: Partially Swinging Pivot*

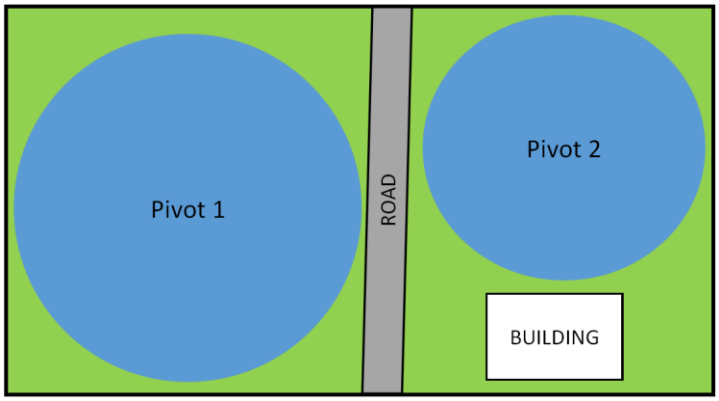



*Figure A4: Multiple Pivots*

## Appendix B: Setup Script

```
# -*- coding: utf-8 -*-
# ---------------------------------------------------------------------------
# Setup.py
# Created on: 2014-11-14
# Programmer: Justin Dietrich, Utah State University, Logan, UT, USA
# License: Use at own risk.
# Description: This script is intended for use with the Pivot_Finder ArcGIS Tool Package.
# Objective 1: Create an empty geodatabase to house tool inputs and outputs.
# Objective 2: Create feature dataSets to organize inputs and outputs.
# Objective 3: Create empty input feature classes.
# Learning Notes: It is NOT required to have ArcMap open while running ArcGIS Tools,
#   However, the scripts are designed for use with a GUI within the ArcGIS environment.
#   The scripts will need to be modified to to run outside of GIS and is not recommended.
# ---------------------------------------------------------------------------
print'Importing arcpy. This might take a while...'

# _____________________IMPORT STATEMENTS_____________________
import arcpy
from arcpy import env
import os

# mxd = arcpy.mapping.MapDocument("CURRENT")        # haven't found a use for this yet

# Start Message
arcpy.AddMessage(' Initiating Setup (this may take a while)...')

# _____________________FUNCTIONS_____________________
# No functions required for this script

# _____________________OPERABLE SCRIPT_____________________
# Allow code to overwrite results and set environment
env.overwriteOutput = True

# ------------- input -------------
# Get Path for GeoDatabase from GUI
out_folder_path = arcpy.GetParameterAsText(0)       # = System Folder Location
out_name = arcpy.GetParameterAsText(1) + '.gdb'     # = String + '.gdb'

# Get Spacial Reference for Feature DataSets
srtext = arcpy.GetParameterAsText(2)
sr = arcpy.SpatialReference()
env.outputCoordinateSystem = sr.loadFromString(srtext)
# ----------- end input -----------

# Execute CreateFileGDB
arcpy.AddMessage(' Creating GeoDatabase...')
arcpy.CreateFileGDB_management(out_folder_path, out_name)

# Execute Create Feature Datasets
arcpy.AddMessage(' Creating Feature Datasets...')
geoDB_path = out_folder_path + os.sep + out_name
# arcpy.AddMessage(geoDB_path)
arcpy.CreateFeatureDataset_management(out_dataset_path=geoDB_path, out_name='PivotFinder_Input',
spatial_reference=sr)
arcpy.CreateFeatureDataset_management(out_dataset_path=geoDB_path, out_name='PivotFinder_Output',
spatial_reference=sr)
arcpy.CreateFeatureDataset_management(out_dataset_path=geoDB_path, out_name='PivotFinder_Solution',
spatial_reference=sr)

# Execute Create Feature Classes
arcpy.AddMessage(' Creating Feature Classes...')
inputFDS_path = geoDB_path + os.sep + 'PivotFinder_Input'
FC = ['LineObstructions', 'PointObstructions', 'PolygonObstructions', 'SearchSpaces']
FC_type = ['POLYLINE', 'POINT', 'POLYGON', 'POLYGON']
for x in range(4):
    arcpy.CreateFeatureclass_management(out_path=inputFDS_path, out_name=FC[x],
geometry_type=FC_type[x])

# Done statement
arcpy.AddMessage('Setup Complete.')
```

## Appendix C: Full Circle Pivot Script

```
# -*- coding: utf-8 -*-
# ---------------------------------------------------------------------------
# Setup.py
# Created on: 2014-11-14
# Programmer: Justin Dietrich, Utah State University, Logan, UT, USA
# License: Use at own risk.
# Description: This script is intended for use with the Pivot_Finder ArcGIS Tool Package.
# Objective 1: Pull in Parameters.
# Objective 2: Create Full Circle Pivot Possibilities
# Objective 3: Consider the gross coverage area of each pivot.
# Objective 4: Choose the pivot with the maximum coverage area.
# Learning Notes: It is NOT required to have ArcMap open while running ArcGIS Tools,
#   However, the scripts are designed for use with a GUI within the ArcGIS environment.
#   The scripts will need to be modified to to run outside of GIS and is not recommended.
# ---------------------------------------------------------------------------
print'Importing arcpy. This might take a while...'

# _____________________IMPORT STATEMENTS_____________________
import arcpy
from arcpy import env
import os
import shlex

# Start Message
arcpy.AddMessage('Optimizing Pivot Location (this may take a while)...')

# _____________________FUNCTIONS_____________________
def DeleteIfExist(FileName):
    if arcpy.Exists(FileName):
        print '  Deleting Previous ' + FileName
        arcpy.Delete_management(FileName)

# _____________________OPERABLE SCRIPT_____________________
# Allow code to overwrite results and set environment
env.overwriteOutput = True

# ------------- input -------------
GDB_path = arcpy.GetParameterAsText(0)
env.workspace = GDB_path

Cell_Size_Width = arcpy.GetParameterAsText(1)
if Cell_Size_Width == '#' or not Cell_Size_Width:
    Cell_Size_Width = "10" # provide a default value if unspecified

Cell_Size_Height = arcpy.GetParameterAsText(2)
if Cell_Size_Height == '#' or not Cell_Size_Height:
    Cell_Size_Height = "10" # provide a default value if unspecified

SearchExtents = arcpy.GetParameterAsText(3)      # Extent to create search Grid, would be better to
avoid this input and have this read directly from the Search Space Feature Class Extent
if SearchExtents == '#' or not SearchExtents:
    SearchExtents = "1870309.4019413 55661.7690603137 1870555.37717114 55849.4278057253" # provide a
default value if unspecified
                        # Should change this default to something automatic from arcpy.mapping (like the
current extent or something)
SearchExtentsList = shlex.split(SearchExtents)
                        # This creates a list version of SearchExtents for easy call

SearchSpace = arcpy.GetParameterAsText(4)
if SearchSpace == '#' or not SearchSpace:
    SearchSpace =
"C:\\Users\\Justin\\Documents\\ArcGIS\\CEE_6440_HW\\TermProject\\Test1_Bluffdale.gdb\\PivotFinder_Input\
\SearchSpaces" # provide a default value if unspecified
                        # Default on this needs to be fixed to the actual searchSpace feature class in
the prepared .gdb

LineObstructions = arcpy.GetParameterAsText(5)
if LineObstructions == '#' or not LineObstructions:
```

```
    LineObstructions =
"C:\\Users\\Justin\\Documents\\ArcGIS\\CEE_6440_HW\\TermProject\\Test1_Bluffdale.gdb\\PivotFinder_Input\
\LineObstructions"
                        # Default on this needs to be fixed to the actual searchSpace feature class in
the prepared .gdb

PointObstructions = arcpy.GetParameterAsText(6)
if PointObstructions == '#' or not PointObstructions:
    PointObstructions =
"C:\\Users\\Justin\\Documents\\ArcGIS\\CEE_6440_HW\\TermProject\\Test1_Bluffdale.gdb\\PivotFinder_Input\
\PointObstructions"
                        # Default on this needs to be fixed to the actual searchSpace feature class in
the prepared .gdb

PolygonObstructions = arcpy.GetParameterAsText(7)
if PolygonObstructions == '#' or not PolygonObstructions:
    PolygonObstructions =
"C:\\Users\\Justin\\Documents\\ArcGIS\\CEE_6440_HW\\TermProject\\Test1_Bluffdale.gdb\\PivotFinder_Input\
\PolygonObstructions"
                        # Default on this needs to be fixed to the actual searchSpace feature class in
the prepared .gdb

srtext = arcpy.GetParameterAsText(8)
sr = arcpy.SpatialReference()
env.outputCoordinateSystem = sr.loadFromString(srtext)
# ----------- end input -----------

# Process: Create Fishnet
arcpy.AddMessage(' Creating Fishnet...')
DeleteIfExist('Fishnet')
DeleteIfExist('Fishnet_label')
arcpy.CreateFishnet_management(GDB_path + os.sep + 'PivotFinder_Output\\Fishnet',
str(SearchExtentsList[0]) + ' ' +
                               str(SearchExtentsList[1]),
                               str(SearchExtentsList[0]) + ' ' + str(SearchExtentsList[3]),
                               Cell_Size_Width, Cell_Size_Height, "", "",
                               str(SearchExtentsList[2]) + ' ' + str(SearchExtentsList[3]),
                               "LABELS", "", "POLYLINE")
                        # This does respond to the env.workspace and will place outputs like Fishnet and
Fishnet_Labels there
                        # SearchExtents overrides the origin and opposite corner inputs so we're ok to
leave the hard values in there.

# Process: Clip
arcpy.AddMessage(' Processing Clip...')
DeleteIfExist('Fishnet_clip')
arcpy.Clip_analysis('Fishnet_label', SearchSpace, GDB_path + os.sep +
'PivotFinder_Output\\Fishnet_clip', "")
                        # This does respond to the env.workspace and will place outputs like Fishnet and
Fishnet_Labels there

# Process: Add Field
arcpy.AddMessage(' Processing Add Field...')
arcpy.AddField_management('Fishnet_clip', "CNTR_NUM", "SHORT", "", "", "", "", "NULLABLE", "REQUIRED",
"")
                        # This just amends the Fishnet_clip table, doesn't create a new one.

# Process: Calculate Field
arcpy.AddMessage(' Processing Calculate Field...')
arcpy.CalculateField_management('Fishnet_clip', "CNTR_NUM", "!OBJECTID!", "PYTHON_9.3", "")
                        # This just amends the Fishnet_clip table, doesn't create a new one.

# Process: Near
arcpy.AddMessage(' Processing Near...')
arcpy.Near_analysis('Fishnet_clip', (PointObstructions, LineObstructions, PolygonObstructions), "",
                    "LOCATION", "ANGLE", "PLANAR")
                        # This just amends the Fishnet_clip table, doesn't create a new one.

# Process: Buffer
arcpy.AddMessage(' Processing Buffer...')
DeleteIfExist('FullCircles')
arcpy.Buffer_analysis('Fishnet_clip', GDB_path + os.sep + 'PivotFinder_Output\\FullCircles',
```

```
                      "NEAR_DIST", "FULL", "ROUND", "NONE", "")
                        # This does respond to the env.workspace for outputs/inputs and spatial
reference

# Process: Make Feature Layer
arcpy.AddMessage(' Processing Make Feature Layer...')
arcpy.MakeFeatureLayer_management('FullCircles', 'SolutionCoverageTemp',
                                  "Shape_Area = (SELECT MAX(Shape_Area) FROM FullCircles)",
                                  "",
                                  "OBJECTID OBJECTID VISIBLE NONE;Shape Shape VISIBLE NONE;"
                                  "NEAR_FID NEAR_FID VISIBLE NONE;NEAR_DIST NEAR_DIST VISIBLE NONE;"
                                  "NEAR_X NEAR_X VISIBLE NONE;NEAR_Y NEAR_Y VISIBLE NONE;"
                                  "NEAR_ANGLE NEAR_ANGLE VISIBLE NONE;NEAR_FC NEAR_FC VISIBLE NONE;"
                                  "BUFF_DIST BUFF_DIST VISIBLE NONE;ORIG_FID ORIG_FID VISIBLE NONE")
                        # This creates a temporary feature layer (I think only in memory), so no need to
DeleteIfExists

# Process: Copy Features
arcpy.AddMessage(' Processing Copy Features...')
DeleteIfExist('SolutionCoverage')
arcpy.CopyFeatures_management('SolutionCoverageTemp', GDB_path + os.sep +
'PivotFinder_Solution\\SolutionCoverage',
                              "", "0", "0", "0")

# Process: Make Feature Layer (2)
arcpy.AddMessage(' Processing Make Feature Layer 2...')
arcpy.MakeFeatureLayer_management('Fishnet_clip', 'Fishnet_clip_layer',
                                  "",
                                  "",
                                  "OBJECTID OBJECTID VISIBLE NONE;"
                                  "Shape Shape VISIBLE NONE;CNTR_NUM CNTR_NUM VISIBLE NONE;"
                                  "NEAR_FID NEAR_FID VISIBLE NONE;NEAR_DIST NEAR_DIST VISIBLE NONE;"
                                  "NEAR_X NEAR_X VISIBLE NONE;NEAR_Y NEAR_Y VISIBLE NONE;"
                                  "NEAR_ANGLE NEAR_ANGLE VISIBLE NONE;NEAR_FC NEAR_FC VISIBLE NONE")
                        # This creates a temporary feature layer (I think only in memory), so no need to
DeleteIfExists

# Process: Add Join
arcpy.AddMessage(' Processing Add Join...')
arcpy.AddJoin_management('Fishnet_clip_layer', "CNTR_NUM", 'SolutionCoverageTemp', "CNTR_NUM",
"KEEP_ALL")
                        # This just temporarily joins the two tables; it doesn't create a new one.

# Process: Copy Features (2)
arcpy.AddMessage(' Processing Copy Features 2...')
arcpy.CopyFeatures_management('Fishnet_clip_layer', GDB_path + os.sep +
'PivotFinder_Solution\\SolutionCenter',
                              "", "0", "0", "0")

# Done statement
arcpy.AddMessage(' Optimization Complete.')
```

## Appendix D: Land Quality Pivot Script

```
# -*- coding: utf-8 -*-
# ---------------------------------------------------------------------------
# LandQuality.py
# Created on: 2014-12-04
# Programmer: Justin Dietrich, Utah State University, Logan, UT, USA
# License: Use at own risk.
# Description: This script is intended for use with the Pivot_Finder ArcGIS Tool Package.
# Objective 1: Pull in Parameters.
# Objective 2: Create Full Circle Pivot Possibilities
# Objective 3: Consider the Land Value Associated with each pivot.
# Objective 4: Choose the pivot with the maximum production value.
# Learning Notes: It is NOT required to have ArcMap open while running ArcGIS Tools,
#   However, the scripts are designed for use with a GUI within the ArcGIS environment.
#   The scripts will need to be modified to to run outside of GIS and is not recommended.
# ---------------------------------------------------------------------------
print'Importing arcpy. This might take a while...'

# _____________________IMPORT STATEMENTS_____________________
import arcpy
from arcpy import env
import os
import shlex

# Start Message
arcpy.AddMessage('Optimizing Pivot Location (this may take a while)...')

# _____________________FUNCTIONS_____________________
def DeleteIfExist(FileName):
    if arcpy.Exists(FileName):
        print '  Deleting Previous ' + FileName
        arcpy.Delete_management(FileName)

# _____________________OPERABLE SCRIPT_____________________
# Allow code to overwrite results and set environment
env.overwriteOutput = True

# ------------- input -------------
GDB_path = arcpy.GetParameterAsText(0)
env.workspace = GDB_path

Cell_Size_Width = arcpy.GetParameterAsText(1)
if Cell_Size_Width == '#' or not Cell_Size_Width:
    Cell_Size_Width = "10" # provide a default value if unspecified

Cell_Size_Height = arcpy.GetParameterAsText(2)
if Cell_Size_Height == '#' or not Cell_Size_Height:
    Cell_Size_Height = "10" # provide a default value if unspecified

SearchExtents = arcpy.GetParameterAsText(3)      # Extent to create search Grid, would be better to
avoid this input and have this read directly from the Search Space Feature Class Extent
if SearchExtents == '#' or not SearchExtents:
    SearchExtents = "1870309.4019413 55661.7690603137 1870555.37717114 55849.4278057253" # provide a
default value if unspecified
                        # Should change this default to something automatic from arcpy.mapping (like the
current extent or something)
SearchExtentsList = shlex.split(SearchExtents)
                        # This creates a list version of SearchExtents for easy call

SearchSpace = arcpy.GetParameterAsText(4)
if SearchSpace == '#' or not SearchSpace:
    SearchSpace =
"C:\\Users\\Justin\\Documents\\ArcGIS\\CEE_6440_HW\\TermProject\\Test1_Bluffdale.gdb\\PivotFinder_Input\
\SearchSpaces" # provide a default value if unspecified
                        # Default on this needs to be fixed to the actual searchSpace feature class in
the prepared .gdb

LineObstructions = arcpy.GetParameterAsText(5)
if LineObstructions == '#' or not LineObstructions:
```

```
    LineObstructions =
"C:\\Users\\Justin\\Documents\\ArcGIS\\CEE_6440_HW\\TermProject\\Test1_Bluffdale.gdb\\PivotFinder_Input\
\LineObstructions"
                        # Default on this needs to be fixed to the actual searchSpace feature class in
the prepared .gdb

PointObstructions = arcpy.GetParameterAsText(6)
if PointObstructions == '#' or not PointObstructions:
    PointObstructions =
"C:\\Users\\Justin\\Documents\\ArcGIS\\CEE_6440_HW\\TermProject\\Test1_Bluffdale.gdb\\PivotFinder_Input\
\PointObstructions"
                        # Default on this needs to be fixed to the actual searchSpace feature class in
the prepared .gdb

PolygonObstructions = arcpy.GetParameterAsText(7)
if PolygonObstructions == '#' or not PolygonObstructions:
    PolygonObstructions =
"C:\\Users\\Justin\\Documents\\ArcGIS\\CEE_6440_HW\\TermProject\\Test1_Bluffdale.gdb\\PivotFinder_Input\
\PolygonObstructions"
                        # Default on this needs to be fixed to the actual searchSpace feature class in
the prepared .gdb

srtext = arcpy.GetParameterAsText(8)
sr = arcpy.SpatialReference()
env.outputCoordinateSystem = sr.loadFromString(srtext)

LandQualityPoints = arcpy.GetParameterAsText(9)
if LandQualityPoints == '#' or not LandQualityPoints:
    LandQualityPoints =
"C:\\Users\\Justin\\Documents\\ArcGIS\\CEE_6440_HW\\TermProject\\Test2b_LaSal.gdb\\PivotFinder_Input\\La
ndQualityPoints" # provide a default value if unspecified
                        # Default on this needs to be fixed to the actual searchSpace feature class in
the prepared .gdb
# ----------- end input -----------


# Process: Create Fishnet
arcpy.AddMessage(' Creating Fishnet...')
DeleteIfExist('Fishnet')
DeleteIfExist('Fishnet_label')
arcpy.CreateFishnet_management(GDB_path + os.sep + 'PivotFinder_Output\\Fishnet',
str(SearchExtentsList[0]) + ' ' +
                               str(SearchExtentsList[1]),
                               str(SearchExtentsList[0]) + ' ' + str(SearchExtentsList[3]),
                               Cell_Size_Width, Cell_Size_Height, "", "",
                               str(SearchExtentsList[2]) + ' ' + str(SearchExtentsList[3]),
                               "LABELS", "", "POLYLINE")
                        # This does respond to the env.workspace and will place outputs like Fishnet and
Fishnet_Labels there
                        # SearchExtents overrides the origin and opposite corner inputs so we're ok to
leave the hard values in there.

# Process: Clip
arcpy.AddMessage(' Processing Clip...')
DeleteIfExist('Fishnet_clip')
arcpy.Clip_analysis('Fishnet_label', SearchSpace, GDB_path + os.sep +
'PivotFinder_Output\\Fishnet_clip', "")
                        # This does respond to the env.workspace and will place outputs like Fishnet and
Fishnet_Labels there

# Process: Add Field
arcpy.AddMessage(' Processing Add Field...')
arcpy.AddField_management('Fishnet_clip', "CNTR_NUM", "SHORT", "", "", "", "", "NULLABLE", "REQUIRED",
"")
                        # This just amends the Fishnet_clip table, doesn't create a new one.

# Process: Calculate Field
arcpy.AddMessage(' Processing Calculate Field...')
arcpy.CalculateField_management('Fishnet_clip', "CNTR_NUM", "!OBJECTID!", "PYTHON_9.3", "")
                        # This just amends the Fishnet_clip table, doesn't create a new one.

# Process: Near
```

```
arcpy.AddMessage(' Processing Near...')
arcpy.Near_analysis('Fishnet_clip', (PointObstructions, LineObstructions, PolygonObstructions), "",
                    "LOCATION", "ANGLE", "PLANAR")
                        # This just amends the Fishnet_clip table, doesn't create a new one.

# Process: Buffer
arcpy.AddMessage(' Processing Buffer...')
DeleteIfExist('FullCircles')
arcpy.Buffer_analysis('Fishnet_clip', GDB_path + os.sep + 'PivotFinder_Output\\FullCircles',
                      "NEAR_DIST", "FULL", "ROUND", "NONE", "")
                        # This does respond to the env.workspace for outputs/inputs and spatial
reference

# Process: Create Thiessen Polygons
arcpy.AddMessage(' Creating Thiessen Polygons...')
arcpy.CreateThiessenPolygons_analysis(LandQualityPoints, GDB_path + os.sep +
'PivotFinder_Output\\Thiessen_Polygons',
                                      "ALL")

# Process: Dissolve
arcpy.AddMessage(' Processing Dissolve...')
arcpy.Dissolve_management('Thiessen_Polygons', GDB_path + os.sep +
'PivotFinder_Output\\ThiessenDissolve', 'LandValue',
                          "", "MULTI_PART", "DISSOLVE_LINES")

# Process: Tabulate Intersection
arcpy.AddMessage(' Processing Tabulate Intersection... \n'
                 '    Note: if the tool stalls on this step, you have likely reached a stack overflow
condition. \n'
                 '    End the task and repeat with either a smaller search area or coarser Search
Density. \n'
                 '    If the working search density is too coarse for decision making, repeat the
process with a \n'
                 '    higher density and with a smaller search area centered around the coarse
solution.')
arcpy.TabulateIntersection_analysis(in_zone_features='FullCircles', zone_fields="CNTR_NUM",
                                    in_class_features='ThiessenDissolve', out_table='ValueTable',
                                    class_fields='LandValue', sum_fields="", xy_tolerance="",
out_units="UNKNOWN")

# Process: Add Field (2)
arcpy.AddMessage(' Processing Add Field (2)...')
arcpy.AddField_management(in_table='ValueTable', field_name="Area_Value", field_type="DOUBLE",
                          field_is_nullable="NULLABLE", field_is_required="NON_REQUIRED")

# Process: Calculate Field (2)
arcpy.AddMessage(' Processing Calculate Field (2)...')
arcpy.CalculateField_management(in_table='ValueTable', field="Area_Value", expression="[LandValue] *
[AREA]",
                                expression_type="VB")

# Process: Summary Statistics (acts like dissolve, but for a table)
arcpy.AddMessage(' Processing Summary Statistics...')
arcpy.Statistics_analysis(in_table='ValueTable', out_table='ValuesDissolved',
statistics_fields="Area_Value SUM",
                          case_field="CNTR_NUM")

# Process: Table Select
arcpy.AddMessage(' Processing Table Select...')
arcpy.TableSelect_analysis(in_table='ValuesDissolved', out_table='ValuesBest',
                           where_clause="SUM_Area_Value = (SELECT MAX(SUM_Area_Value) FROM
ValuesDissolved)")

# Process: Use cursor to obtain the center number representing the optimal pivot
arcpy.AddMessage(' Getting Optimal Pivot Center Number...')
table = 'ValuesBest'            # Does this know that we want the values best table, not just a string
called that?
rows = arcpy.SearchCursor(table)
for row in rows:
    optimalCNTR_NUM = row.getValue('CNTR_NUM')
arcpy.AddMessage('  Optimal Center Number is ' + str(optimalCNTR_NUM))
```

```
# Process: Make Feature Layer
arcpy.AddMessage(' Processing Make Feature Layer...')
arcpy.MakeFeatureLayer_management(in_features='FullCircles', out_layer='SolutionCoverageTemp',
where_clause=
                                  "CNTR_NUM = " + str(optimalCNTR_NUM),
                                  workspace="", field_info=
                                  "OBJECTID OBJECTID VISIBLE NONE;Shape Shape VISIBLE NONE;"
                                  "NEAR_FID NEAR_FID VISIBLE NONE;NEAR_DIST NEAR_DIST VISIBLE NONE;"
                                  "NEAR_X NEAR_X VISIBLE NONE;NEAR_Y NEAR_Y VISIBLE NONE;"
                                  "NEAR_ANGLE NEAR_ANGLE VISIBLE NONE;NEAR_FC NEAR_FC VISIBLE NONE;"
                                  "BUFF_DIST BUFF_DIST VISIBLE NONE;ORIG_FID ORIG_FID VISIBLE NONE")
                         # This creates a temporary feature layer (I think only in memory), so no need to
DeleteIfExists

# Process: Copy Features
arcpy.AddMessage(' Processing Copy Features...')
DeleteIfExist('SolutionCoverage')
arcpy.CopyFeatures_management('SolutionCoverageTemp', GDB_path + os.sep +
'PivotFinder_Solution\\SolutionCoverage',
                               "", "0", "0", "0")

# Process: Make Feature Layer (2)
arcpy.AddMessage(' Processing Make Feature Layer 2...')
arcpy.MakeFeatureLayer_management('Fishnet_clip', 'Fishnet_clip_layer',
                                  "",
                                  "",
                                  "OBJECTID OBJECTID VISIBLE NONE;"
                                  "Shape Shape VISIBLE NONE;CNTR_NUM CNTR_NUM VISIBLE NONE;"
                                  "NEAR_FID NEAR_FID VISIBLE NONE;NEAR_DIST NEAR_DIST VISIBLE NONE;"
                                  "NEAR_X NEAR_X VISIBLE NONE;NEAR_Y NEAR_Y VISIBLE NONE;"
                                  "NEAR_ANGLE NEAR_ANGLE VISIBLE NONE;NEAR_FC NEAR_FC VISIBLE NONE")
                         # This creates a temporary feature layer (I think only in memory), so no need to
DeleteIfExists

# Process: Add Join
arcpy.AddMessage(' Processing Add Join...')
arcpy.AddJoin_management('Fishnet_clip_layer', "CNTR_NUM", 'SolutionCoverageTemp', "CNTR_NUM",
"KEEP_ALL")
                         # This just temporarily joins the two tables; it doesn't create a new one.

# Process: Copy Features (2)
arcpy.AddMessage(' Processing Copy Features 2...')
arcpy.CopyFeatures_management('Fishnet_clip_layer', GDB_path + os.sep +
'PivotFinder_Solution\\SolutionCenter',
                               "", "0", "0", "0")

# Done statement
arcpy.AddMessage(' Optimization Complete.')
```