



Michigan Technological University
Create the Future Digital Commons @ Michigan Tech

Dissertations, Master's Theses and Master's
Reports - Open

Dissertations, Master's Theses and Master's
Reports

2015

DEVELOPMENT OF A FUNDAMENTAL RATING SYSTEM FOR IDENTIFYING SPRAWL: A CASE STUDY UTILIZING SMALL CITIES IN MICHIGAN

Jonathan T. Riehl
Michigan Technological University

Follow this and additional works at: <https://digitalcommons.mtu.edu/etds>



Part of the [Civil Engineering Commons](#), and the [Urban Studies and Planning Commons](#)

Copyright 2015 Jonathan T. Riehl

Recommended Citation

Riehl, Jonathan T., "DEVELOPMENT OF A FUNDAMENTAL RATING SYSTEM FOR IDENTIFYING SPRAWL: A CASE STUDY UTILIZING SMALL CITIES IN MICHIGAN", Dissertation, Michigan Technological University, 2015.

<https://doi.org/10.37099/mtu.dc.etds/1012>

Follow this and additional works at: <https://digitalcommons.mtu.edu/etds>



Part of the [Civil Engineering Commons](#), and the [Urban Studies and Planning Commons](#)

DEVELOPMENT OF A FUNDAMENTAL RATING SYSTEM
FOR IDENTIFYING SPRAWL:
A CASE STUDY UTILIZING SMALL CITIES IN MICHIGAN

By

Jonathan T. Riehl

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Civil Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2015

©2015 Jonathan T. Riehl

This dissertation has been approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY in Civil Engineering.

Department of Civil and Environmental Engineering

Dissertation Advisor: *Dr. William Sproule, Civil Engineering*

Committee Member: *Dr. Bernard Alkire, Civil Engineering*

Committee Member: *Dr. Thomas Merz, Business and Economics*

Committee Member: *Dr. Amlan Mukherjee, Civil Engineering*

Department Chair: *Dr. David Hand, Civil Engineering*

To all who are willing to think beyond
short-term economic incentives

Table of Contents

List of Figures.....	ix
List of Tables.....	xi
List of Acronyms.....	xii
Acknowledgements	xiv
Abstract.....	xvii
Chapter 1: Introduction.....	1
1.1 Background.....	1
1.2 Research Focus	1
1.3 Research Need	2
1.4 Research Questions.....	4
1.5 Dissertation Outline	4
Chapter 2: Background on Urban Sprawl	5
2.1 Competing Definitions of Sprawl	5
2.2 Causes of Sprawl.....	6
2.3 Components of Sprawl.....	7
2.4 Consequences of Sprawl	7
2.5 A Working Definition of Sprawl	11
Chapter 3: Literature Review of Sprawl Metrics	12
3.1 Inability to Consistently Rank Sprawl for Large Cities.....	12
3.2 Variety of Metrics Used.....	14
3.3 Using Remote Sensing to Define Urban Areas and Measure Sprawl.....	18
3.4 Other Considerations	19
3.4.1 Walkability / Livability	19
3.4.2 Environmental Impacts	20
3.4.3 Basic Psychology	21
3.4.4 Empirical Studies	22
3.5 Summary of Sprawl Metrics and Metric Selection.....	24

Chapter 4: Study Methodology and Tools.....	25
4.1 Pilot City Selection Methodology.....	25
4.2 City Scope Selection	28
4.2.1 Range of Geographical Areas Defining a Cities Scope	29
4.2.2 Analysis of Potential Geographies	31
4.2.3 Selection of the Appropriate Geography.....	34
4.3 Metric and Sprawl Score Calculation	35
Chapter 5: Extended Urban Clusters	37
5.1 Development of the EUC Algorithm	37
5.1.1 Extended Urban Areas	37
5.1.1 Extending the EUA Algorithm to Extended Urban Clusters	38
5.2 Python Script – EUC_Creator.py.....	39
5.2.1 Files needed for EUC_Creator.py	41
5.2.2 Overview of EUC_Creator.py.....	41
5.2.2.1 Selection of UA area and Overlapping Block Groups	42
5.2.2.2 Selection of Outlying Blocks.....	44
5.3 Final EUC Boundaries	48
Chapter 6: Sprawl Metric Assessment	51
6.1 Selected Metrics.....	51
6.1.1 Development Density Metrics.....	52
6.1.1.1 Population density	53
6.1.1.2 Employment density	53
6.1.1.3 Percentage of the population living at suburban densities	54
6.1.1.4 Percentage of the population living at urban densities	54
6.1.1.5 Net amount of urban lands	55
6.1.1.6 Estimated density at the center of the metro area.....	55
6.1.1.7 Population Density of CBD.....	57
6.1.1.8 Employment Density of CBD	58
6.1.2 Land Use Mix Metrics	58
6.1.2.1 Job / population balance	59

6.1.2.2 Degree of Job Mixing.....	60
6.1.2.3 Walk Score	61
6.1.3 Activity Centering Metrics.....	61
6.1.3.1 Coefficient of variation in block population densities	62
6.1.3.2 Coefficient of variation in block employment densities.....	62
6.1.3.3 Density gradient moving outward from the CBD	63
6.1.3.4 Percentage of population in the CBD	64
6.1.3.5 Percentage of employment in the CBD	64
6.1.4 Street Accessibility Metrics	65
6.1.4.1 Percentage of small urban blocks	65
6.1.4.2 Average block size	66
6.1.4.3 Average block length.....	66
6.1.4.4 Intersection Density.....	67
6.1.4.5 Percentage of 4-or-more-way intersection	68
6.2 Python Script: EUC_Metrics.py	68
6.2.1 Files needed for EUC_Metrics.py.....	69
6.2.2 Overview of EUC_Metrics.py	70
6.2.3 General Notes for Improving Algorithm.....	71
6.3 Support Scripts.....	71
6.3.1 Preparation Work for EUC and Block Shapefiles.....	71
6.3.1.1 Employment Data.....	72
6.3.1.2 Land Use / Land Cover Data	72
6.3.1.3 Central Business District Point File	73
6.3.1.4 Intersection Point File.....	73
6.3.2 Python Script: CBD_Creator.py.....	74
6.3.3 Python Script: WalkScore_Locator.py.....	75
Chapter 7: Sprawl Score.....	77
7.1 Principal Components Analysis of Metrics	77
7.2 Python Script: EUC_Sprawl_Score.py	78
7.2.1 Data preparation for script.....	78
7.2.2 Principal Components Analysis Algorithm	80

7.2.3 Sprawl Score Algorithm.....	82
7.3 Rank of Pilot Cities with Sprawl Score	82
7.4 Analysis of Rankings	87
7.4.1 Overall Winners and Losers Based on Factor Categories.....	87
7.4.2 How Communities Should Use the Sprawl Scorecard.....	90
 Chapter 8: Conclusions and Recommendations	92
8.1 Conclusions.....	92
8.1.1 Research Question Summary	92
8.1.2 Other Conclusions.....	93
8.2 How the Sprawl Score can be used.....	93
8.3 Recommendations for Future Research	94
8.3.1 Perform In-Depth Analysis of Sprawl Score for Pilot Cities.....	95
8.3.2 Extend the Scope of the Sprawl Score	95
8.3.3 Run Sensitivity Testing to Improve Geographical Area Selection	95
8.3.4 Reduce the Number of Metrics	96
8.3.5 Improve Tools and Extend Tools to Other Areas	96
8.3.6 Address Other Research Realms of Sprawl	97
8.3.7 Perform Historical Comparisons with Sprawl Score	97
8.3.8 Make the Sprawl Score more Accessible.....	97
8.3.9 Analyze Inter-Urban Influence	98
 References	99
 Appendix A: Copyright Permission Form for Figure 3.1.....	107
 Appendix B: EUC_Creator.py – Overview Full Python Script	108
B.1 Overview for Programmers.....	108
B.1.1 Reasoning and Methodology of Loops in Algorithm.....	109
B.1.2 Donut Addition Algorithm	110
B.1.3 RUCA Algorithm	110
B.1.4 Issues and Fixes.....	111

B.2 EUC_Creator.py – Full Python Script.....	113
Appendix C: EUC_Metrics.py – Full Python Script.....	121
Appendix D: CBD_Creator.py – Full Python Script.....	132
Appendix E: WalkScore_Locator.py – Full Python Script	137
Appendix F: EUC_Sprawl_Score.py – Full Python Script	140
Appendix G: Sprawl Scorecards for All 42 Pilot Cities.....	148
Appendix H: Support Files on Supplemental DVD	191

List of Figures

Figure 2.1: Sustainability triangle	9
Figure 3.1: Changing land use patterns of Houghton, MI over past forty years [84]	20
Figure 4.1: Map of Michigan pilot cities by population and region.....	28
Figure 4.2: Standard hierarchy of census geographic entities	30
Figure 4.3: Geographic units and their various ranges for Grand Rapids, Michigan.....	32
Figure 4.4: Graphic example of preliminary Houghton EUC	35
Figure 5.1: Basic flowchart for EUA shapefile Creation	38
Figure 5.2: Basic flowchart for EUC shapefile creation	40
Figure 5.3: Flowchart for UAC overlapping block group / block removal.....	43
Figure 5.4: Flowchart for EUC overlapping block group / block removal	44
Figure 5.5: Flowchart for removal of blocks with under 30% commuters to UAC	47
Figure 5.6: Map of 116 EUCs in Michigan (created by EUC_Creator.py).....	48
Figure 5.7: Map of 42 pilot EUCs in Michigan.....	49
Figure 5.8: Graphic example of Houghton EUC, including final Houghton EUC	50
Figure 6.1: Basic flowchart for EUC metric calculation.....	70
Figure 6.2: Basic flowchart for CBD polygon creation	75
Figure 6.3: Basic flowchart for Walk Score locator program	76
Figure 7.1: Basic flowchart for Sprawl Score calculator program	79
Figure 7.2: Example Sprawl Scorecard for Houghton, Michigan	90
Figure G.1: Adrian Sprawl Scorecard	149
Figure G.2: Alma Sprawl Scorecard	150
Figure G.3: Alpena Sprawl Scorecard.....	151
Figure G.4: Bad Axe Sprawl Scorecard	152
Figure G.5: Big Rapids Sprawl Scorecard	153
Figure G.6: Cadillac Sprawl Scorecard	154
Figure G.7: Caro Sprawl Scorecard.....	155
Figure G.8: Charlevoix Sprawl Scorecard.....	156
Figure G.9: Cheboygan Sprawl Scorecard	157
Figure G.10: Coldwater Sprawl Scorecard.....	158
Figure G.11: East Tawas Sprawl Scorecard	159

Figure G.12: Escanaba Sprawl Scorecard	160
Figure G.13: Fremont Sprawl Scorecard	161
Figure G.14: Gaylord Sprawl Scorecard	162
Figure G.15: Gladwin Sprawl Scorecard.....	163
Figure G.16: Grayling Sprawl Scorecard	164
Figure G.17: Greenville Sprawl Scorecard	165
Figure G.18: Harrison Sprawl Scorecard	166
Figure G.19: Hart Sprawl Scorecard	167
Figure G.20: Hastings Sprawl Scorecard	168
Figure G.21: Hillsdale Sprawl Scorecard.....	169
Figure G.22: Houghton Sprawl Scorecard	170
Figure G.23: Houghton Lake Sprawl Scorecard	171
Figure G.24: Ionia Sprawl Scorecard	172
Figure G.25: Iron River Sprawl Scorecard.....	173
Figure G.26: Kalkaska Sprawl Scorecard	174
Figure G.27: Lapeer Sprawl Scorecard	175
Figure G.28: Ludington Sprawl Scorecard.....	176
Figure G.29: Manistee Sprawl Scorecard.....	177
Figure G.30: Manistique Sprawl Scorecard	178
Figure G.31: Marquette Sprawl Scorecard.....	179
Figure G.32: Mount Pleasant Sprawl Scorecard	180
Figure G.33: Munising Sprawl Scorecard	181
Figure G.34: Newberry Sprawl Scorecard	182
Figure G.35: Owosso Sprawl Scorecard	183
Figure G.36: Petoskey Sprawl Scorecard.....	184
Figure G.37: Rogers City Sprawl Scorecard	185
Figure G.38: Saint Ignace Sprawl Scorecard	186
Figure G.39: Sandusky Sprawl Scorecard.....	187
Figure G.40: Sault Sainte Marie Sprawl Scorecard.....	188
Figure G.41: Sturgis Sprawl Scorecard	189
Figure G.42: Traverse City Sprawl Scorecard	190

List of Tables

Table 3.1: Comparison of city rankings among four foundational studies	13
Table 3.2: Metrics used to measure sprawl in previous research	15
Table 3.3: Definitions of metrics used to measure sprawl in previous research	15
Table 3.4: 22 Metrics across four dimensions used by Ewing et al. [3].....	16
Table 3.5: 12 Metrics across 8 dimensions used by Galster et al. [8]	17
Table 3.6: 38 Metrics across six dimensions hypothesized by author	23
Table 3.7: 21 Metrics across four dimensions used by Ewing et al. (2014) [3]	24
Table 4.1: List of 42 pilot cities in Michigan for this study	27
Table 4.2: Geographic areas in consideration for this study	33
Table 5.1: Files needed to run EUC_Creator.py Python script	41
Table 5.2: Comparison of Wolman et al.'s EUA and Riehl's EUC algorithms	42
Table 5.3: Table of RUCA codes [98].....	45
Table 6.1: Table of 21 metrics used in Sprawl Score calculation	52
Table 6.2: Files needed to run EUC_Metrics.py Python script.....	69
Table 7.1: Metric Weights as Determined by Principal Components Analysis	81
Table 7.2: Metric Weights as Determined by Principal Components Analysis	81
Table 7.3: Sprawl Scores (including all metrics) for mid-size regional cities	83
Table 7.4: Sprawl Scores (including all metrics) for small regional cities	84
Table 7.5: Sprawl Scores (including all metrics) for large subregional cities.....	85
Table 7.6: Sprawl Scores (including all metrics) for mid-size subregional cities.....	86
Table 7.7: Best and worst EUCs based on factor categories and Sprawl Scores	87

List of Acronyms

AARP – American Association of Retired Persons (formerly)

ACS – American Community Survey

AHS – American Housing Survey

BEA – Bureau of Economic Analysis

CBD – Central Business District

CBSA – Core-Based Statistical Area

CCD – Census County Subdivision

CEA – Component Economic Area

CFCC – Census Feature Class Code

CSA – Combined Statistical Area

CSV – Comma-Separated Value

CTTP – Census Transportation Planning Package

DVD – Digital Versatile Disc

EA – Economic Area

ERS – Economic Research Service

ESRI – Environmental Systems Research Institute (formerly)

EUA – Extended Urban Area

EUC – Extended Urban Cluster

FID – Field Identifier

FIPS – Federal Information Processing Standard

GIS – Geographical Information System

IQ – Intelligence Quotient

LEHD – Longitudinal Employer Household Dynamics

LODES7 – LEHD Origin-Destination Employment Statistics version 7

MRC – Metropolitan Research Center

MRLC – Multi-Resolution Land Characteristics

MSA – Metropolitan Statistical Area

NLCD – National Land Cover Database

OMB – Office of Management and Budget

PCA – Principal Components Analysis

RUCA – Rural-Urban Commuting Area

RUCC – Rural-Urban Continuum Codes

SQL – Structured Query Language

TRB – Transportation Research Board

TCRP – Transit Cooperative Research Program

UA – Urban Area

UAC – Urban Areas and Clusters

UC – Urban Cluster

UP – Upper Peninsula

USDA – United States Department of Agriculture

UZA – Urbanized Area

WAC – Workplace Area Characteristics

μSA – Micropolitan Statistical Area

Acknowledgements

My educational background is quite varied with degrees in electrical/computer engineering, civil engineering, geography, and business. In completing this work it became apparent just how much each and every element of this education has fit into the work in one way or another as listed below.

- Electrical/computer engineering has facilitated a systems approach to this work and allowed for the use of complex computer programming to drive the research
- Civil engineering has enabled a transportation-centered approach to the work and an understanding of the close ties between the transportation network and urban development
- Geography (at a school with a strong planning department conjoined with geography) has provided an appropriate background to understand the characteristics of urban development patterns as well as the policies surrounding this development.
- Business has grounded me with an understanding of the economics and incentives involved with urban development which helps in understanding the motivations for different development schemes

In addition, a love for travel has fueled the motivation that drives this research. Over the past 10 years, my wife and I have traveled the US and Canada extensively by vehicle. In driving through countless downtowns, suburbs, exurbs, and rural areas, I became very interested in urban sprawl.

One large gap in the literature was the lack of study of sprawl for small cities. Although most people would say that this was because sprawl does not exist in small cities, this did not fit with my active travel research. I firmly believe that studying sprawl at a small scale is the only way to get a handle on sprawl at a larger scale. This belief is the foundation for this work.

Although I do not like sprawl-style development for many reasons, many people prefer it to urban development. In many ways I have benefited from sprawl, but now that I am open to thinking about it, I have discovered that there are many ways I have been a victim of it as well.

However, there is a fundamental underpinning to this idea that goes beyond opinions. Sprawl development is not sustainable. Although any growth is not sustainable, traditional development and upkeep thereof offers a chance to be sustainable. And in a world that is spiraling out of control with growth and unsustainable development, I feel compelled to develop a mechanism to help solve the problem versus perpetuating it.

I would like to thank the following people for their contribution to my life and this work.

Committee Members and Former Committee Members

- Bill Sproule: For narrowing my focus to make the workload manageable, supporting me throughout the process, and pushing me to finish this work
- Bernie Alkire: For suggesting the idea of a PhD in 2010, suggesting Sproule as an advisor, and volunteering to join the committee late in the process
- Tom Merz: For grounding me in the economics of sprawl, his experience in community planning, and his rigorous and critical editing of the dissertation
- Amlan Mukherjee: For teaching me a systems approach to modeling real-world situations and for long discussions of this work at the Library
- Louise Dyble: For supporting my views on sprawl and providing many references on the social aspects of sprawl
- Jeff Lidicker: For suggesting statistical analysis methods for this work

Family and Friends

- Jennifer Riehl: For always being there for me and dealing with the stresses that accrued while working full time and completing a PhD; all while she was completing her PhD as well
- Barbara Riehl: For always encouraging me and never letting me think any achievement is impossible
- Richard Riehl: For sparking my interest in urban planning (although he did not know this at the time) and traveling
- David Riehl: For showing me the realities in big-box retail store management and talking to me about topics such as hockey and video games to keep me sane
- Bob Riehl: For strongly suggesting engineering over computer science and helping me attain excellent work experience while in undergrad
- Andy Riehl: For fighting developers and keeping Union Township, NJ a beautiful rural community

- Luke Gublo: For taking interest in my work and challenging me to learn more about the causes and components of urban sprawl
- The rest of my extended family and friends: For continued support and encouragement as well as dealing with my crazy travel habits that reduce times to visit with them

Other Academic Colleagues

- Colleagues in the Engineering Fundamentals Department: For supporting me in the pursuit of my PhD and allowing me the flexibility to finish
- John McDonald: For supporting me in my first graduate school endeavor and believing in my potential to be a graduate student
- Michael Wozny: For showing me that many paths in life are possible
- Todd Fabozzi: For introducing me to the concept of urban sprawl and for being so passionate about the issue
- Floyd Henderson: For encouraging and enabling my pursuit of a Masters in Geography
- Catherine Lawson: For advancing my knowledge of urban planning and the history of urban sprawl
- UN1001 Students: For helping me think about sprawl from many angles for a semester and providing me with many ideas that made it into this work
- Michigan Technological University: For essentially funding and supporting this work through their Employee Education Program

And Finally

- The Intrepid: For taking me to all of the places that made me care about urban sprawl and its impacts and for being a beast of a vehicle for so many years

Abstract

Urban sprawl research generally fits into one or more of four realms including definitions, causes, components, and consequences. Although research on consequences continues to thrive, research on components is in its adolescence, primarily due a lack of consensus on definition. Recent studies such as Ewing et al. 2014 have narrowed the list of sprawl metrics to about 20 within four factors including development density, land use mix, activity centering, and street accessibility.

This main product of this research is a Sprawl Scorecard for small Michigan cities varying in size from Traverse City, nearly 50,000 people in the urban cluster, down to Saint Ignace, with only 2,500. 42 small cities are included in the study, with an even spread of cities across the state.

One of the limitations with sprawl research is the focus on large cities. There is good reason to study large cities. Large cities affect more people, have more economic influence, and suffer recognizable consequences of sprawl (e.g., traffic congestion). However, large cities have more confounding variables at play than small cities making it difficult to narrow down the components. Even assuming components could be measured well, large cities have more players making change difficult.

In small cities, sprawl may not affect everyone's lives in the same magnitude (e.g., Houghton's "rush-minute"), but sprawl does exist and is noticeable. Sprawl is easier to measure in small cities and if measured well, policy is much easier to change as there are many fewer players involved and less existing development.

The Sprawl Scorecard provides insight to local and regional planners to mitigate sprawl in their regions. This research also offers researchers several paths for future work in all four areas of sprawl research.

Included with the development of the Sprawl Scorecard is original software written in Python using ArcGIS. The first program generated Extended Urban Clusters, based on Extended Urban Areas developed by Wolman et al. The second calculated 21 sprawl metric scores for each city and the third used principal components analysis to combine the metrics into four component scores and an overall Sprawl Score for each city.

Chapter 1: Introduction

This chapter includes a brief introduction to the history of the sprawl debate (Section 1.1), a discussion of the niche filled by this research (Section 1.2), the need for this research (Section 1.3), the specific research questions that were addressed (Section 1.4), and an outline of the dissertation (Section 1.5).

1.1 Background

In many planning meetings, urban sprawl is seen as the elephant in the room that no one wants to talk about. This is in part because its definition is nebulous and in part because many deem it to be either unobjectionable or incorrigible. Although most planners can intuitively distinguish sprawling development from more sustainable forms of development, there is a widespread inability to precisely define sprawl's characteristics. Planners are faced with the dilemma of dealing with a dichotomous group of individuals who have ambiguous beliefs on sprawl. Various groups argue whether sprawl is a problem, and whether it can be fixed if indeed it is a problem. With vague definitions and no direct way to measure sprawl despite a wide array of research on the matter, neither side has a great argument nor are they willing to pursue compromises. This fact has led to widespread haphazard planning in the United States which does little to satisfy parties on either side of the debate.

Sprawl in a city is widely regarded as a very complex system that is much easier to qualitatively describe than to quantitatively model due to several convoluted and overlapping variables. The growth of cities is extremely complex. Because even a small change in an urban area might affect the entire area in many ways, there are too many variables to model the system perfectly. Bruegmann concludes that "trying to understand the reciprocal relationships among city, suburb, and exurbs is like trying to focus the eye simultaneously on numerous objects ricocheting wildly around a confined space" [1].

1.2 Research Focus

Generally research divides the dynamic system of sprawl into four major realms of investigation including definitions, causes, components, and consequences. These realms as well as a working definition of sprawl are discussed in Chapter 2. Although there is also significant work being done on solutions to sprawl including transit-oriented development, infill development, open space preservation, and other urban planning techniques, these are not the focus of this research [2]. Instead, the thrust of this research is on the components realm of sprawl. The major question addressed in this research is *what metrics are best for quantifying sprawl?* To accomplish this, prior measures of sprawl were addressed, components of these measures were integrated, and a wide-ranging list of metrics were integrated in a comprehensive Sprawl Score.

Specifically, this study focused on determining appropriate measures to develop a Sprawl Score for 42 small pilot cities in Michigan. Details on the methods used to develop this score are discussed in Chapters 4-7.

The Sprawl Score contains component scores summarizing areas of concern including development density, land use mixing, activity centering, and street accessibility. With these metrics, policy makers and planners will be able to look at specific changes that can be made for each component and how this will improve their Sprawl Score by pursuing specific development programs to mitigate sprawl. Overall, both sides of the sprawl argument should have a better way to quantify their argument and this should lead to more compromises, and ultimately better planning and quality of life for citizens of these communities.

1.3 Research Need

Much of the research in the past 15 years has focused on the consequences of sprawl, but without an accepted sprawl measuring tool, much of this research suffers from either being too qualitative or using imprecise or incomplete metrics to measure consequences. It is clear from prior research on sprawl components that there is a need to validate a standard measurement strategy for sprawl. This study laid the groundwork for this effort by adapting the most comprehensive scoring mechanism to date for use in small cities [3,4].

The fundamental hypothesis for this study is that studying sprawl beginning at a smaller scale provides a more consistent and reliable way to measure sprawl. Most research to date has addressed sprawl on a large scale using America's megalopolises like New York, Los Angeles, and Chicago, although results have been quite different. Conflicting rankings that place Los Angeles and Chicago as the least sprawling cities in some studies and the most sprawling in others indicate that there is little consistency in measuring sprawl at a large scale. These conflicting measures only serve to broaden the belief that sprawl is either not a problem, or not a resolvable problem.

Unlike a computer that is created following systematic logic, sprawl development mimics natural systems and thus is very complex. Biologists would not be able to understand the human body without first understanding how single cell organisms function. Similarly, why would engineers and planners try to understand sprawl in a megalopolis without first understanding how small cities and micropolitan areas function?

It is easy to see why researchers studying sprawl have difficulty trying to measure sprawl when starting at a large scale. There are so many competing variables that overly complex systems of a megalopolis, such as Los Angeles, could be seen in a different light depending on which factors are measured and at what scale they are measured.

Scale itself can be a huge differentiator in measuring sprawl. Most groups citing New York as a prime example of a sustainable city often use only Manhattan for their study indicators [5], whereas those citing New York as a sprawling city often refer to the metropolitan statistical area or an even larger region [6].

For this study, the primary focus was on determining the causes of sprawl by researching the phenomenon at a smaller scale using extended urban clusters (EUCs). Urban clusters (UCs) are cities and surrounding areas with populations under 50,000 people in a single core whose boundaries are defined by the United States Census Bureau [7]. EUCs include additional territory around the UC are derived and discussed in Chapter 5.

This study included only Michigan cities (and their urban extents) with populations between 2,500 and 50,000 people. Although cities over 50,000 certainly exhibit characteristics of sprawl, they often have too many confounding variables at play to precisely determine the fundamental measures of sprawl. Cities with populations under 2,500 may exhibit a few of the characteristics of sprawl being measured, but do not have enough population to be reliably analyzed.

Several groups have determined a sprawl ranking for large American cities, however one gap in the literature is the lack of study of sprawl for small cities. The following points summarize the motivation for using small urban areas for this research:

- Past research results on large urban areas have been mixed and have not led to consensus.
- Small urban areas do have noticeable sprawl which makes it easier to judge the relationship between measures and reality, improving metric validation and comparisons between cities.
- Small urban areas provide a better mechanism for developing and measuring the critical components of sprawl as their variables are not as confounded as with larger urban areas.
- Measures developed and validated for small urban areas should scale up well to large urban areas.
- The likelihood that government and planning agencies responsible for development changes in small urban areas will be able to make systemic, impactful changes is much greater than in larger urban areas where there are many more organizations involved.

In addition, the measures determined to be appropriate and effective for small urban areas should scale well to larger cities and ultimately provide a more reliable measure that will complement past research on large urban areas.

1.4 Research Questions

The primary question addressed in this research is *what metrics are best for quantifying sprawl?* A discussion of this question is developed in the literature review (Chapter 3).

This study develops a method for measuring sprawl in small cities, using small cities in Michigan as a pilot group. The research addressed prior measures of sprawl, integrated components of these measures, and incorporated a wide-ranging list of metrics that helped lead to a comprehensive Sprawl Score.

Future research should branch from and build off this foundation by examining small cities across the United States and ultimately developing more precise measures for larger cities with the end goal of leading to a consensus in measuring sprawl. A consensus would allow policy makers and planners to better understand sprawl and pursue more logical development programs.

To answer the primary question, three additional questions were considered:

1. What measures have been used consistently to measure sprawl?
2. Does sprawl exist at small scales?
3. How do various measures of sprawl compare for the pilot cities of this study?

1.5 Dissertation Outline

Chapters 2 through 3 are introductory chapters defining the problem and reviewing the literature. Chapter 2 provides a background on urban sprawl along with a working definition of sprawl that was used throughout this report. Chapter 3 is a literature review of sprawl metric research.

Chapter 4 introduces pilot city selection and the tools used to analyze the datasets from various providers, including the US Census, to develop the Sprawl Score.

Chapter 5, 6, and 7 are technical chapters where mapping in ArcGIS and programming in Python are used to develop tools leading to the Sprawl Score. Chapter 5 reviews the selection of Extended Urban Clusters for the city scope, based on Extended Urban Areas developed by Wolman et al. Chapter 6 describes the calculation of the 21 sprawl metric scores for each city. Chapter 7 examines the use of principal components analysis to combine the 21 metrics into four component scores and an overall Sprawl Score for each city.

Finally, Chapter 8 concludes that sprawl exists and is measureable for small cities, offers advice to local and regional planners to mitigate sprawl in their regions using the results of their Sprawl Scorecard, and offers researchers several paths to use this research for future work in all four areas of sprawl research.

Chapter 2: Background on Urban Sprawl

One of the dilemmas with quantifying urban sprawl is that there are many definitions and lack of consensus as to what sprawl is, what causes it, or why it occurs. As a result, it difficult to study sprawl. For this reason, a working definition for this research was derived from the literature and defined at the end of this section to be used throughout this research. A working definition is needed if a rating system is going to be useful.

In the development of the working definition of sprawl (Section 2.5), a discussion of the realms of sprawl including the variety of definitions (Section 2.1), the causes (Section 2.2), the components (Section 2.3), the consequences (Section 2.4) are summarized.

2.1 Competing Definitions of Sprawl

Sprawl, also referred to as urban sprawl (to denote the origin of the sprawl) and suburban sprawl (to denote the location of the sprawl) has many definitions. A sampling of the leading ones include:

Urban sprawl is a form of spatial development, characterized by low densities, scattered and discontinuous “leapfrog” expansion, and segregation of land uses, encouraging the massive use of private vehicles and strip-malls; this form of development is found mainly in open, rural lands on the edge of metropolitan areas [8].

Sprawl (n.) is a pattern of land use in a[n] Urbanized Area that exhibits low levels of some combination of eight distinct dimensions: density, continuity, concentration, clustering, centrality, nuclearity, mixed uses, and proximity [9].

Sprawl can be defined as a pattern of urban and metropolitan growth that reflects low-density, automobile-dependent, exclusionary new development on the fringe of settled areas often surrounding a deteriorating city [10].

Sprawl is “low-density, scattered, urban development without systematic large-scale or regional land-use planning” [1].

In order to define sprawl, it is important to define its alternative, the traditional neighborhood, as well as another option, rural development.

A traditional neighborhood exists in an urban area [11]. Dwellings are relatively close to each other, daily destinations (work, school, shopping, civic) are walkable, public transit and walking dominate, and the street network is connective.

Rural development includes agricultural land and houses on larger, generally forested lots. Rural development is offered few, if any, municipal services by townships or

counties as compared to suburban development. Although rural living is a good option for many people, sprawl discussions generally focus on living in either an urban or suburban area. Minimal development in rural areas is generally considered inconsequential to sprawl, although over-developing rural areas into suburban areas is the fundamental nature of sprawl-development.

2.2 Causes of Sprawl

The cause of sprawl can be debated due to “chicken and egg” like arguments of whether certain conditions cause or are the result of sprawl. Regardless of this debate, the root cause of sprawl is and must be the short-term economic incentive for sprawl to exist, otherwise it would not exist. This simply means that it was perceived to be either less expensive or more desirable to build on undeveloped land further from the city center than on developed land closer to the city center.

Some argue that there is a demand for sprawl [12]. This demand stemmed from the onset of the automobile which spurred the growth of a highway network to support automobile transportation. There was certainly a huge demand, and a largely positive stigma, to move to the suburbs in the middle of the 20th century right after World War II. In fact, it was, and in many cases still is, the American dream. This dream continued throughout the 20th century as the automobile became the primary mode of transportation.

Some early researchers were discussing sprawl well before the advent of what we call sprawl today, when streetcars were causing a change from what had been the traditional development pattern for centuries [13]. Research expanded in the late 1980s and much of the substantial research in sprawl began in the mid-1990s.

Towards the end of the 20th century and into the 21st, more people, especially millennials, are demanding more walkable living [14]. This transformation is helping mitigate the demand for sprawl and is making it easier for sprawl opponents to make inroads in urban planning, however there is still demand for development resulting in sprawl as defined in this study.

Although sprawl is initiated due to the demand for sprawl, what facilitates the existence of sprawl? Sprawl is continuously enabled by many factors including population growth (e.g., immigration and baby boomers), lack of coordinated regional planning, and transportation infrastructure spending which subsidizes the costs of building on otherwise undeveloped land.

The fundamental element of determining sprawl is choice. Where a person, business, or group chooses to live based on benefits in the short term affects urban development patterns in the long term. The problem is that with each choice, the next choice becomes progressively skewed in the same direction in somewhat of a vicious cycle,

sprawl begets more sprawl. Decades of small choices have established sprawl in America and it is much harder to remove than to expand.

Many have claimed that affluence and personal preference make sprawl inevitable [1], but Lewyn argues that when one compares Europe and the United States, this theory of inevitability breaks down.

(1) Europe is far less automobile dependent than the United States; and (2) to the limited extent that Europe has sprawled, European governments' pro-sprawl public policies may be partially to blame. It logically follows that the Inevitability Theory is simply wrong—sprawl can be, and in fact has been, limited in the affluent societies of Western Europe [15].

2.3 Components of Sprawl

The component that almost all researchers agree on that measures sprawl is low density development, which amounts to the unnecessary overconsumption of land. However low density is not the only component of sprawl and this is where views diverge. A literature review on sprawl is presented in Chapter 3 and common themes emerge.

Some of these common themes include dispersed land uses (homes, schools, jobs, shopping, civic institutions, and recreation) [16] and poor street accessibility including lack of a connected street network and suburban inventions such as cul-de-sacs.

Squires summarizes many of the components of sprawl used in current research and these are discussed in much more detail in Chapter 3:

Among the traits of metropolitan growth frequently associated with sprawl are unlimited outward extension of development; low-density housing and commercial development; leapfrog development, “edge cities” and more recently “edgeless cities;” fragmentation of land use planning among multiple municipalities; reliance on private automobiles for transportation; large fiscal disparities among municipalities; segregation of types of land use; race and class-based exclusionary housing and employment; congestion and environmental damage; and a declining sense of community among area residents [10].

2.4 Consequences of Sprawl

Some now argue that the short-term benefits of sprawl were mostly perceived, not real, and that the negative consequences of sprawl far outweigh its benefits [17-20]. There are still some groups that benefit economically, but the externalities are generally much more dire than the gains for that group. The main problem is that most of the benefits of sprawl are short-term and most of the consequences are long term.

It is well known in economics and business that the short-term is of primary importance. This is not because people are willfully negligent of the long term, but because it is 1) easier to understand as well as quantify short-term gains and losses and 2) long-term costs are generally broader in scope (societal/environmental) than short-term gains which are more at an individual level. What this means is that most people will recognize a short-term benefit easier than they would recognize long-term cost. As an example, sprawl may offer less expensive land so a person can build a bigger house and have a bigger yard as compared to what they could get for the same amount at the city center. However this decision is rarely coupled with the fact that the person's increased transportation requirements will result in increased stress and pollution which could reduce the length of their life. This is how sprawl-living, by many people's and company's standards, is more desirable than urban living.

Even if there still is much demand for sprawl (either induced, perceived, or actual), the fact remains that sprawl is not sustainable. Although one could argue that growth of any kind is unsustainable (and they would be correct), it is interesting to note how much more unsustainable sprawl really is than alternatives such as high-density mixed-use design. The sustainability triangle (environment, society, economy), shown in Figure 2.1, is one which has been popularized in recent literature, and is the model can be used to look at the adverse effects of sprawl [21].

The foundational leg of the sustainability triangle is the environment. Without an environment, society could not exist and neither could the economic function of this society. Sprawl creates tremendous overconsumption of both land and resources in that more space is being used for less activities. Land which had been used previously for agriculture is now being converted to developed land. Traditional cities required one or two square miles of land with densities in the thousands of people per square mile. Each sprawl development can easily take up the same or greater land area for instead only hundreds of people. It is also important to note how much additional raw material is required to build these residential developments. The only research on positive environmental effects shows that environmental costs, such as pollution and wildlife habitat encroachment, are declining, but these arguments are generally myopic and do not consider entire metro areas [22].

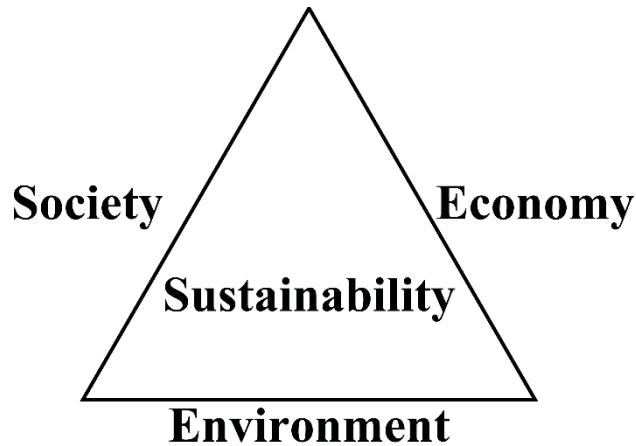


Figure 2.1: Sustainability triangle

The next leg of the sustainability triangle is society. A large problem that sprawl has caused is social reclusion due to lack of face-to-face social interaction. In a traditional city, it was very difficult and undesirable to go throughout the day without talking to tens of people, maybe even a hundred of them. In suburban life, one may see the same number of people, but there is not much value in talking to them. People in suburbia live in neighborhoods where many of them do not even know who their neighbors are nor do they care to. This social reclusion causes many other adverse societal effects.

One other broad social problem is increased stress. Because commutes are long and slow, people are not as sociable, and pressure to consume is high, most people act at a higher stress level in a sprawl environment. The speed at which people complete things is outstripping the quality of the things that they complete.

The third leg of the triangle, economy, is also important if a society is to be sustainable and healthy. Productivity is one of the most important drivers of the economy. The problem is that productivity is indirectly correlated to length of commute, stress, and many of the things that sprawl is predicated on. That is to say that as people become more stressed, they become less productive, and therefore the economy suffers. [23]

Sprawl has enabled population growth and in turn, population growth has enabled more sprawl [24]. The population of the United States has more than doubled since the advent of sprawl in the 1940s, and there are few signs of this declining any time soon due to continued immigration. The most unfortunate part about this statistic is that many city populations, especially in the Northeast, are declining, yet the overall population is soaring due to suburban growth.

Many of the benefits and consequences of sprawl are listed on the following page. These are broken down into the three components of the sustainability triangle, environmental, social, and economic. Although the list is not comprehensive, it provides good context to understand the impact of sprawl.

Economic Benefits of Sprawl [25]

- Decreased expense for land acquisition
- Increased short-term profits for corporations
- Increased decision-making ability (left open to the free market)

Economic Consequences of Sprawl [26,27]

- Increased costs to municipalities due to businesses externalizing costs (e.g., tax breaks, infrastructure, public transportation)
- Increased property taxes to residents to cover costs
- Decreased viability of city centers

Social Benefits of Sprawl [22]

- Decreased crime rates in sprawl
- Increased amount of land for residents versus urban dwellings
- Decreased noise and air pollution in the sprawl

Social Consequences of Sprawl transportation requirements

- Increased (generally automobile dependence) [28-31]
- Increased racial / ethnic disparity [10]
- Increased gap between classes [10,32]
- Decreased socialization / citizenship [33]
- Decreased physical activity leading to increased obesity [34-45]
- Increased traffic and pedestrian fatalities [46]
- Increased emergency response times [47]

Environmental Benefits of Sprawl [22]

- No substantial benefits

Environmental Consequences of Sprawl

- Increased consumption of land [16,48,49]
- Increased air pollution in city centers [50-52]
- Increased water consumption [53]
- Increased energy consumption [54]
- Decreased wildlife habitat and fragmentation / damaged ecosystems [48,49]
- Increased number of wasted resources and overconsumption [16]
- Decreased overall sustainability [55]

To close the discussion on consequences of sprawl and better understand how the consequences are interrelated, consider the following example.

A big-box supercenter selling department store items and groceries is built two miles from the downtown of a city. This allows for market efficiencies leading to increased convenience for consumers, reduced direct prices to consumers, and higher corporate

profits. But are these benefits worth the costs? The supercenter is built on the site of a former wetland, decreasing wildlife habitat and biodiversity. The store's target market includes all people living in the city. Over the course of 10 years, a downtown clothing shop, variety store, pharmacy, grocer, and electronics store (to name a few) all shut down leaving the only option for shopping for such items two miles away from downtown. This increases the need for people to use personal vehicles to commute to the shopping destination. There are also now five vacant buildings in downtown, which makes for a less viable downtown and can lead to further business closures, decreased tourism, and more crime, to name a few. All of the former employees of the closed business who had ample salary and benefits are now forced to commute further to work for less income or take up a new profession. Traffic congestion and travel time is increased as a result of more traffic to the area and increased number of traffic lights. Then the business decides that their big box is not a big enough box. They shut down the old location and move another two miles out from the center of the city. This leads to a whole slew of additional consequences. And this is just the tip of the iceberg as far as consequences of this type of scenario which has occurred and is continuing to occur across the United States and increasingly in other developed and developing nations.

2.5 A Working Definition of Sprawl

Causes and consequences of sprawl certainly help to understand sprawl, but a definition of sprawl should primarily focus on what sprawl is (the components). Therefore, the following definition is proposed:

Sprawl is low-density developed land on the fringes of an urban area that includes dispersed land uses and gaps of open space. It is characterized by residential subdivisions, strip malls including many big box stores, and a large private vehicle traffic to pedestrian traffic ratio, all flanked by small swaths of nonfunctional open space.

Although sprawl can be measured independently of a core, it is important to link sprawl with the corresponding development at the urban core. Therefore an urban area's Sprawl Score, as developed in this study, reflects the quantity of sprawl as it is related to the quantity and quality of urban development within the central business district. What this means is that even if two urban areas have an identical amount of sprawl, if one of them has a much greater amount of downtown health it will score better than the one with lesser downtown health. Downtown health refers to few vacant buildings, many pedestrians present, clean streets, and a high level of safety.

Chapter 3: Literature Review of Sprawl Metrics

Land development and growth has been a topic of urban geography research since the early 1900s, however more recently interest in sprawl has become quite multidisciplinary. Several researchers have attempted to tackle the incredibly complex system of sprawl across various fields including planning, public policy, geography, sociology, economics, and civil engineering. As presented in Chapter 2, the four major realms of research on sprawl are definitions, causes, components, and consequences. Although these are all intertwined, the focus of this study is regarding the measurement of sprawl using metrics. Although much progress has been made on developing and enhancing metrics to measure components of sprawl in the past 15 years, no consensus has been achieved.

Many researchers have identified ways to measure sprawl in an urban area, and have explained many of the characteristics that play a role in this phenomenon. For instance, population density is consistently included in sprawl measurement systems and there is a consensus among authors that low density is a factor in sprawl. Although population density is widely believed to be a necessary item in sprawl analysis, it is not sufficient to measure sprawl.

Studies such as Galster et al. (2001) and Ewing et al. (2002) laid the foundation for studying sprawl using metrics tied to the fundamental components of sprawl [9,56]. More recent studies have begun to selectively evaluate these measures and have attempted to come to a consensus means of measuring sprawl, however no consensus has been achieved. Ewing et al.'s updated 2014 study has the potential to begin building consensus, thus this work became the primary source for this research [3].

This literature review of the research of sprawl metrics focused on three major aspects:

- Inability to consistently rank sprawl for large cities (Section 3.1)
- Variety of metrics used (Section 3.2)
- Using remote sensing to define urban areas and measure sprawl (Section 3.3)

Additional considerations are outlined in Section 3.4 and a summary of metrics and the list of chosen metrics are presented in Section 3.5.

3.1 Inability to Consistently Rank Sprawl for Large Cities

There have been several studies to comprehensively rank sprawl in large cities in the United States, with widely different results. Four of the major foundational studies are summarized in Table 3.1. This table shows the best five cities and the worst five cities in each study. Also, six standard cities are used to show how each measure widely varies with respect which cities are most and least sprawled. These are standardized

using each study's final weighted average to a ranking from 0 to 100, with 0 indicating considerable sprawl and 100 being less sprawl.

Table 3.1: Comparison of city rankings among four foundational studies

Study	Ewing et al. [56]	Galster et al. [9]	USA Today [6]	Kahn [57]
Year	2002	2001	2001	2001
# of Metrics	22	12	2	1
Categories of Metrics Used	Density (7) Mix (6) Centers (6) Streets (3)	Density Continuity Concentration Clustering Centrality Nuclearity Mixed Uses Proximity	Centering - Current Centering - Temporal	Employment Decentralization
New York	100	100	79	94
Chicago	65	7	73	25
Los Angeles	54	41	80	27
Atlanta	27	0	18	24
Dallas	39	21	61	34
Portland	68	N/A	52	100
Least Sprawl	New York, NY Jersey City, NJ Providence, RI San Francisco, CA Honolulu, HI	New York, NY Philadelphia, PA Chicago, IL Boston, MA Los Angeles, CA	Laredo, TX Lincoln, NE Colorado Springs, CO Anchorage, AK Fargo, ND	Portland, OR New York, NY Anaheim, CA Denver, CO Milwaukee, WI
Most Sprawl	Riverside, CA Piedmont Triad, NC Raleigh, NC Atlanta, GA Greenville, SC	Atlanta, GA Miami, FL Detroit, MI Denver, CO Dallas, TX	Ocala, FL San Luis Obispo, CA Johnstown, PA Charlottesville, VA Sumter, SC	Detroit, MI Tampa, FL Oakland, CA Atlanta, GA Chicago, IL

As can be seen from these rankings, New York tends to be rated as a low-sprawl metropolitan area and Atlanta tends to be rated as a high-sprawl metropolitan area. Los Angeles, Chicago, Dallas, and Portland however tend to vary across the board. It should be noted that all of these studies focus on large cities with populations exceeding 50,000 people.

Both Ewing et al. and Galster et al. strived to summarize the research of the 1990s to derive a comprehensive sprawl scoring mechanism. Ewing et al.'s approach was to utilize many metrics to derive component scores in a few key areas whereas Galster et al.'s approach was to determine eight distinct areas with one metric each.

As not many sprawl scores had been released at the time, the USA Today score received a fair amount of press even though only two metrics based on the same category, centering, were used to develop the score [6]. Kahn's score offered even less breadth as it was derived purely based on one metric, the degree of employment decentralization in a metro area [57]. The Ewing et al. and Galster et al. studies by far the most comprehensive of the era.

Lopez also developed a sprawl index using a similar strategy to the USA Today methodology [58]. Lopez's metric dealt with percentage of people living in low density versus high density census tracts. The interesting portion of this study was that his index was defined for 4 different decennial census years (1970-2000), which allows comparing cities "growth" of sprawl, or "second order" sprawl growth. Atlanta increased 20 percentage points from 1970 to 1980, but stayed relatively stable to 2000. Portland showed a decrease of seven percentage points from 1980 to 2000, perhaps giving credence to the argument that their urban growth boundaries have at least been metering sprawl [59].

It should be noted that many of these rankings were developed over 15 years ago. More recent studies have realized these discrepancies and tried to focus on developing better metrics as opposed to rankings. Most current studies that do use specific geographic boundaries do so using only a few case study cities. The one exception is Ewing et al.'s updated 21 metric sprawl score, and this is discussed in Section 3.2 and used in the research [3].

3.2 Variety of Metrics Used

Seeing an emerging body of research and a necessity for identifying the components of sprawl, the Transportation Research Board (TRB) commissioned a major study. Through their Transit Cooperative Research Program (TCRP), they released a report entitled "The Costs of Sprawl - Revisited" summarizing research on sprawl to date in 1998 [60]. Since then, researchers have been striving to get a grip on which metrics best measure sprawl. The TCRP report outlined the following 10 defining characteristics of Sprawl:

- Low density
- Unlimited outward extension
- Land uses spatially segregated
- Leapfrog development
- No central ownership on planning
- Transport dominance by motor vehicles
- Highly fragmented land-use governance
- Great variance in local fiscal capacity
- Widespread commercial strip development
- Reliance on filtering for low-income housing

This report spurred the two most rigorous early studies which developed metrics to rate sprawl in the United States, as discussed in the prior section, which were Galster et al. and Ewing et al. [9,56]. Since then, many groups have continued to investigate these and other measures. Table 3.2 summarizes the characteristics used in the most prominent research over the past 15 years, organized by date of publication. Table 3.3 lists a description of each measure synthesized from definitions in the sources using the measures.

Table 3.2: Metrics used to measure sprawl in previous research

Source	[61]	[57]	[6]	[9]	[56]	[62]	[63]	[64]	[65]	[66]	[67]	[3]	Totals
Year	98	01	01	01	02	06	07	07	07	10	13	14	
Density	X		X	X	X	X	X	X	X	X	X	X	11
Accessibility	X	X		X	X	X				X	X	X	8
Centrality				X	X	X		X	X	X	X	X	8
Continuity				X		X	X	X	X	X			6
Porosity	X			X		X	X	X	X	X			6
Mixed Use				X	X	X				X	X	X	6
Historical Change	X		X				X		X				4
Complexity						X		X					2
Impervious Surface						X			X				2
Clustering				X									1
Nuclearity				X									1
Total Metrics	4	1	2	12	22	17	9	7	25	6	18	4	Avg. 10.6

Table 3.3: Definitions of metrics used to measure sprawl in previous research

Metric	Definition
Density	Population per unit area (can also be density of a specific function)
Accessibility	Proximity to services as well as accessibility via the transportation network
Centrality	Proximity of services to the core / CBD
Continuity	Uniformity of density gradient and no leapfrog development or scattering
Porosity	Ratio of open space to urban area
Land Use Mix	Diversity in land uses / proximity of housing to other land uses
Historical Change	Historical data used to compare growth patterns
Complexity	Irregularity of urban area's patch shape
Impervious Surfaces	Amount of impervious surface in urban area
Clustering	Clustering within small units scattered throughout the urban area
Nuclearity	Mononuclear versus multinuclear development

Ewing et al.'s study sponsored by Smart Growth America involved one of the most rigorous metrics of its time [56]. Prior studies had focused on one or only a few metrics, whereas this study focused on 22. The 22 were grouped into categories of density, land use mix, degree of centering, and street accessibility with most of the data coming from the US Census Bureau, the Census Transportation Planning Package (CTPP), a previous study by Glaeser et al. on zip code business patterns [68], and the American Housing Survey (AHS). The list of metrics used in this study are shown in Table 3.4. One of the main findings from this study was not the sprawl score for the areas, but the quality and quantity of the metrics used to operationalize sprawl. Many of these metrics are improved and updated in Ewing et al.'s 2014 study which was used as a basis for the study [3].

Table 3.4: 22 Metrics across four dimensions used by Ewing et al. [4]

Density	Population density
	Percentage living at low suburban density
	Percentage living at transit-supportive urban density
	Estimated density at center of CBD
	Population density of urban lands
	Average lot size for single family dwellings
	Weighted density of all population centers in a metro
Land Use Mix	Percentage with businesses/civic areas within block of home
	Percentage with shopping within 1 mile
	Percentage with public elementary school within 1 mile
	Job-resident balance
	Population-serving job-resident balance
	Population-serving job mix
Centering	Variation of population density across census tracts
	Density gradient
	Percentage less than 3 miles from CBD
	Percentage less than 10 miles from CBD
	Population relating to centers within same MSA
	Ratio of density of centers within same MSA to related center
Street Accessibility	Average block length in urbanized portion of metro
	Average block size in square miles
	Percentage of small blocks

Galster et al.'s study was the second of the two rigorous sprawl measurement studies done at the turn of the century [9]. Rather than focusing on the measurement of many cities, it concentrated on developing and defining metrics which could be used to measure sprawl, as shown in Table 3.5 below. A significant element of this study was the scale of metropolitan area used. The researchers decided to use Urbanized Areas (UZAs) as opposed to the larger Metropolitan Statistical Areas (MSAs) used by most other studies. Critics of this research, as well as more recent research by some of these

authors, noted that UAs did not include a lot of the suburban and exurban territory which is generally considered to be related to sprawl of the metro area. Wolman et al. (including Galster) would refine their geographic unit by creating Extended Urban Areas (EUAs) which is modified and used in this study [69].

Table 3.5: 12 Metrics across 8 dimensions used by Galster et al. [9]

Density	Number of housing units/employees in UA per square mile
Continuity	Proportion of developed grids in UA to undeveloped
Concentration	Proportion of very high density grids to all grids
	Variation of housing unit density among grids
	Share of units needed to shift to achieve uniform distribution across UA
Clustering	Degree of clustering within quarters of one-mile grids
Centrality	Average distance of a land use from the CBD
	Accumulation of land use as distance from CBD increases
Nuclearity	Number of nodes in a UA
	Ratio of housing units/employees in largest nucleus to all nuclei
Mixed Use	Average density of a particular land use in another land use's area
Proximity	Degree to which a land use or pair of land uses are close to each other across UA

The USA Today and Sierra Club studies were examples of many of the less sophisticated studies linking sprawl mostly to density [6,61]. Although most research on sprawl will list density as a key factor, recent studies have realized this is only one of many factors. The one interesting measure that the USA Today study considered was historical change of density from 1990 to 2000, as historical growth patterns certainly do play a significant role in measuring sprawl. The Sierra Club study used many subjective measures identified by a group of committee members.

Kahn's study was primarily focused on a theorized consequence of sprawl, the black/white housing consumption gap [57]. However, Kahn did use a unique measure of sprawl to determine a sprawl ranking at the beginning of his study. He used location of employment facilities in the metropolitan area as his sole measure of sprawl. The critical radius was 10 miles; if all businesses were located within this radius, a city would receive the lowest sprawl rating whereas if all business were located outside the radius, the city would get the highest sprawl rating. The limitation of this metric is that city populations should allow for a changing critical radius instead of an arbitrary constant radius. One of the main contributions of this study was a procedure to systematically define the central business district (CBD) of a city.

Lowry and Lowry statistically compared 18 metrics across four categories using three scales of defining the urban area of a city using Salt Lake City as a case study [67]. An interesting piece of their research was the use of three distinct zones in a city, classified as pre-suburban, suburban, and late-suburban. After comparing their 18 metrics at the three different scales, the authors concluded that 13 of these metrics were found to best

quantify urban form: median single family residential lot size, housing density, average household size, mean distance to commercial zones, mean distance to K-12 schools, street connectivity, median perimeter of blocks, dendritic street pattern, median length of cul-de-sacs, land use contiguity, land use richness, population percentage working outside the city, and renter-owner balance.

In two papers in *Ecological Indicators*, Jaeger et al. discussed a thirteen point suitability criteria for sprawl measurements testing four potential sprawl measures used in other research [70,71]. In addition to the contribution of selection criteria, the first paper stated an important conclusion:

For urban sprawl, the ideal case would be that one indicator quantifies the degree of urban sprawl, while a set of additional indicators measure relevant causes, consequences, and attributes of urban sprawl.

Zito and Salvo as well as Schwarz took a transportation-centric approach to sprawl using European cities as their case study [72,73]. Zito and Salvo looked at 32 indicators across eight macro-categories surrounding transportation and planning aspects of the design of the city. Their eight macro-categories were planning, public transportation, transportation demand management, economics, private transportation, externalities, energy consumption, and co-benefits. Although some of their metrics are related to consequences of sprawl, many of the unique metrics used could be good component indicators as well. In addition Schwarz offers a comprehensive table of 27 common metrics that have been used in several related studies [73].

3.3 Using Remote Sensing to Define Urban Areas and Measure Sprawl

Recently researchers have used remote sensing strategies to aid in defining boundaries for urban areas as well as for landscape-based metrics to measure sprawl. With the increase in the quantity and quality of imagery, this approach is gaining prominence.

Huang et al.'s research explored cities across the world using some unique measures of sprawl as well as a novel method of determining the scale area to be studied [64]. Satellite imagery was used to determine the area of study for each city, with the major limitation being availability of data and cloud cover causing a limited selection of cities to be chosen. The main contribution from this work was the definition of the urban areas using remote sensing strategies instead of population and housing data.

In Sutton's research, nighttime imagery was used to determine the extent of an urban area to use in sprawl measurement [74]. He used two thresholds yielding 244 and 300 areas in the United States, of which many aligned with urban areas. The study did show a wider variation in population in his estimated urban areas versus actual urban area populations. One benefit of this study is that areas are better adjusted for scale, making for better comparisons between large and small urban areas.

Ji et al. used images during fast-growing vegetation periods in Kansas City to first determine a boundary area for the metropolitan region and then to look at temporal expansion of the built area for the region [75]. Included in their research was a discussion of patch identification. Patches, groupings of pixels of the same land cover classification, are used throughout the majority of remote sensing sprawl research as these are defined by land use patterns within the region and are agreed to play multiple roles in sprawl [64].

Wilson et al. devised a complex algorithm to land cover data to create urban growth maps [76]. The group determined five distinct urban growth patterns in their Northeastern United States study area: infill, expansion, isolated, linear branch, and clustered branch. These findings are significant because these five distinct patterns of growth resemble varied levels of sprawl development.

Herold et al. provided a discussion about the importance of using the highest resolution land cover data available when measure characteristics of sprawl [77]. They also discussed the importance of using a robust collection of metrics in order to fully characterize sprawl in the study area.

Bhatta et al. summarized the development of landscape metrics and concluded that although there were many landscape-based metrics developed in the past few years, the concept still needs refinement [78]. They also add to the case that, in general, there will be higher demand for sprawl metric development in the future.

Even Ewing et al., in their recent study, have incorporated one element of remote sensing. Land use data developed from remote sensing are used to determine the percentage of population in urban lands [3].

3.4 Other Considerations

This section presents other research considerations related to the measurement of sprawl. Among these are walkability / livability (Section 3.4.1), environmental impacts (Section 3.4.2), basic psychology (Section 3.4.3), and empirical studies (Section 3.4.4)

3.4.1 Walkability / Livability

Two recent topics that certainly pertain to level of sprawl in a city are walkability and livability. Recent studies have developed a user-friendly website to determine the Walk Score of an area [79]. Walk Score can be utilized for a variety of metrics, specifically centrality and land use mix. Walk Score releases rankings for Walk Score, Transit Score, Bike Score, and City and Neighborhood Rankings. Walk Score is such an integral factor in measuring elements of sprawl that Ewing et al. included the weighted average Walk Score as one of their metrics [4].

Along similar lines, groups such as the Economist Intelligence Unit [80], the Mercer Quality of Living Survey [81], and AARP, formerly the American Association of Retired Persons [82], rank cities as most “livable.” These studies are of particular interest because of the fact that many of their indicators will overlap with sprawl indicators. The Mercer survey alone uses 39 indicators. By studying these links in more detail, metric viability can be assessed and potential conclusions can be marketed more appropriately to planning groups and municipalities.

3.4.2 Environmental Impacts

Studies on environmental impacts may also prove useful in the sprawl discussion. Particularly when a change in land cover strategy occurs for measuring environmental impacts, these measures will begin to overlap with the sprawl discussion [83]. Research at Michigan Tech has shown locally how Houghton has changed ecologically in tandem with urban sprawl [84]. Figure 3.1 below, reproduced from the report, shows the changing landscape of the area in Houghton that has been most wrought by ecological destruction and low-density, sprawl-related development over the past forty years. This report helps lead into the analysis of the Sprawl Score when looking at Houghton as a case study to draw parallels between metrics on environment and sprawl.

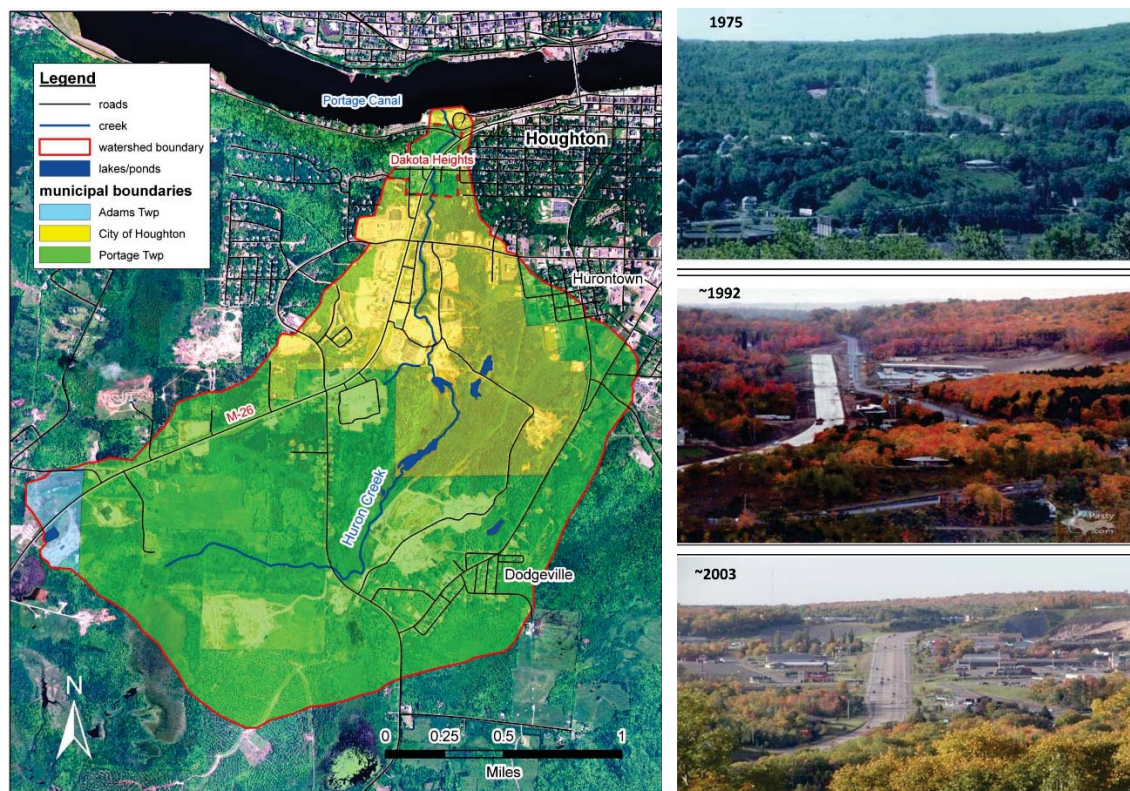


Figure 3.1: Changing land use patterns of Houghton, MI over past forty years [84]
 Courtesy of the Alex Mayer, Michigan Technological University, Houghton, Michigan.
 See Appendix A for copyright permission form (signatures redacted for privacy)

3.4.3 Basic Psychology

An important aspect in developing the Sprawl Score was identifying the right objectives and metrics to assess the situation accurately and consistently for each city. From the prior research discussed earlier, there have been hundreds of metrics used ranging from statistical measurements of census data to social/psychological studies to remote sensing of imagery. One goal in this study was to determine which metrics were the most appropriate to calculate the Sprawl Score.

Before metrics are developed, it is important to take a step back and look at the bigger picture. What attributes of an urban area make it more versus less sprawled? To answer this question, a step away from urban planning to basic psychology needs to be taken. Consider what people do in their daily lives and what a city is used for. From Maslow's hierarchy of needs, human needs can be broken down to physiological, safety, love, esteem, and self-actualization [85]. Among the main physiological needs, humans need access to shelter, food, and water, which can be simplified to housing and grocery stores in modern society. For safety and security, humans need government and civic services as well as employment. The higher order needs of love, esteem, and self-actualization begin to bring in requirements of recreation which can take the form of parks, athletic venues, airports, and other services one would expect to find in an urban area. All of these items require some mode of transportation which necessitates a transportation network to be developed to get between venues, whether it be walking, taking public transit, or taking a personal vehicle.

In summary, the typical components of any urban area are residential dwellings, commercial/industrial facilities, civic/cultural sites, shopping retailers, and transportation networks. In a traditional urban area, the CBD housed all of these sites in a mixed use environment. With the advent of faster transportation methods, these services have been able to separate and spread out while the direct costs of time and income have not been affected. This is the fundamental underpinning of sprawl and must be inherent in any properly calculated sprawl score. Categories that directly coincide with this trend include density, land use mix, and accessibility, all of which are used to calculate the Sprawl Score.

3.4.4 Empirical Studies

Over the past 10 years, the author has traveled by automobile through every county and most county seats in the continental United States by vehicle. While empirically observing downtown and suburb interaction in large and small cities, several patterns emerged:

- It was often faster to drive through downtown than to drive through the suburban areas surrounding downtown
- Downtowns all seemed regional and somewhat unique
- Suburbs all looked and felt identical no matter the region
- Architecture in downtowns was inspired as suburban architecture was mainly functional
- Many downtowns were necrotic (e.g., many closed storefronts, no pedestrians) even though their suburbs were bustling, whereas there were few to no examples of the opposite
- Older bypasses were filled with storefronts and traffic signals and new bypasses around them were also beginning to fill up with storefronts and traffic lights
- Older big-box stores closer to downtown sat vacant as newer, bigger big-box stores (most of the time, the same companies) were built further from the downtown
- Downtowns that had been improved by bringing in more stores and making them more walkable were filled with people and were enjoyable places to be
- Not just large cities have sprawl; small cities have sprawl too and in many cases the effects seem more damning on small cities

Over the course of these travels, the author has postulated many potential metrics to measure sprawl which are summarized in Table 3.6 [86]. These are summarized in six dimensions including density, land use mix, centering, transportation / accessibility, historical relationships, and other, uncategorized metrics. Many of these overlap with the current research, but many are unique and although they were not used in this study, the gamut of different measures is useful to consider for future work.

Table 3.6: 38 Metrics across six dimensions hypothesized by author

Density	Density of CBD
	Ratio of density of CBD to density of EUC
	Ratio of area of CBD to area of EUC
	Net residential density
	Net commercial density
	Ratio of living in EUC to working in EUC
	Ratio of living in CBD to working in CBD
	Ratio of number of dwellings in CBD to EUC
	Average block size
Land Use Mix	Percentages of buildings condemned in CBD
	Percentages of buildings in use in CBD
	Ratio of new businesses starting in CBD versus outlying areas
	Footprint of big box stores outside CBD
	Footprint of retail stores inside/outside CBD
	Number of locally owned business versus chains
Centering	Distance from courthouse, city hall, schools, etc. to center of CBD
	Distance from population center to center of CBD
Transportation / Accessibility	AADT of main street in CBD
	AADT of suburban arterials
	Commuteshed
	Commuteshed per capita
	Average commute times
	Transit usage statistics
	Vehicle miles travelled in all city (CBD and suburbs)
	Vehicle ownership statistics
	Walkability score of CBD
	Walkability of other key areas
	Bikeability statistics
	Bypass routes and effects
	Number of cul-de-sacs
Historical Relationships	Shape of city / changes over time
	Historic values and trends
	Suburbs with no center and large pop increase in recent history
Other Metrics	Happiness
	Livability
	Building height versus population
	Neighborhood completeness [87]
	Ratio of jobs to population

3.5 Summary of Sprawl Metrics and Metric Selection

After reviewing the literature, it is clear that although there is no consensus on a sprawl scoring mechanism, there are a variety of metrics that have been used. This being the case, the research presented in Ewing et al.'s 2014 report on sprawl was used as a primary reference for this project. Their 2002 research was already one of the top measurements of sprawl components and with the integration of elements such as land use from remote sensing and the Walk Score, the study has the best chance to gain consensus over the next 10 years. These metrics are listed in Table 3.7.

Table 3.7: 21 Metrics across four dimensions used by Ewing et al. (2014) [4]

Development Density	Population density
	Employment density
	Percentage of the population living at suburban densities
	Percentage of the population living at urban densities
	Net population density of urban lands
	Estimated density at the center of the metro area
	Population density of CBD
	Employment density of CBD
Land Use Mix	Job / population balance
	Degree of job mixing
	Walk Score
Activity Centering	Coefficient of variation in block population densities
	Coefficient of variation in block employment densities
	Density gradient moving outward from the CBD
	Percentage of population in the CBD
	Percentage of employment in the CBD
Street Accessibility	Percentage of small urban blocks
	Average block size
	Average block length
	Intersection density
	Percentage of 4-or-more-way intersection

Chapter 4: Study Methodology and Tools

Research has been conducted in five phases: the selection of cities to study, the decision on where to draw the line around the city for analyzing sprawl, the selection and calculation of sprawl metrics, the calculation and analysis of the Sprawl Score, and the recommendation of future work. Each of these elements is outlined below.

1. 42 Pilot cities were selected from among the chosen scale range (2,500 to 50,000) of small cities in Michigan. These were selected to narrow the study and select common groups for analysis. This process is detailed in Section 4.1.
2. Extended Urban Clusters (EUCs) were created to achieve the ideal geographic scope of the research area for pilot cities. The process is outlined in Section 4.2 and the algorithm to create the EUCs is the focus of Chapter 5.
3. Twenty-one sprawl metrics were calculated for each EUC in the study determined by using Ewing et al.'s 2014 approach [3]. The reason for selecting Ewing et al.'s approach is outlined in Chapter 3 and the algorithm to calculate the 21 metrics is outlined in Section 4.3 and is the focus of Chapter 6.
4. The Sprawl Score for the 42 pilot cities was calculated using the chosen metrics. The algorithm is introduced in Section 4.3 and the full algorithm and results are the focus of Chapter 7.
5. Conclusions were drawn and suggestions for future work were made. These are discussed in Chapter 8.

Several datasets and tools were used in this study. The most important dataset was the United States Census Bureau's block data and demographic data from the 2010 census [88]. Other datasets used are described in Chapters 5, 6, and 7.

The main tool used in the research was ArcGIS, a Geographical Information System (GIS) package produced by ESRI [89]. To develop the algorithms created in Chapters 5, 6, and 7, Python was used to create scripts, which are computer programs that run a sequence of commands [90]. Python is an open source programming language which is compatible with ArcGIS and includes the ability to call ArcToolbox applications.

4.1 Pilot City Selection Methodology

As sprawl is a consequence of moving away from a centralized city, pilot cities must be selected from a group of cities. What constitutes a city's extent is well described by the United States Census Bureau's definition of urban areas [91]. By this definition, an urban area is any area with a central core with a total area population of 2,500 or greater. Urban areas (UAs) are sub-classified into two sub groups: urban areas with a

population of at least 50,000 are classified as urbanized areas (UZAs) and the rest of the areas (between 2,500 and 50,000) are classified as urban clusters (UCs). Pilot cities were selected from among Michigan's 116 urban areas as defined and for this study, only urban clusters were used which decreased the number of possible cities for inclusion to 95.

Although all 95 of these cities are the proper size range for the study group, some of these cities are not ideal for various reasons. First, economic and cultural influence of the city must be considered. For instance, Goodrich is essentially a bedroom community for people working in Flint, Detroit, and the Detroit suburbs. Including Goodrich in this study would not align with the study's mission of focusing on small communities that are not widely influenced by large metropolitan areas. Thus, the next elimination criteria used was to remove UCs within a county that contains a nontrivial amount of a UA, which reduced the total prospective cities to 65.

Next, UCs were eliminated if they contained territory from multiple states. Statewide legislation and incentive programs have a lot to do with sprawl development patterns, so these hybrid UCs were removed to reduce this factor from convoluting the pilot city's Sprawl Score results. This criteria removed three more cities, all in the Upper Peninsula of Michigan, from the pool reducing the number eligible to 62.

Along similar lines to metropolitan area influence is county influence. Counties are utilized by most groups as boundaries of their general area. The county seat is generally, but not always, the most influential city of a county. As such, the number of UCs per county in this study was reduced to one. Although other UCs in the county may be involved with sprawl of the main UC in the county, this is considered as another variable in the conclusions of the study rather than an elimination criteria for all UCs in such a county.

To pick one UC per county, if the UC containing the county seat had either the largest population or population density, implying it has the greatest influence in the county, it was chosen. If there was a UC in a county with no seat present, or no seat that had the largest population or population density, the UC with the largest population density was chosen. This reduction resulted in the final group of 42 cities in the pilot group.

Table 4.1 lists all 42 cities in the pilot group. This table lists cities by classification, which is used in the final Sprawl Score calculation to compare only cities of a similar size and regional influence. The table also lists the county and prosperity region [92].

Table 4.1: List of 42 pilot cities in Michigan for this study

Classification	EUC Name	Primary County	Prosperity Region	Population (2010)	Area (mi ²)	Population Density
Mid-Size Regional (25,000-49,999) 4	Traverse City	Grand Traverse	Northwest	47109	43.62	1079.9
	Adrian	Lenawee	Southeast	44823	24.02	1865.8
	Mount Pleasant	Isabella	East Central	37447	16.63	2252.2
	Marquette	Marquette	Central UP	26946	15.68	1718.9
Small Regional (10,000-24,999) 14	Owosso	Shiawassee	East	22426	11.25	1993.3
	Escanaba	Delta	Central UP	20850	17.94	1162.0
	Alma	Gratiot	East Central	16924	7.43	2276.7
	Coldwater	Branch	Southwest	16876	11.52	1464.5
	Houghton	Houghton	Western UP	15452	8.46	1826.8
	Ionia	Ionia	West	14409	7.15	2015.4
	Alpena	Alpena	Northeast	14258	9.46	1506.9
	Big Rapids	Mecosta	West Central	14241	9.32	1528.5
	Lapeer	Lapeer	East	13424	10.16	1321.0
	Sault Sainte Marie	Chippewa	Eastern UP	13114	5.90	2223.4
	Sturgis	Saint Joseph	Southwest	13040	9.56	1364.6
	Cadillac	Wexford	Northwest	11690	8.74	1337.6
	Hillsdale	Hillsdale	Southeast	11646	7.52	1548.2
	Ludington	Mason	West Central	10710	8.43	1269.8
Large Subregional (5,000-9,999) 7	Greenville	Montcalm	West	9743	6.17	1580.0
	Manistee	Manistee	Northwest	9606	7.56	1270.0
	Houghton Lake	Roscommon	Northeast	8300	8.90	932.2
	Gaylord	Otsego	Northeast	8298	10.21	813.1
	Petoskey	Emmet	Northwest	8210	5.97	1375.2
	Hastings	Barry	West	7713	4.69	1644.9
	Caro	Tuscola	East	5113	3.53	1449.6
Mid-Size Subregional (2,500-4,999) 17	Cheboygan	Cheboygan	Northeast	4517	3.48	1297.3
	Fremont	Newaygo	West Central	4496	3.15	1427.0
	East Tawas	Iosco	Northeast	4372	3.15	1388.1
	Charlevoix	Charlevoix	Northwest	4179	4.53	922.8
	Grayling	Crawford	Northeast	3858	4.13	933.7
	Harrison	Clare	East Central	3589	3.21	1116.4
	Bad Axe	Huron	East	3490	2.49	1399.5
	Manistique	Schoolcraft	Central UP	3482	3.46	1005.1
	Newberry	Luce	Eastern UP	3225	1.49	2165.3
	Iron River	Iron	Western UP	3208	2.03	1578.8
	Munising	Alger	Central UP	2972	3.28	905.4
	Gladwin	Gladwin	East Central	2934	2.72	1080.5
	Sandusky	Sanilac	East	2775	1.77	1568.2
	Kalkaska	Kalkaska	Northwest	2668	2.35	1135.5
	Rogers City	Presque Isle	Northeast	2560	1.40	1824.3
	Hart	Oceana	West Central	2556	1.98	1289.4
	Saint Ignace	Mackinac	Eastern UP	2531	2.13	1186.9

Figure 4.1 shows the location of the pilot cities in the study. The chosen pilot cities are distributed geographically with cities included from all regions except South Central (Lansing) and Metro Detroit.

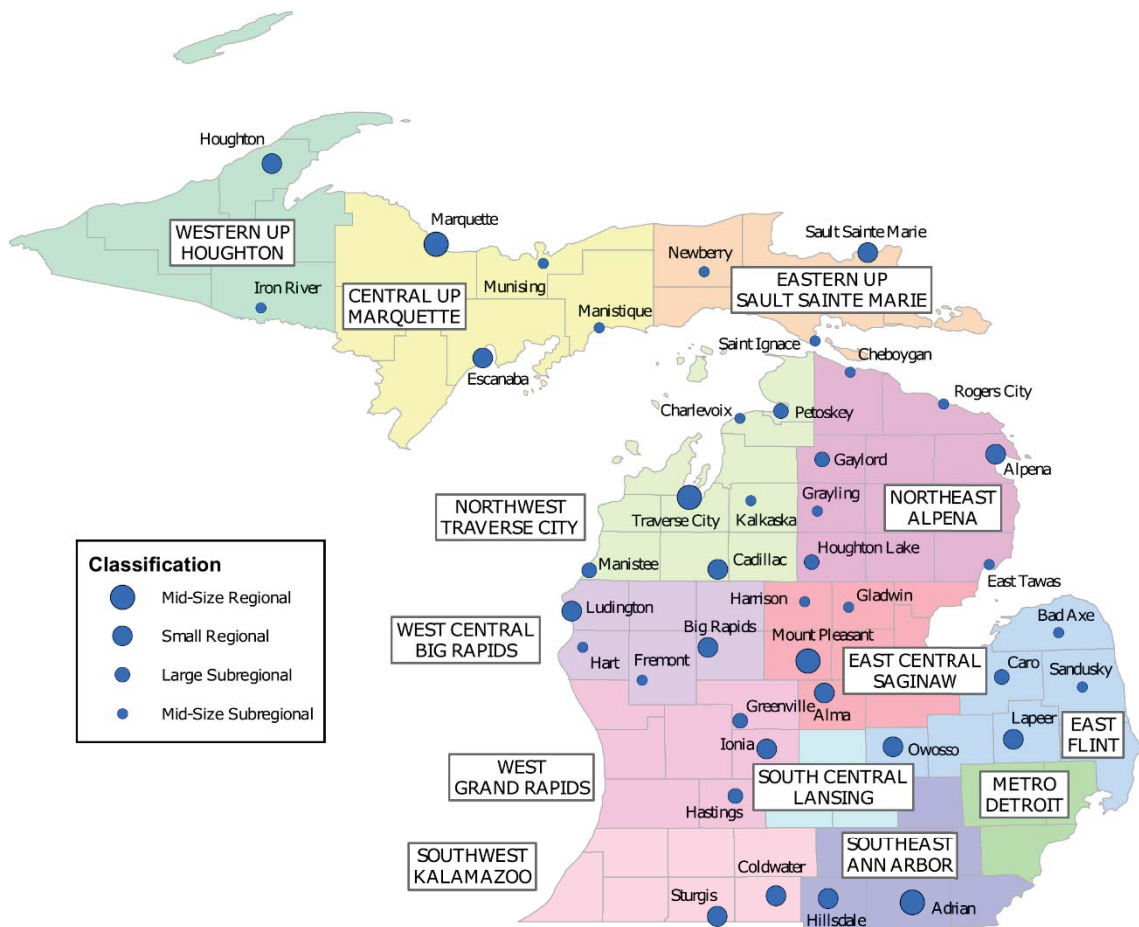


Figure 4.1: Map of Michigan pilot cities by population and region

4.2 City Scope Selection

When one is asked where they are from, a person will generally give a different answer depending on who they are responding to and the person's perception of the other's knowledge of their home area. For example, consider a person from the west central portion of rural Blendon Township in Michigan. To people from other parts of their township, this individual may state they are from Borculo, a small unincorporated community on the western edge of the township. To those from Holland, they may state that they are from Zeeland. To those in Michigan, but not in their county, they might say they are from Holland. To those living in another region of the United States, they may claim to be from Grand Rapids. Finally, to those from another country, they might simply claim to be from Michigan, or perhaps even just the Midwest of the United States.

The important point in the previous scenario is that cities have spheres of influence well beyond their boundaries. So the appropriate question for all sprawl studies is how far this influence matters pertaining to sprawl and measurement thereof. This is a standard Goldilocks syndrome in that some geographical units are too large to define areas for sprawl and some are too small.

Arguably the most important aspect about measuring any social or environmental trend is identifying a definitive geographical region as the study area. If an area is chosen haphazardly, then the results, no matter how statistically significant, will be meaningless at best and misrepresentative at worst. The most severe example of misrepresentation through geographic boundary selection is the gerrymandering that occurs when deciding on boundaries of electoral districts. Even when the intent is not malicious, the results may lead to misinterpretation.

The possible choices of geographical regions were many and generally based on political boundaries. The array of geographical areas are presented in Section 4.2.1, followed by an analysis of these areas in Section 4.2.2, concluding with the selection of a geographic area for this research in Section 4.2.3.

4.2.1 Range of Geographical Areas Defining a Cities Scope

There are many geographical areas defined by the US Census covering the entirety of the United States with no gaps. Each is shown in Figure 4.2 below and are outlined in brief from smallest area to largest area [93].

The Census defines blocks to be the basis for all tabulated data at all other geographic levels. Blocks are bounded by visible features. In a downtown, blocks are bounded by each street in the downtown grid. In suburban and rural regions, nonvisible features such as property lines and political boundaries are used. Block groups are collections of blocks containing between 600 and 3000 people. Tracts are larger collections of block groups containing between 1200 and 8000 people. Both are designed with much input from local governments.

Administrative units are defined by each state to separate the state into logical sub-units for many purposes. Counties are the largest sub-state administrative region and vary in area from small city size to the size of a small state. Although some counties have one administrative center (generally the county seat), many counties have a few centers of population, especially in the more populated regions of the United States. Below the county level are county subdivisions, which include civil divisions such as townships, towns, cities, and villages. Cities may further be subdivided into units such as wards or boroughs. Other units exist that are similar in size to county subdivisions including traffic analysis zone, voting districts, census-designated places, and unincorporated

places. Above the county-level are legislative districts, school districts, zip code tabulation areas, and urban growth areas.

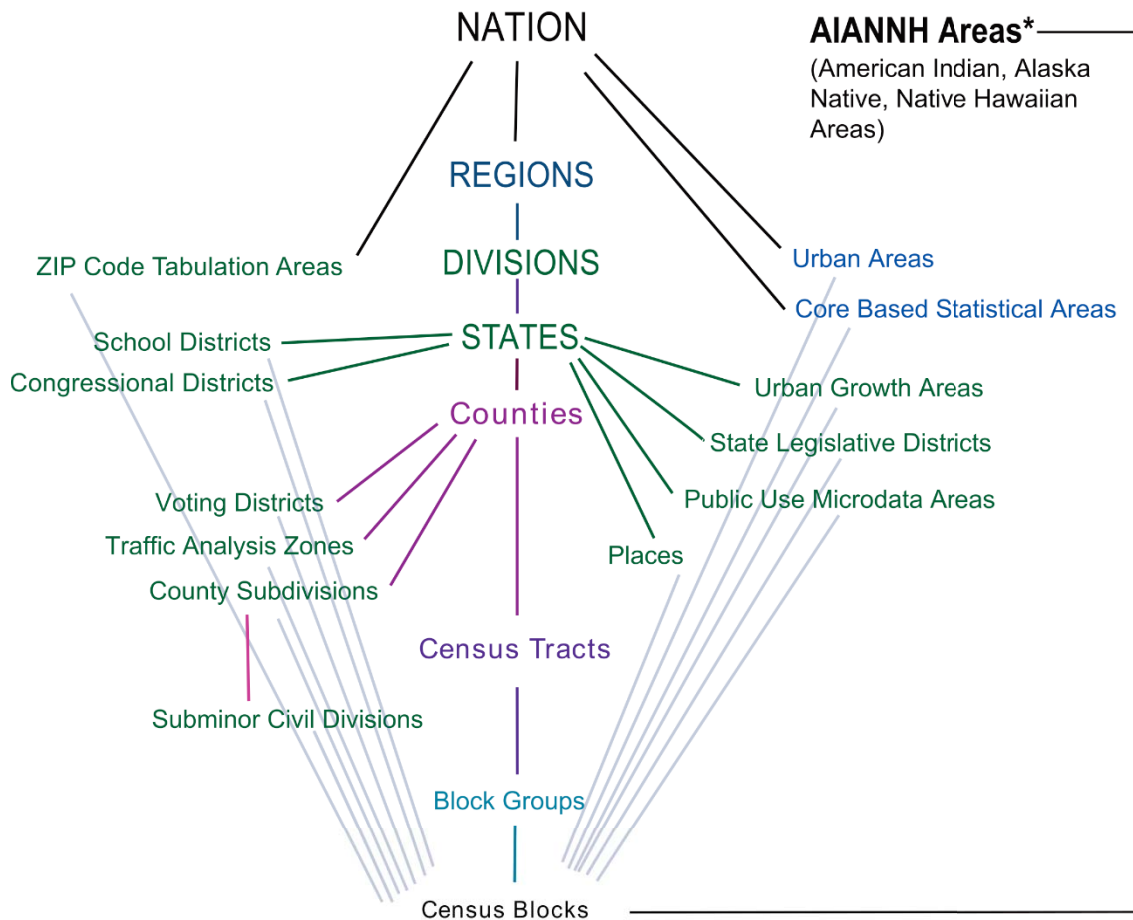


Figure 4.2: Standard hierarchy of census geographic entities

The Census Bureau defines urban areas, ranging from sub-county to multi-county in size, which include a central core city and the surrounding area around that city that is deemed to be economically tied to the city. The short definition, as outlined on the Census' website is [91]:

An urban area will comprise a densely settled core of census tracts and/or census blocks that meet minimum population density requirements, along with adjacent territory containing non-residential urban land uses as well as territory with low population density included to link outlying densely settled territory with the densely settled core. To qualify as an urban area, the territory identified according to criteria must encompass at least 2,500 people, at least 1,500 of which reside outside institutional group quarters. The Census Bureau identifies two types of urban areas:

Urbanized Areas (UZAs) of 50,000 or more people;
Urban Clusters (UCs) of at least 2,500 and less than 50,000 people.

The United States Office of Management and Budget (OMB) defines larger units called metropolitan areas. The smallest level of these are called Core Based Statistical Areas (CBSAs) which include Metropolitan Statistical Areas (MSAs) and Micropolitan Statistical Areas (μ SAs). Although the center cities are chosen using similar criteria as UAs, CBSAs are created with the county being the smallest unit, resulting in small μ SAs encompassing only one county to large MSAs encompassing five or more counties. Larger units still are Combined Statistical Areas (CSAs) which are groupings of adjacent MSAs and μ SAs.

The largest units below the state level were defined by the United States Bureau of Economic Analysis (BEA) in 2004, called Component Economic Areas (CEAs) and Economic Areas (EAs). EAs, the larger of the two, are comprised of 20+ counties that are considered to be economically tied to the central city more so than any other city. These units were discontinued in 2014.

As can be seen in this brief outline, many units could be used to define a geographical area, from very small (block level) to very large (EA level). Although one could choose any in this group as the area of study for sprawl measurement, most have chosen between the CBSA and the UA.

4.2.2 Analysis of Potential Geographies

To illustrate the various geographical units found in Table 4.1, a map for Grand Rapids, Michigan is shown in Figure 4.3. The extent at which these different areas extend from a city is vast, ranging from the just the city core to over a 50 mile radius around the city. Although there is significant reason to believe, and certainly the Bureau of Economic Analysis would have agreed when they developed Economic Areas [94], that economic ties certainly can extend this far from a city's core, it is clear from the other cities shown on the map that there are certainly other large urban centers in these areas that convolute the sprawl discussion. Indeed even the metropolitan area for Grand Rapids includes parts of the Holland and Muskegon UAs as well as 8 UCs.

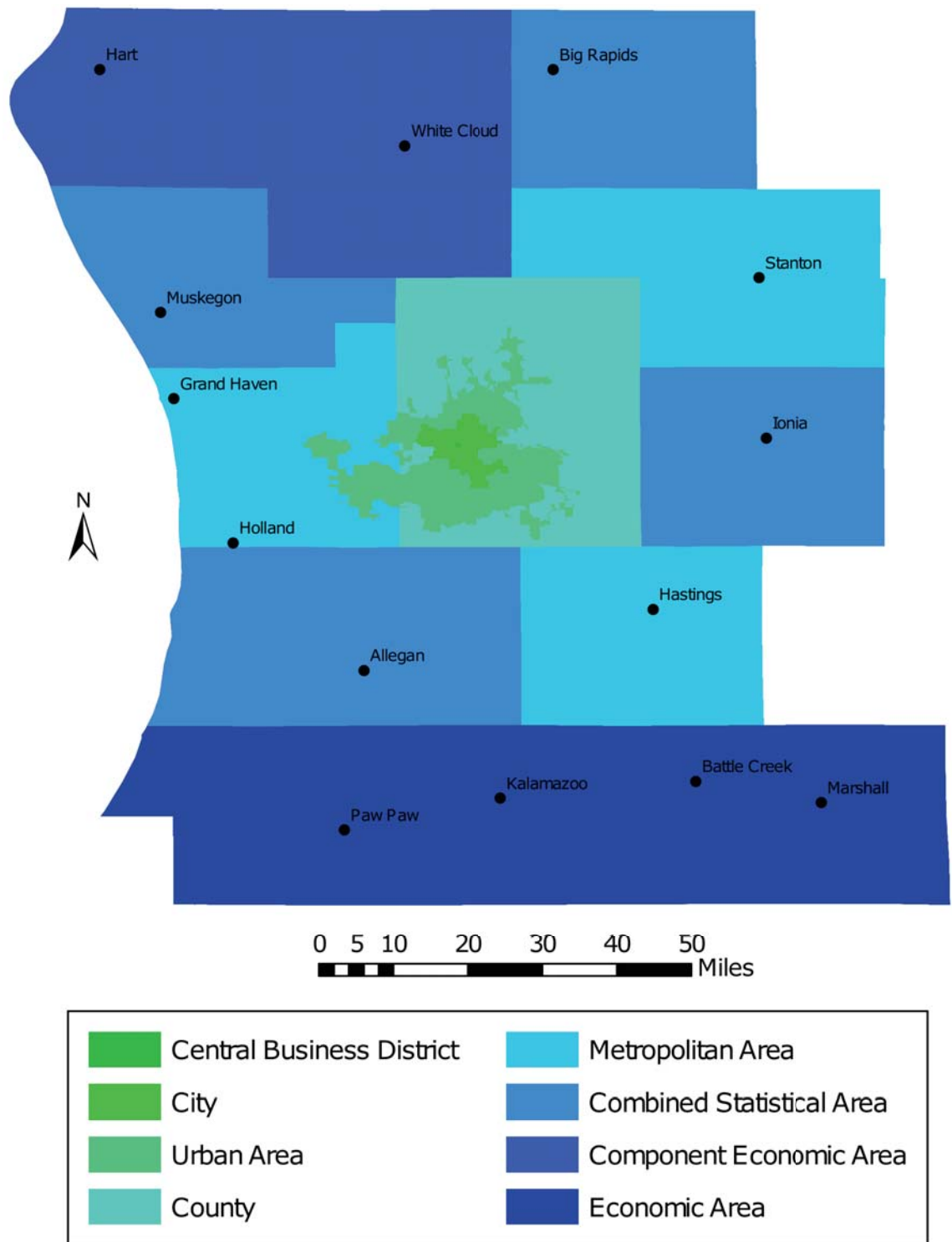


Figure 4.3: Geographic units and their various ranges for Grand Rapids, Michigan

The geographic areas are compared and reasons for elimination (from largest unit to smallest unit) are presented in Table 4.2. From this table, it can be seen why most researchers use either MSAs or UAs as their unit of measure for sprawl.

Table 4.2: Geographic areas in consideration for this study

Geographic Area	# in MI	Explanation for Elimination
State	1	Too many regionally important cities and many rural expanses between centers
Economic Area (EA)	8	Too many regionally important cities and many rural expanses between centers
Component Economic Areas (CEAs)	15	Some include too many regionally important cities with own characteristics
Combined Statistical Areas (CSAs)	7	Only focus on large cities and their suburbs
Metropolitan / Micropolitan Statistical Areas (MSA / μ SAs)	15/20	μ SAs are not a bad choice for core cities, but focus more on political boundaries at their limits rather than sociological boundaries
Urbanized Areas / Clusters (UZAs / UCs)	21/95	UCs are not a bad choice as they are established along economic/transportation boundaries, but some relevant external tracts are missing
Counties	83	Counties are political boundaries, which do not tie directly to socioeconomic boundaries
Census Subdivisions (Cities/Townships/CCDs)	1533	Many townships are socioeconomically tied to bordering or nearby cities; cities usually sprawl beyond their borders
Municipalities (Cities/Townships/Villages)	1789	Many townships and villages are socioeconomically tied to bordering or nearby cities
Census Tracts	2767	Many census tracts compose a city
Census Block Groups	8157	Many block groups compose in a city
Census Blocks	329610	Many blocks compose a city

Although most studies have utilized Core Based Statistical Areas, or more specifically the Metropolitan Statistical Areas (MSAs), these tend to outstrip areas of study for sprawl as some include multiple cores and most of these areas include rural regions that have little primary economic connection with the urban core being studied. These studies should not be criticized too much for using these geographical areas, as many data are available at this scope and thus would be a logical choice when trying to study many different regions. However, the Census actually has data available at many different geographies down to the block level, so one can actually focus down to that level of granularity to setup the geographical areas.

In contrast, some studies have used urban areas, specifically urbanized areas (UAs). These are smaller regions that cut out rural territory surrounding the core and often under-bound areas of study for sprawl. This leaves out many suburban development areas at the urban fringe and thus disregard some of the most fundamental locations where sprawl is very likely to occur.

4.2.3 Selection of the Appropriate Geography

For the reasons discussed in the previous section, Wolman et al. developed a scope metric in their important paper entitled “The fundamental challenge in measuring sprawl: Which land should be considered?” [69]. In their paper, two main reasons are cited for their skepticism towards many studies of sprawl. First, study areas are general either over or under-bounded. Although it could be argued from an idealist perspective that indeed no area will ever be perfectly bounded for a given region, let alone as a definition for a multi-area study, areas will certainly be more appropriate if chosen using rigor over mere convenience. Second, it is argued that most study areas do not eliminate land unavailable for development, which causes a distortion in density calculations and a bias between regions.

Although urban clusters are a good choice, Wolman et al.’s research into Extended Urban Areas (EUAs) seemed to be most appropriate. From this paper, the author’s states the definition for EUAs, as outlined in Section 5.1.1.

This definition was expanded in this research for urban clusters to form Extended Urban Clusters (EUCs) using a modified strategy of the extension of UAs to EUAs which involves the use of Rural-Urban Continuum Codes (RUCCs) [95].

The first course of action in creating EUC boundaries was to create an example map for one of the pilot cities using ArcGIS [89]. The Houghton EUC was used as the sample EUC and is shown in Figure 4.4, which includes municipal, UC, and the original potential EUC boundaries superimposed on a satellite image of the area. The satellite imagery is sourced from ESRI’s World Imagery dataset [96]. This rudimentary test algorithm was enhanced and after much mathematical analysis a satisfactory study scope methodology was developed that can be consistently applied across all pilot cities.

These geographic units (EUCs), based primarily on housing density and commuting patterns, formed the foundation for the study areas. The US Census Bureau has defined a very carefully reasoned standard with criteria for urban area boundary creation, and similar care will be put into developing these boundaries for this study [97].

After this step was completed, parameters for EUCs were refined and a test case was used for five cities chosen from different size categories and regions in the pilot group. These test cases were refined until satisfactory EUCs were achieved. At this point, the

method will be employed as an algorithm written in Python using ArcGIS to define EUC boundaries for all pilot cities. The final algorithm is outlined in Chapter 5.

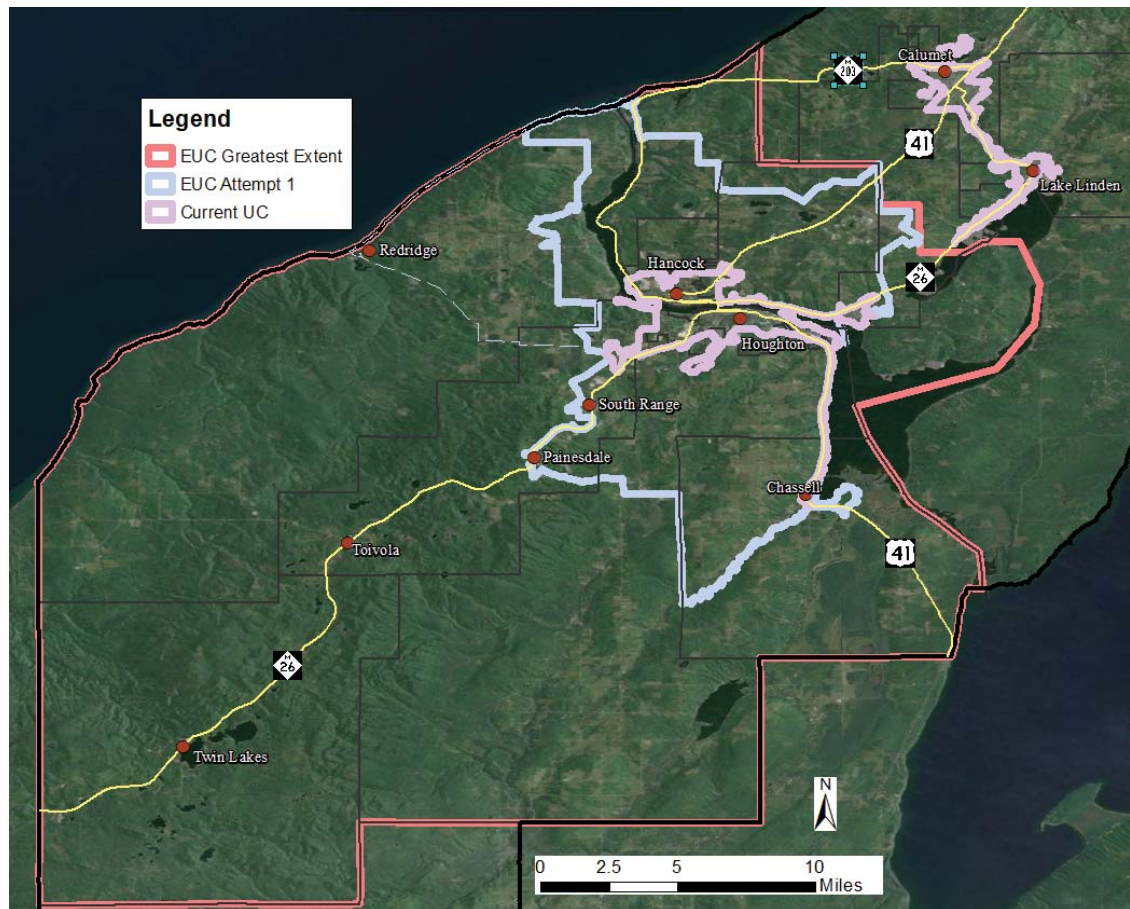


Figure 4.4: Graphic example of preliminary Houghton EUC

4.3 Metric and Sprawl Score Calculation

As discussed in Section 3.5, 21 metrics were selected and used to calculate an overall Sprawl Score. Data requirements for each metric were identified in Ewing et al.'s paper [3], but several other smaller datasets were needed to calculate all 21 metrics. Once all data sources were obtained, sample data analysis was completed on a small scale for the Houghton EUC to verify algorithms and operation. After this testing phase, the full algorithm for calculating each metric was developed and a Python script was written with two smaller support scripts. Details are discussed in Chapter 6.

Once the metrics were calculated for all 42 pilot cities, the metrics were evaluated both qualitatively and quantitatively using the statistical method called Principal Components Analysis (PCA) similar to that presented in Ewing et al. and Cutsinger et

al. [3,97,98]. Both groups used this method to reduce redundancy and hone in on fundamental metrics.

Using PCA, the metrics were summarized and an overall Sprawl Score was calculated for each of the 42 pilot cities. After these were calculated, an analysis was done on the rankings and conclusions were drawn from the overall rankings and component scores. The findings are discussed in Chapter 8.

Although this project focused on this primary group of metrics, it is recommended that this group be further analyzed to select a secondary group. This should be a cyclic process which will allow these groups to be analyzed and pared down until a fundamental grouping is determined. The goal of this would be to reduce the number of metrics to simplify the modeling of the sprawl system. A simpler system will make the measure more transparent and more usable. That being said, a sufficient amount of variables should be used to make sure that the model is robust and effective.

These metrics should also be applied to a study of causes and effects to gauge effectiveness. Future work opportunities based on the methodologies and algorithms developed in this research are discussed further in Chapter 8.

Chapter 5: Extended Urban Clusters

As discussed in Section 4.2, the geographical area used to calculate sprawl metrics has a direct effect on the quality of the metrics. The area chosen determines which parts of the region get counted towards the metric's score and serves as the area of influence for the community. The data sources and specific data used from the sources depends upon the unit of geographical area chosen.

Several researchers have used either Urban Areas (UAs) or Metropolitan Statistical Areas (MSAs) as units of geographical area. An appropriate area for this study was a hybrid of the two called the Extended Urban Cluster (EUC).

This chapter discusses the development of the EUCs which are used to determine the data for developing the metrics for the Sprawl Score. In many ways, this algorithm is the most important in the study. If a poor bounding size is chosen for the city, many metrics could be quite skewed resulting in an inaccurate set of Sprawl Scores. For this reason, considerable work was put into the creation of these EUCs.

The Chapter begins with a discussion of the development of the algorithm to build the EUC which is the geographic unit used in this study (Section 5.1).

Because of the complexity of the EUC, there is also a discussion of the algorithm used to create the geography. This is done at a high level in this section (Section 5.2), with an appropriate audience including geographers, planners, and other professionals with limited or no computer programming experience.

A low level discussion for advanced GIS users with programming experience is included in Appendix B along with the full Python script file. These explanations exist in order to leave no ambiguity in how the algorithms are developed so that researchers can use the algorithms and extend them to other regions as they deem appropriate.

To close the chapter, there is a discussion of the EUCs created for the pilot cities in this study (Section 5.3).

5.1 Development of the EUC Algorithm

This section will be a guide through the development of the structure of the EUC algorithm including understanding the algorithm for Extended Urban Areas (EUA) creation (Section 5.1.1) and the conversion to an EUC Algorithm (Section 5.1.2).

5.1.1 Extended Urban Areas

Wolman et al. developed EUAs to deal with the Goldilocks syndrome at hand between MSAs and UAs. EUAs seek to include land area in a consistent, logical fashion to better characterize the sphere of influence of the primary UA. As opposed to adding on

a common radius around UAs, EUAs systematically add on populated grids which have commuting ties to the urban core. The definition of a EUA is:

The Extended Urban Area (EUA) consists of the Census Bureau-defined urbanized area, modified to follow census tract boundaries, as well as additional “outlying” one mile square grid cells that contain 60 or more dwelling units (identified using data at the census block level) and are located in [a] census tract from which at least 30 percent of the workers commute to the urbanized area. [69]

The general algorithm for the process outlined above is shown in Figure 5.1.

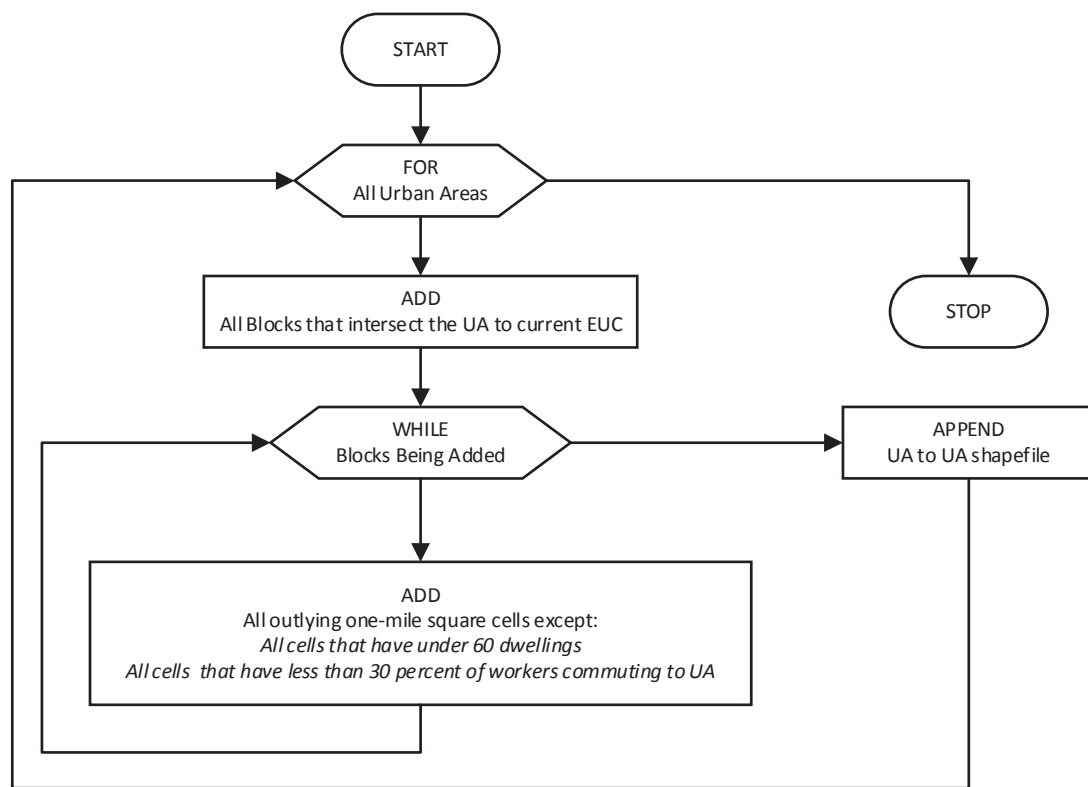


Figure 5.1: Basic flowchart for EUA shapefile Creation

5.1.1 Extending the EUA Algorithm to Extended Urban Clusters

The concept of EUAs works well for what it was designed for, extending urbanized areas to include all land that is tied to the core of a city without including rural areas beyond the fringes. However, since this research is focused on smaller urban area from 2,500 to 50,000 people, this concept cannot be translated directly as UAs are only defined for cities with populations of more than 50,000.

The focus cities for this study fall within the range of Census-defined urban clusters (UCs). These are extended in a similar procedure as UAs into elements defined here as Extended Urban Clusters (EUCs). For each urban cluster, the following steps are followed:

- all block groups (excluding water-only block groups) unique to the UC that contain no more than five times the area of the UC are selected
- outlying blocks unique to the UC which have at least 60 dwellings per square mile and 30 percent of workers commuting to the UC are selected
- Donuts holes, which are gaps in the EUC that are completely surrounded by the EUC, are filled in throughout such that the EUC is one continuous polygon

The basic algorithm for creating EUCs is shown in Figure 5.2.

5.2 Python Script – EUC_Creator.py

As this algorithm is complex, manual creation of even the 42 pilot cities would be time consuming and repetitive. For this reason, and for the sake of extending this algorithm in the future, a script was developed so that any EUC can be created following the same algorithm.

The benefits of a script include

- reduction of creation time and repetition
- elimination of human error in processing all of the complex selections
- consistent application of the algorithm
- ease of change to the algorithm
- ease of expansion to a broader study area

The script to create the EUCs was created using Python. An overview of the script's basic operation, intended for all audiences, is included in the following subsections. A detailed analysis of the script as well as the full Python script is included in Appendix B. The Python file is also included in Appendix H, the enclosed digital versatile disc (DVD) that is included in the media pocket on the inside back cover of this dissertation.

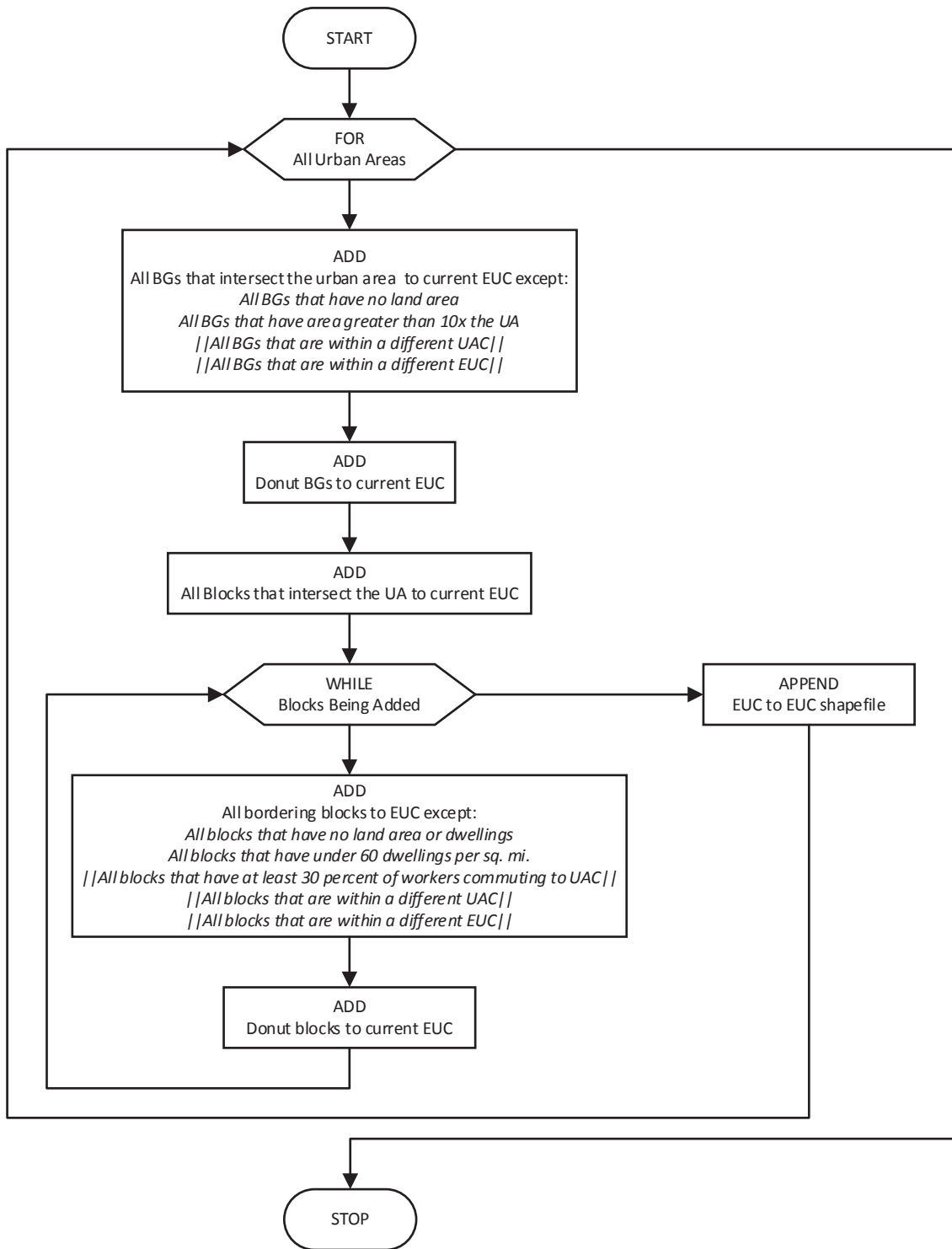


Figure 5.2: Basic flowchart for EUC shapefile creation

5.2.1 Files needed for EUC_Creator.py

In order to create the Extended Urban Cluster map, five files are needed. These are listed in Table 5.1. All files are standard files that are freely available to the public and updated with the decennial census. These files are all from the 2010 census year.

Table 5.1: Files needed to run EUC_Creator.py Python script

File Name	File Scope	Filename	File Developer
Block Groups	State	tl_2010_26_bg10.shp	US Census Bureau
Blocks	State	tl_2010_26_tabblock10.shp	US Census Bureau
Blocks	State	tabblock2010_26_pophu.shp	US Census Bureau
Urban Areas (UACs)	National	tl_2010_us_uac10.shp	US Census Bureau
Rural-Urban Commuting Area (RUCA) Codes	National	ruca_2010.xlsx	US Dept. of Agriculture

*Pre-joined with population and housing counts

The only file manipulations that are needed prior to the execution of the script include appending the population, housing, and density columns to the blocks file as well as appending UAC population counts to the UAC file. Other than these, the EUC creator script will run without any prior modification.

5.2.2 Overview of EUC_Creator.py

The Python script is relatively basic in its operation, although several support operations need to be completed for many of the basic steps which makes syntax and operation hard to follow. Although programmers will be able to follow with written comments throughout the script, those with only a basic knowledge of ArcGIS will benefit from the broader overview provided in this section.

The flowchart for the main operation of the script is shown in Figure 5.3. This script will actually create both Extended Urban Areas and Extended Urban Clusters using the modified version of the EUA algorithm developed by Wolman et al. The Wolman et al. approach for EUAs and the modified approach for EUCs used in this research are compared in Table 5.2.

Table 5.2: Comparison of Wolman et al.’s EUA and Riehl’s EUC algorithms

Extended Urban Area (EUA) – Wolman et al.	Extended Urban Cluster (EUC) – Riehl
Census Bureau defined Urbanized Area	Census Bureau defined Urbanized Area/Cluster
Modified to follow census tract boundaries	Modified to follow census block group boundaries
Adds additional “outlying” one-mile square grid cells that <ul style="list-style-type: none">• Contain 60 or more dwelling units• Are located in census tracts from which at least 30 percent of workers commute to urbanized area	Adds additional “outlying” blocks that <ul style="list-style-type: none">• Contain 60 or more dwellings per square mile• Are located in census tracts from which at least 30 percent of workers commute to urbanized area/cluster

As seen in Table 5.2, the algorithms are very similar. The EUC algorithm is meant to mimic the EUA algorithm when possible, and remain as close as possible to the EUA algorithm. More advanced functions of the algorithm address some minor flaws in the EUA algorithm. The EUC algorithm will be explained at a high level for non-programmers in the following sections.

5.2.2.1 Selection of UA area and Overlapping Block Groups

The EUC algorithm runs for each Urban Area in Michigan, including all urbanized areas and clusters, even those not included in the list of 42 pilot cities in this study. This was done in order to eliminate overlap between EUCs as well as to help troubleshoot and verify the algorithm to eliminate all minor flaws in the creation of the EUCs. For the purpose of this report, all of the elements created by this script will be referred to as EUCs, even though extensions of urbanized areas are also included in the file.

For each urban area, block groups are selected which intersect the urban area to create a base set for the EUC. In order to eliminate water-only block groups which have no (or more correctly very little) potential for human expansion, block groups with no land area are removed from this set.

In less populated areas of Michigan, for example most of the Upper Peninsula, block groups tend to get very large. On original runs of the algorithm, cities such as Manistique had boundaries extending up to Munising simply because of a small overlap of the UC and a block group that extends from Lake Michigan to Lake Superior. To eliminate this problem, block groups greater than five times the size of the original UA were eliminated from the EUC at this stage. Theoretically they might be added in block by block at a later stage in the program based on a high enough population, but this has a very small probability being the block group is large due to low population. To derive 5 as the size limit, many trial iterations of different cutoff sizes between 1 and 20 times resulted in the selection of five as a good standard with the most reasonable results.

Next, all block groups that were within another UA or EUC were eliminated to remove overlap potential. Because the loop runs from the largest UA population to the smallest, larger cities get precedence over block group inclusion, which is the most reasonable option as overlap does not make sense. The elimination of blocks in this manner involves two more loops, one through each of the UAs and one through each of the completed EUCs. These algorithms are shown in Figures 5.3 and 5.4, respectively.

In order to remove block groups that overlap with another UA, a loop through each UA is executed. As long as the UA is not the same as the current one being used to create the EUC, all intersecting block groups are removed from the current EUC. This process is shown in Figure 5.3.

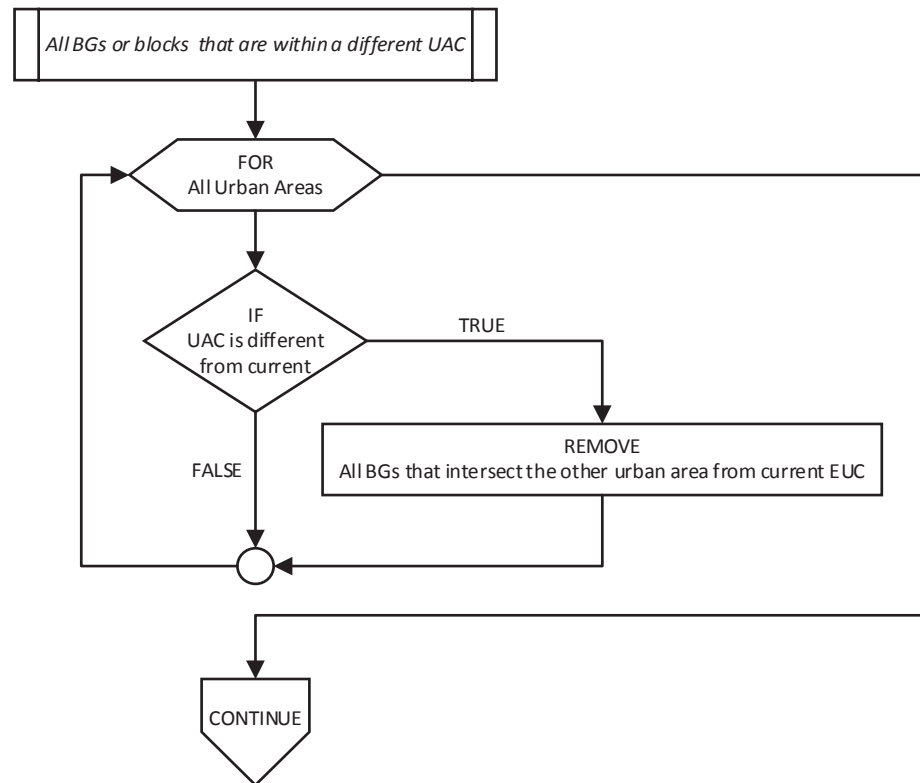


Figure 5.3: Flowchart for UAC overlapping block group / block removal

Similarly, as long as an EUC exists, a loop is run through each EUC and all intersecting block groups are removed from the current EUC. This process is shown in Figure 5.4.

Through many trials with the software, another flaw was determined in that block groups were often selected as part of the EUC that fully surrounded other block groups that were not selected. To eliminate this problem, a process is run to add in these so-called “donut” block groups to the EUC to form a continuous polygon. This process is explained in more detail in Appendix B.

The final process before selecting outlying block groups was to add in blocks from block groups that were eliminated due to a size constraint. To do this, all blocks that intersect the original UA that were not yet selected were then added into the EUC. Because blocks are the fundamental building block of UAs, this results in full coverage of the UA.

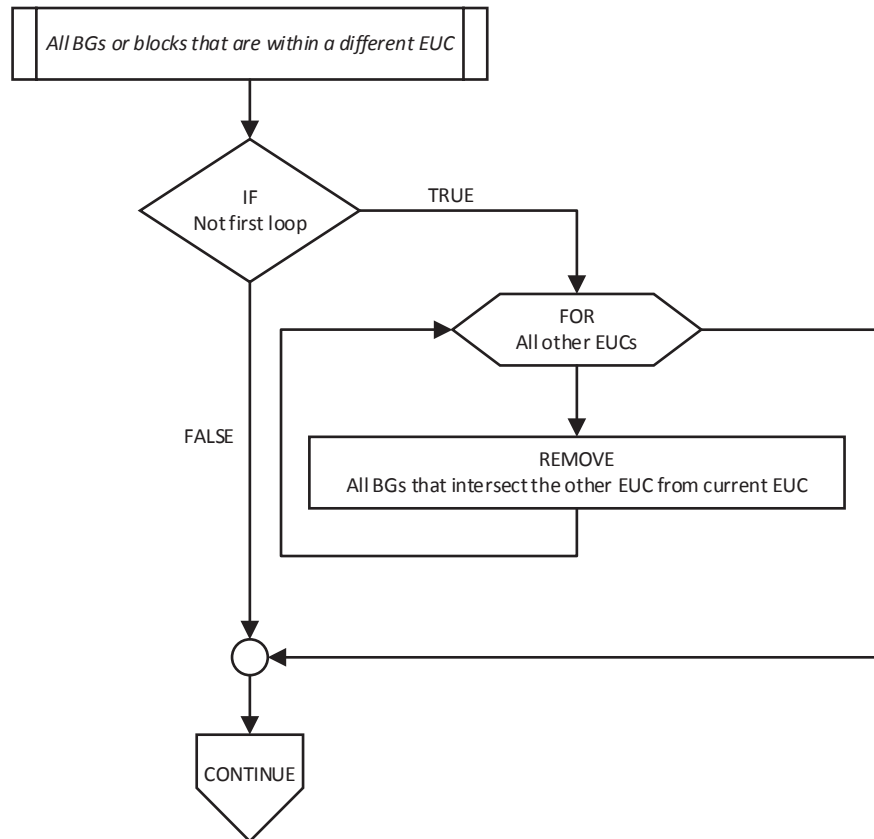


Figure 5.4: Flowchart for EUC overlapping block group / block removal

5.2.2.2 Selection of Outlying Blocks

Once the initial boundaries of the EUC are identified, outlying blocks of significance must be added to finalize the boundaries of the EUC. To do this, a loop is run in order to add bordering blocks until no more bordering blocks meet the addition criteria.

With each run through this loop, all bordering blocks are selected for potential addition to the EUC. All of these blocks are then processed to determine if they are fit for addition. First, blocks with no land area or dwellings are removed from the selection. This is followed by removal of blocks that have under 60 dwellings per square mile.

Next, all blocks that do not have at least 30 percent of workers commuting to the EUC must be removed. To determine this value, the Rural-Urban Commuting Area (RUCA) Codes, maintained by the US Department of Agriculture, are used [99,100]. RUCA codes are calculated for each census tract in the country. These codes define the number of commuters between a census tract and the city center. Primary and secondary codes are set up with secondary codes offering the level of granularity needed in this study. The codes are shown in Table 5.3 below. These codes are used in urban planning and traffic engineering to study flows of people and traffic.

Table 5.3: Table of RUCA codes [99]

Primary RUCA Codes		Secondary RUCA Codes	
1	Metropolitan area core: primary flow within an urbanized area (UZA)	1.0	No additional code
		1.1	Secondary flow 30% to 50% to a larger UA
2	Metropolitan area high commuting: primary flow 30% or more to a UZA	2.0	No additional code
		2.1	Secondary flow 30% to 50% to a larger UA
3	Metropolitan area low commuting: primary flow 10% to 30% to a UZA	3.0	No additional code
4	Micropolitan area core: primary flow within an Urban Cluster of 10,000 to 49,999 (large UC)	4.0	No additional code
		4.1	Secondary flow 30% to 50% to a UA
5	Micropolitan high commuting: primary flow 30% or more to a large UC	5.0	No additional code
		5.1	Secondary flow 30% to 50% to a UA
6	Micropolitan low commuting: primary flow 10% to 30% to a large UC	6.0	No additional code
7	Small town core: primary flow within an Urban Cluster of 2,500 to 9,999 (small UC)	7.0	No additional code
		7.1	Secondary flow 30% to 50% to a UA
		7.2	Secondary flow 30% to 50% to a large UC
8	Small town high commuting: primary flow 30% or more to a small UC	8.0	No additional code
		8.1	Secondary flow 30% to 50% to a UA
		8.2	Secondary flow 30% to 50% to a large UC
9	Small town low commuting: primary flow 10% to 30% to a small UC	9.0	No additional code
		10.0	No additional code
10	Rural areas: primary flow to a tract outside a UZA or UC	10.1	Secondary flow 30% to 50% to a UA
		10.2	Secondary flow 30% to 50% to a large UC
		10.3	Secondary flow 30% to 50% to a small UC
99	Not coded: Census tract has zero population and no rural-urban identifier information		

Depending on which type of UAC is the basis for the current EUC, certain codes are deemed acceptable based on whether or not at least 30% or more of commuters in the block in question commute to the UAC. Although UACs include only two types, urban areas and urban clusters, RUCA codes include three types, urbanized areas (>50,000 population), large urban clusters (10,000 – 50,000 population), and small urban clusters (<10,000 population). The following codes were selected as acceptable for each of these three types:

- Urbanized Areas: 1.0, 1.1, 2.0, 2.1, 4.1, 5.1, 7.1, 8.1, 10.1
- Large Urban Clusters: 4.0, 5.0, 7.2, 8.2, 10.2
- Small Urban Clusters: 7.0, 8.0, 10.3

Note that “acceptable” means that a tract with the listed code is allowed for inclusion into the EUC whereas other codes are not allowed. These codes are considered acceptable because they mean that at least 30% of commuters commute to the UAC in question.

This portion of the algorithm requires additional steps as shown in Figure 5.5. This algorithm is discussed in more detail in Appendix B.

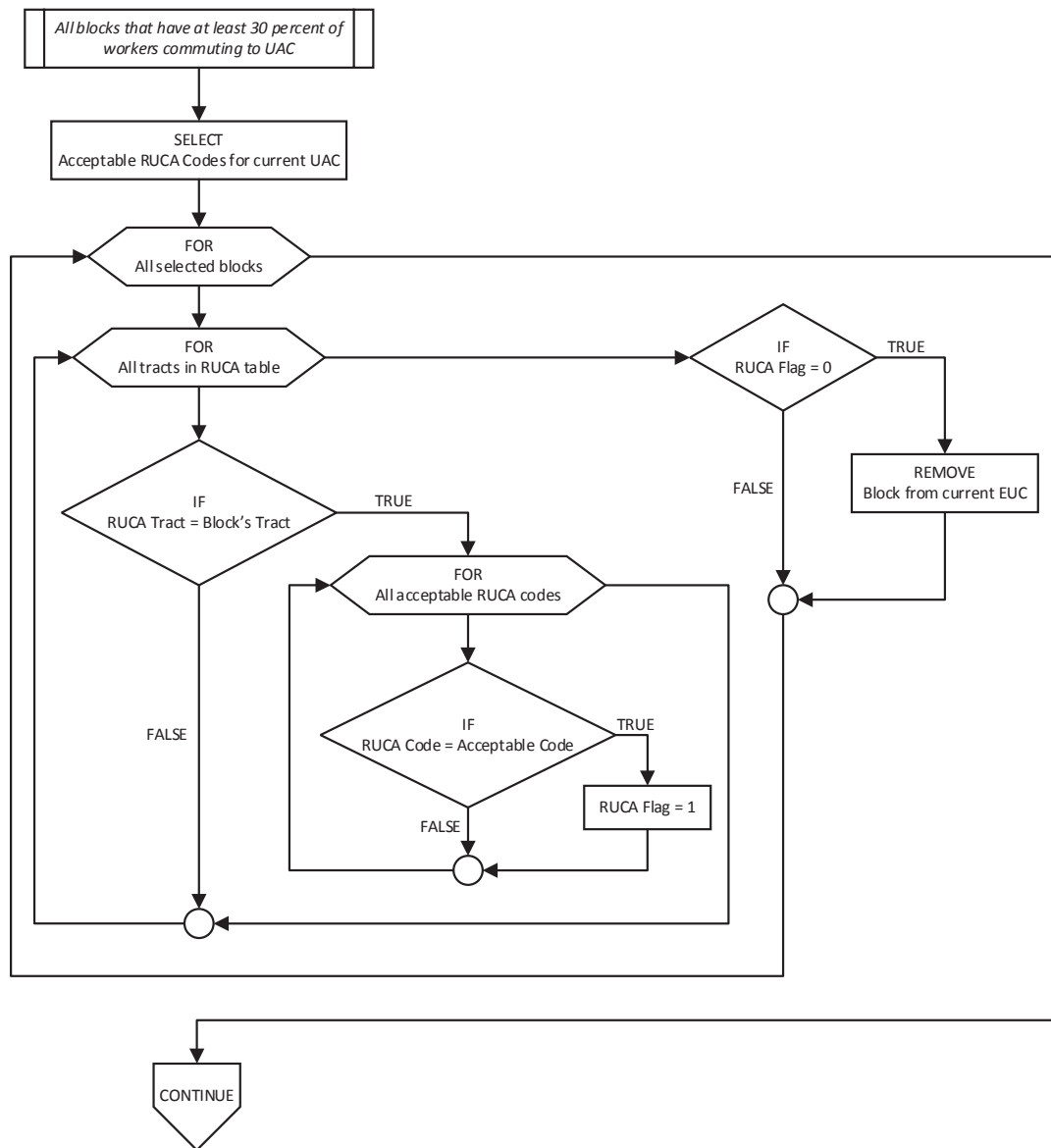


Figure 5.5: Flowchart for removal of blocks with under 30% commuters to UAC

5.3 Final EUC Boundaries

After executing `EUC_Creator.py`, as described in Section 5.2, a final EUC boundary file was created. The file created includes all 116 UACs that have land in the state of Michigan. The EUCs are shown in Figure 5.6.

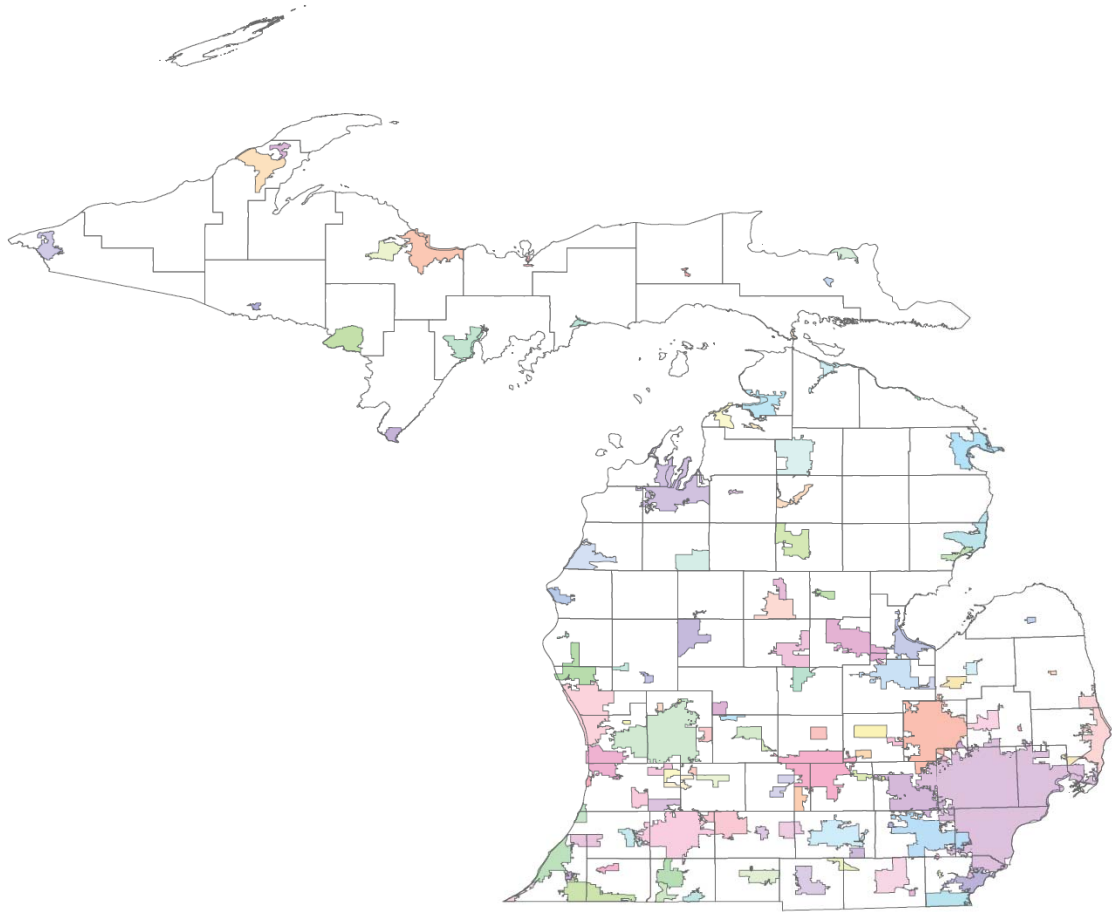


Figure 5.6: Map of 116 EUCs in Michigan (created by `EUC_Creator.py`)

A simple selection of the 42 pilot cities created the final pilot city EUC file. This file is used in Chapter 6 to develop the sprawl score for these EUCs. The final map of 42 pilot EUCs is shown in Figure 5.7.

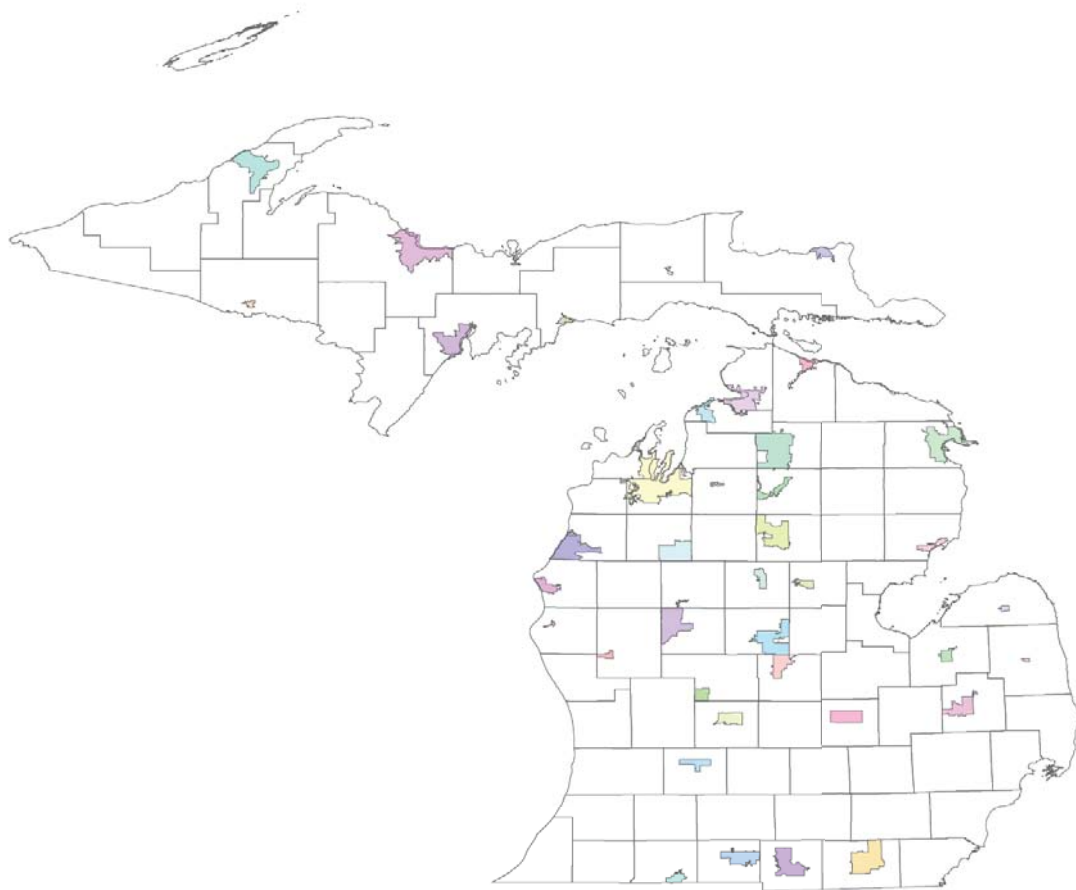


Figure 5.7: Map of 42 pilot EUCs in Michigan

As an example to compare the resulting EUCs to original estimates, the Houghton EUC, as shown in Figure 5.8, was compared to Figure 4.4. The Houghton comparison is necessary as the original attempt at rough EUC areas prior to algorithm creation were done using Houghton. The satellite imagery in Figure 5.8 is sourced from ESRI's World Imagery dataset [96]

What is interesting is that the EUC Attempt 1 polygon with the dotted Redridge addition, outlined in light blue on the figure, is very similar to the final EUC polygon. Perhaps the most striking difference is the exclusion of South Range and Painesdale in the final EUC, as well as some of the land area of the Houghton County Memorial Airport.

Some would argue that South Range should be included in the EUC as there is near continuous development between downtown Houghton and South Range along the M-26 corridor. This could a topic for future discussion of EUC boundaries.

An adjustment to the algorithm was necessary as there will be with many cities especially in Lower Michigan where interference between EUCs is an issue. For instance, the Houghton EUC is adjacent to the Laurium EUC to the north. Other than these small issues, the area is quite inclusive of the commute-shed and overall expanse of the city. Although some might argue this area is too large, at least it is nowhere near as large as the Houghton Micropolitan Statistical Area which includes all of Houghton and Keweenaw Counties.

The problem with any area selection is that all areas are subjective, but as long as an area is close to what most people would consider the urban area plus important outlying areas, the algorithm is a success. Considerable effort went into developing this algorithm and eliminating inconsistencies. That being said, future work should focus first on an update to this algorithm to improve EUC selection.

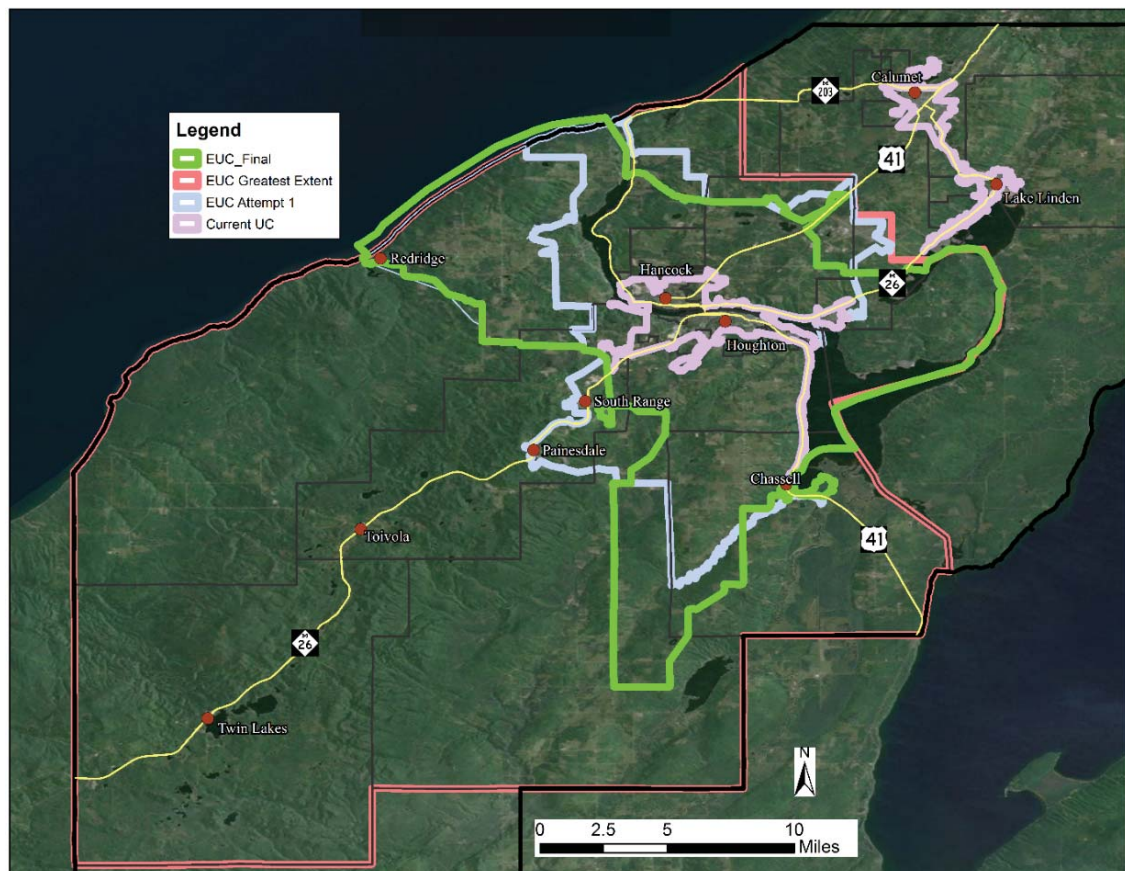


Figure 5.8: Graphic example of Houghton EUC, including final Houghton EUC

Chapter 6: Sprawl Metric Assessment

Following the extensive review of literature, as described in Chapter 3, it was decided to use the 21 metrics from Part 3 of Ewing et al.'s updated paper measuring urban sprawl for large metropolitan areas [3].

Section 6.1 outlines the benefits of using these metrics and describes each metric in such detail that anyone should be able to understand how each metric is derived. Understanding this is important, because it plays into not only how the score is calculated, but how the score is interpreted.

Section 6.2 discusses the main python script in terms that should be understandable to those with limited programming experience. Section 6.3 describes support scripts needed to complete metric calculation.

As this script was not as complex as the EUC creator script, there is no programmer-specific information other than that information included in the comments of the program.

The full program for the main EUC metrics script is included in Appendix C. The support script programs are included for the CBD polygon creator in Appendix D and for the Walk Score value locator in Appendix E. The three Python files are also included in Appendix H, the enclosed DVD that is included in the media pocket on the inside back cover of this dissertation.

6.1 Selected Metrics

Using Ewing et al.'s 2014 metrics as primary metrics is beneficial for several reasons. First, these metrics are updated based on research from 2002 to 2014 by comparing many of the metrics discussed in Chapter 3 of this dissertation and selecting the optimal set. Second, although these metrics have been used on large cities, this research is the first to utilize these metrics in small cities. This offers a way to compare metrics at different scales and is a good starting point to reduce and identify the most important metrics.

A variety of these metrics were used on various city boundaries by Ewing et al., including counties, UAs, and MSAs. The MSA metrics were chosen because they are the most diverse metrics. Some of the metrics were modified slightly to account for the difference in size between MSAs and EUCs as well as the difference between larger and smaller cities.

Table 6.1 reviews the list of 21 metrics used in this study divided into 4 factor categories: development density, land use mix, activity centering, and street accessibility. Each of the metrics used will be discussed by factor category in Sections

6.1.1-6.1.4 below. This discussion includes an overview of the category and a detailed discussion for each metric including the algorithm used to calculate the metric. Once all metric algorithms were determined and all data were collected, a python script (*EUC_Metrics.py*) was created to process all of the metrics for all EUCs (Section 6.2).

Table 6.1: Table of 21 metrics used in Sprawl Score calculation

Factor Category	Metric	Short Name
Development Density	Population density	EWpopden
	Employment density	EWempden
	Percentage of the population living at suburban densities	EWlt1500
	Percentage of the population living at urban densities	EWgt12500
	Net population density of urban lands	EWurbden
	Estimated density at the center of the metro area	EWdgcent
	Population density of CBD	EWpopdcen
	Employment density of CBD	EWempdcen
Land Use Mix	Job / population balance	EWjobpop
	Degree of job mixing	EWjobmix
	Walk Score	EWwalkscor
Activity Centering	Coefficient of variation in block population densities	EWvarpop
	Coefficient of variation in block employment densities	EWvaremp
	Density gradient moving outward from the CBD	EWdgrad
	Percentage of population in the CBD	EWpopcen
	Percentage of employment in the CBD	EWempcen
Street Accessibility	Percentage of small urban blocks	EWsmlblk
	Average block size	EWavgbklsz
	Average block length	EWavgbklkn
	Intersection density	EWintden
	Percentage of 4-or-more-way intersection	EW4way

6.1.1 Development Density Metrics

Since most definitions of sprawl include development density it is necessary to include density in the Sprawl Score. Of major interest is density of population (residential) and density of employment (commercial). The problem is that overall density within the region only somewhat characterizes the development in the region. Density in the central business district (CBD) can vary when compared to the overall regional density, so it must also be included. These ideas help make up eight metrics which combine to measure development density for the EUC. Each is discussed below.

6.1.1.1 Population density

Description: Overall population density measures the overall amount of persons per unit area across the entire EUC.

Units: persons per square mile

Relationship: Higher values equate to less sprawl

Datasets Used:

Census 2010 block data including demographic data (for populations and areas)

Algorithm: The populations and areas for all census blocks in the EUC were added together and the total population was divided by the total area.

Differences between this and Ewing et al. metric: Blocks were used instead of tracts. Also, all blocks were used, as opposed to cutting out blocks with very low populations or areas.

6.1.1.2 Employment density

Description: Overall employment density measures the overall amount of jobs per unit area across the entire EUC.

Units: jobs per square mile

Relationship: Higher values equate to less sprawl

Datasets Used:

Census 2010 block data including demographic data (for areas)

Longitudinal Employer Household Dynamics (LEHD) Origin-Destination

Employment Statistics version 7 (LODES7) Workplace Area

Characteristics (WAC) 2010 (for jobs) [101]

Algorithm: First, the job data from the LED LODES7 WAC 2010 data file for total number of primary jobs in the block was merged with the block file. Then the job totals and areas for all census blocks in the EUC were added together and the total number of jobs was divided by the total area.

Differences between this and Ewing et al. metric: Blocks were used instead of tracts. Also, all blocks were used, as opposed to cutting out blocks with very low populations or areas.

6.1.1.3 Percentage of the population living at suburban densities

Description: This metric essentially is measuring how many p are living in sprawling parts of the EUC. To do this, the percentage of persons living at population densities less than 1,500 people per square mile is used. Higher population density is more compact. One downside to this metric is that it does not eliminate rural living from the calculation. Another downside is that results are not normalized by overall population in the EUC.

Units: percentage (number of blocks at low density over total number of blocks)

Relationship: Higher values equate to more sprawl

Datasets Used:

Census 2010 block data including demographic data (for populations and areas)

Algorithm: The population of each census block in the EUC was divided by the block's area to calculate the population density of each block. The block was counted if its density was below 1,500 people per square mile. After processing this count for all blocks in the EUC, this count was divided by the total number of blocks in the EUC.

Differences between this and Ewing et al. metric: Blocks were used instead of tracts. Also, all blocks were used, as opposed to cutting out blocks with very low populations or areas.

6.1.1.4 Percentage of the population living at urban densities

Description: This metric essentially is measuring how many people are living in compact parts of the EUC. To do this, the percentage of people living at population densities greater than 12,500 people per square mile is used. One downside to this metric is that results are not normalized by overall population in the EUC. In some smaller communities that do not have enough population to build up, these densities are not very common.

Units: percentage (number of blocks at high density over total number of blocks)

Relationship: Higher values equate to less sprawl

Datasets Used:

Census 2010 block data including demographic data (for populations and areas)

Algorithm: The population of each census block in the EUC was divided by the block's area to calculate the population density of each block. The block was counted if its

density was above 12,500 people per square mile. After processing this count for all blocks in the EUC, this count was divided by the total number of blocks in the EUC.

Differences between this and Ewing et al. metric: Blocks were used instead of tracts. Also, all blocks were used, as opposed to cutting out blocks with very low populations or areas.

6.1.1.5 Net amount of urban lands

Description: This metric gives the percentage of area in the EUC that is developed land. This includes classes 21-24 of the NLCD dataset (Developed, Open Space; Developed, Low Intensity; Developed, Medium Intensity; and Developed, High Intensity).

Units: percentage (amount of urban land over total land)

Relationship: Higher values equate to less sprawl

Datasets Used:

Census 2010 block data including demographic data (for populations and areas)
National Land Cover Database (NLCD) 2011 [102]

Algorithm: The majority of this algorithm is completed before the metric script is executed, and this algorithm is discussed in Section 6.2.1. Essentially, the NLCD dataset had all urban lands selected and summed. This total was then divided by the total land in the EUC to give a percentage of urban lands.

Differences between this and Ewing et al. metric: Note that this metric is not consistent with Ewing et al.'s metric because this is a percentage of urban lands instead of a population density of urban lands. The algorithm should be updated to sum the populations over the urban areas selected and this total should be divided by the total of urban lands. This would be a slightly more normalized metric across different sized regions. It is unknown which categories of land Ewing et al. used for urban land, but it is reasonable to believe the same categories were used.

6.1.1.6 Estimated density at the center of the metro area

Description: This metric is an estimate of the density at the center of the EUC based upon how quickly the population exponentially diminishes from the center of the EUC's CBD outward. This metric and the density gradient (discussed in Section 6.1.3.3) work in tandem. The equation coefficients are calculated using population density of each block and distance of the center of each block to the center of the EUC's CBD.

Units: persons per square mile

Relationship: Higher values equate to less sprawl

Datasets Used:

CBD Points (created manually by the author)

Census 2010 block data including demographic data (for populations, areas, latitudes, and longitudes)

Algorithm: First, a CBD points file must be generated with points denoting the center of the CBD of the primary city in the EUC. A discussion of this file's creation is included in Section 6.3.1. The first step in the algorithm was to calculate the difference in latitude and longitude between the CBD's center point and each block's center point (using the Census' internal point, which is the geographical center except for special cases where the point would fall outside the block's boundaries in which case the point inside the block closest to the center point is chosen [103]). Because these distances are around a geoid (the Earth) and not linear, the Haversine formula, recommended by the US Census Bureau to calculate short distances equations, was used to calculate each distance [104]:

$$d = 2 \cdot R \cdot \tan^{-1} \left(\sqrt{\sin^2 \frac{\phi_d}{2} + \cos \phi_b \cdot \cos \phi_e \cdot \sin^2 \frac{\lambda_d}{2}} \right)$$

where d is the distance between block and EUC's CBD center points, in mi.,
 R is the radius of the Earth, in m.,
 $\phi_d = \phi_b - \phi_e$,
 ϕ_b is the latitude of the block center point, in decimal degrees,
 ϕ_e is the latitude of the EUC's CBD center point, in decimal degrees,
 $\lambda_d = \lambda_b - \lambda_e$,
 λ_b is the longitude of the block center point, in decimal degrees,
 λ_e is the longitude of the EUC's CBD center point, in decimal degrees

The population density for each block was also stored. After all distances and densities were calculated and stored, a polynomial fitting function was used to derive the coefficients of the following exponential function using the natural log of each variable:

$$D = D_0 \cdot e^{-b \cdot d}$$

where D is the density of the block in persons per sq. mi.,
 D_0 is the estimated density at the EUC center in persons per sq. mi.,
 b is estimated density gradient moving outward from CBD, in mi^{-1} ,
 d is the distance between block and EUC's CBD center points, in mi.

The value of D_0 is the metric. The value of b is used as a centering metric, discussed in Section 6.1.3.3.

Differences between this and Ewing et al. metric: Blocks were used instead of tracts. CBD center points were manually calculated to determine more accurate locations. It is unknown how Ewing et al. calculated distances between points, but they likely used a similar method and even if not, these relatively short distances will have little error.

6.1.1.7 Population Density of CBD

Description: This metric is the weighted average population density of the primary CBD in the EUC. This is essentially the average population density of all blocks within the CBD area, except each density is weighted by the blocks influence within the EUC, as discussed in the algorithm below.

Units: persons per square mile

Relationship: Higher values equate to less sprawl

Datasets Used:

- CBD Polygons (created using the program describe in 6.3.2)
- Census 2010 block data including demographic data (for populations and areas)
- Longitudinal Employer Household Dynamics (LEHD) Origin-Destination Employment Statistics version 7 (LODES7) Workplace Area Characteristics (WAC) 2010 (for jobs)

Algorithm: First, a CBD polygon file must be generated for all EUC's primary cities. A discussion of this file's creation is included in Section 6.3.2. Each block in the CBD area had its population weighted by multiplying it by the sum of its population and employment then dividing it by the sum of the population and employment for the entire EUC. This resulted in a population weighted by the block's influence within the EUC. The weighted populations were then summed along with the areas for each block in the CBD and the total weighted population was divided by the total area resulting in a weighted population density.

Differences between this and Ewing et al. metric: There was some difference in how CBDs were created, which is discussed in detail in Section 6.3.2. Also, Ewing et al.'s metric included other employment sub-centers besides the CBD. These were not used in this study because smaller urban areas have fewer or no sub-centers and more importantly sub-centers can be considered sprawl in their own right and this study is designed to eliminate these from skewing results.

6.1.1.8 Employment Density of CBD

Description: This metric is the weighted average employment density of the primary CBD in the EUC. This is essentially the average employment density of all blocks within the CBD area, except each density is weighted by the blocks influence within the EUC, as discussed in the algorithm below.

Units: jobs per square mile

Relationship: Higher values equate to less sprawl

Datasets Used:

- CBD Polygons (created using the program describe in 6.3.2)
- Census 2010 block data including demographic data (for populations and areas)
- Longitudinal Employer Household Dynamics (LEHD) Origin-Destination Employment Statistics version 7 (LODES7) Workplace Area Characteristics (WAC) 2010 (for jobs)

Algorithm: First, a CBD polygon file must be generated for all EUC's primary cities. A discussion of this file's creation is included in Section 6.3.2. Each block in the CBD area had its number of jobs weighted by multiplying it by the sum of its population and employment then dividing it by the sum of the population and employment for the entire EUC. This resulted in a job total weighted by the block's influence within the EUC. The weighted employments were then summed along with the areas for each block in the CBD and the total weighted employment was divided by the total area resulting in a weighted employment density.

Differences between this and Ewing et al. metric: There was some difference in how CBDs were created, which is discussed in detail in Section 6.3.2. Also, Ewing et al.'s metric included other employment sub-centers besides the CBD. These were not used in this study because smaller urban areas have fewer or no sub-centers and more importantly sub-centers can be considered sprawl in their own right and this study is designed to eliminate these from skewing results.

6.1.2 Land Use Mix Metrics

One of the tenants of sprawl development beyond merely low density is the separation of land use types from each other. Although this was originally done to eliminate the pesky commercial and industrial problems from everyday life, this type of development forces auto dependence and wastes time for users of the systems. One of the major mixing ideas is mixing residences with businesses. Also of interest is the mixing of different job types. Lastly there are many other factors in land use mixing including mixing all types of commercial, civic, and recreational opportunities within walking distance of residences. Walk Score was developed using many metrics of its own to

assess walkability for residences. These three areas make up the metrics which combine to measure land use mix for the EUC. Each of these three is discussed below.

6.1.2.1 Job / population balance

Description: This metric looks at the number of jobs as they are spread out with population across the EUC. If the ratio of jobs to persons in a block is the same as the EUC's ratio, this is considered to be a better balance. If only jobs or persons exist in a block, this is considered to be worse balance.

Units: unitless

Relationship: Higher values equate to less sprawl

Datasets Used:

Census 2010 block data including demographic data (for populations)
Longitudinal Employer Household Dynamics (LEHD) Origin-Destination
Employment Statistics version 7 (LODES7) Workplace Area
Characteristics (WAC) 2010 (for jobs)

Algorithm: To calculate job-population balance, the jobs in a block compared to the average jobs in the EUC were compared and then weighted by the job population total in the block to the job population total in the EUC. This is summarized by the following equation which was used for the calculation, from Ewing et al.:

$$jpb = \sum_{i=0}^n \left(1 - \frac{|J_i - JP \cdot P_i|}{J_i + JP \cdot P_i} \cdot \frac{J_i + P_i}{J_T + P_T} \right)$$

where jpb is job / population balance,
i is the current census block entry,
n is the total number of census blocks,
J_i is the number of jobs in the census block,
P_i is the number of persons in the census block,
JP is the jobs per person total in the EUC,
J_T is the total number of jobs in the EUC,
P_T is the total number of persons in the EUC

Blocks with population densities under 100 were not included in this total.

Differences between this and Ewing et al. metric: The main differences between the two metrics is that this study used census blocks instead of census tracts and instead of using 1-mile rings to narrow block groups, blocks were used in their entirety.

6.1.2.2 Degree of Job Mixing

Description: This metric looks at the number of different types of jobs as they are mixed across the EUC. If the number of jobs in each sector is the same for a block, this is considered to be a better mixing. If only jobs in a single sector exist in a block, this is considered to be worse mixing as there is not diversity in job types.

Units: unitless

Relationship: Higher values equate to less sprawl

Datasets Used:

Census 2010 block data including demographic data (for populations)
Longitudinal Employer Household Dynamics (LEHD) Origin-Destination
Employment Statistics version 7 (LODES7) Workplace Area
Characteristics (WAC) 2010 (for jobs and jobs by sector)

Algorithm: Degree of job mixing was calculated using an entropy formula as put forth by Ewing et al. Number of jobs in each sector are weighted by the natural log of jobs in the sector over the natural log of the number of sectors as well as by the job population total in the block to the job population total in the EUC. The degree of job mixing for each sector is added and the result is the overall degree of job mixing for the EUC. The five sectors used in the calculation were retail, entertainment, health services, education, and personal services which includes finance, real estate, and information, tech, hotel, food, and other services. All of these jobs values were gathered from the LEHD LODES7 WAC 2010 dataset. The equation for one sector is shown below:

$$m = \sum_{i=0}^n \left(\frac{J_S \cdot \ln J_S}{\ln j} \cdot \frac{J_i + P_i}{J_T + P_T} \right)$$

where m is degree of job mixing for the current sector,
 i is the current census block entry,
 n is the total number of census blocks,
 J_S is the number of jobs in the sector,
 j is the number of sectors,
 J_i is the number of jobs in the census block,
 P_i is the number of persons in the census block,
 J_T is the total number of jobs in the EUC,
 P_T is the total number of persons in the EUC

Differences between this and Ewing et al. metric: The main differences between the two metrics is that this study used census blocks instead of census tracts and instead of using 1-mile rings to narrow block groups, blocks were used in their entirety.

6.1.2.3 Walk Score

Description: The Walk Score is a proprietary calculation that measures foot access to various services [79]. The algorithm weights services differently and also weights them by distance, with closer services increasing the Walk Score. The Walk Score alone is a powerful measure, and certainly correlates with compact development. It should be noted that the Walk Score measures ability to walk, not the percentage of walkers.

Units: undefined (Walk Score ranges from 0 to 100)

Relationship: Higher values equate to less sprawl

Datasets Used

- Census 2010 block data including demographic data (for populations)
- Longitudinal Employer Household Dynamics (LEHD) Origin-Destination
Employment Statistics version 7 (LODEST7) Workplace Area
Characteristics (WAC) 2010 (for jobs)
- Walk Score Inc., Walk Scores [79]

Algorithm: First, the Walk Score for every block center point must be gathered. The Walk Score script (outlined in Section 6.3.3) was used to acquire Walk Scores for each block in the EUC. Each Walk Score value was then weighted by the sum of the block's population and jobs versus the sum of the EUC's population and jobs.

Differences between this and Ewing et al. metric: Other than the fact that the data in this study was acquired using a script versus buying the data, the only difference is that blocks were used instead of tracts and block groups.

6.1.3 Activity Centering Metrics

Often confused with density and/or land use mix, activity centering looks at how well various activities are arranged in communities. High density can be achieved in the sprawl; think of the exurban (i.e., outlying suburban development area) community of Levittown, NY. Land use mixing is concerned with how well land uses are integrated throughout a community, but even a community with a good mix of uses might not have the density to sustain compact ideals such as walkability. Centrality is not concerned with land use mixes nor density, merely that the people (and/or the jobs) that exist are located nearer to each other and nearer to the CBD, no matter how dense or mixed this CBD might be.

The standard deviation of densities (population or employment) measures the scatter in density values around the mean. If these densities are more scattered, this will generally mean that there is more centering. Density gradient is a similar measure, but provides some added insight to centrality. A purer measure of centrality is to look at the percentage of people or jobs inside the CBD as compared to the total in the EUC. These areas make up the five metrics which combine to measure activity centering for the EUC. Each of these five is discussed below.

6.1.3.1 Coefficient of variation in block population densities

Description: This metric is the standard deviation of population density scaled by the average population density of blocks.

Units: persons per square mile

Relationship: Higher values equate to less sprawl

Datasets Used:

Census 2010 block data including demographic data (for populations and areas)

Algorithm: The standard deviation of population densities for all blocks within the EUC is found and divided by the mean of population densities for all blocks within the EUC.

Differences between this and Ewing et al. metric: Blocks were used instead of tracts.

6.1.3.2 Coefficient of variation in block employment densities

Description: This metric is the standard deviation of employment density scaled by the average employment density of blocks.

Units: jobs per square mile

Relationship: Higher values equate to less sprawl

Datasets Used:

Census 2010 block data including demographic data (for areas)

Longitudinal Employer Household Dynamics (LEHD) Origin-Destination

Employment Statistics version 7 (LODES7) Workplace Area

Characteristics (WAC) 2010 (for jobs)

Algorithm: The standard deviation of employment densities for all blocks within the EUC is found and divided by the mean of employment densities for all blocks within the EUC.

Differences between this and Ewing et al. metric: Blocks were used instead of tracts.

6.1.3.3 Density gradient moving outward from the CBD

Description: This metric is an estimate of the density gradient moving outward from the CBD based upon how quickly the population exponentially diminishes from the center of the EUC's CBD outward. This metric and the estimated central density (discussed in Section 6.1.1.6) work in tandem. The equation coefficients are calculated using population density of each block and distance of the center of each block to the center of the EUC's CBD.

Units: The inverse of miles

Relationship: Higher values equate to more sprawl

Datasets Used:

CBD Points (created manually by the author)

Census 2010 block data including demographic data (for populations, areas, latitudes, and longitudes)

Algorithm: First, a CBD points file must be generated with points denoting the center of the CBD of the primary city in the EUC. A discussion of this file's creation is included in Section 6.3.1. The first step in the algorithm was to calculate the difference in latitude and longitude between the CBD's center point and each block's center point (using the Census' internal point, which is the geographical center except for special cases where the point would fall outside the block's boundaries in which case the point inside the block closest to the center point is chosen [103]). Because these distances are around a geoid (the Earth) and not linear, the Haversine formula, recommended by the US Census Bureau to calculate short distances equations, was used to calculate each distance [104]:

$$d = 2 \cdot R \cdot \tan^{-1} \left(\sqrt{\sin^2 \frac{\phi_d}{2} + \cos \phi_b \cdot \cos \phi_e \cdot \sin^2 \frac{\lambda_d}{2}} \right)$$

where d is the distance between block and EUC's CBD center points, in mi.,
 R is the radius of the Earth, in m.,
 $\phi_d = \phi_b - \phi_e$,
 ϕ_b is the latitude of the block center point, in decimal degrees,
 ϕ_e is the latitude of the EUC's CBD center point, in decimal degrees,
 $\lambda_d = \lambda_b - \lambda_e$,
 λ_b is the longitude of the block center point, in decimal degrees,
 λ_e is the longitude of the EUC's CBD center point, in decimal degrees

The population density for each block was also stored. After all distances and densities were calculated and stored, a polynomial fitting function was used to derive the coefficients of the following exponential function using the natural log of each variable:

$$D = D_0 \cdot e^{-b \cdot d}$$

where D is the density of the block in persons per sq. mi.,
 D_0 is the estimated density at the EUC center in persons per sq. mi.,
b is estimated density gradient moving outward from CBD, in mi^{-1} ,
d is the distance between block and EUC's CBD center points, in mi.

The value of b is the metric. The value of D_0 is used as a density metric, discussed in Section 6.1.1.6.

Differences between this and Ewing et al. metric: Blocks were used instead of tracts. CBD center points were manually calculated to determine more accurate locations. It is unknown how Ewing et al. calculated distances between points, but they likely used a similar method and even if not, these relatively short distances will have little error.

6.1.3.4 Percentage of population in the CBD

Description: This metric is the ratio of the sum of population of the EUC's CBD to the sum of the population of the whole EUC.

Units: percentage (sum of population of CBD over sum of the population of EUC)

Relationship: Higher values equate to less sprawl

Datasets Used:

Census 2010 block data including demographic data (for populations)

Algorithm: First, a CBD polygon file must be generated for all EUC's primary cities. A discussion of this file's creation is included in Section 6.3.2. The population of each block in the EUC's CBD was then added together. The populations of each block in the EUC were also summed. The CBD sum was divided by the EUC sum and multiplied by 100 to calculate the percentage of population in the CBD.

Differences between this and Ewing et al. metric: There was some difference in how CBDs were created, which is discussed in detail in Section 6.3.2. Also, Ewing et al.'s metric included other employment sub-centers besides the CBD. These were not used in this study because smaller urban areas have fewer or no sub-centers and more importantly sub-centers can be considered sprawl in their own right and this study is designed to eliminate these from skewing results.

6.1.3.5 Percentage of employment in the CBD

Description: This metric is the ratio of the sum of jobs of the EUC's CBD to the sum of the jobs of the whole EUC.

Units: percentage (sum of employment of CBD over sum of the employment of EUC)

Relationship: Higher values equate to less sprawl

Datasets Used:

Longitudinal Employer Household Dynamics (LEHD) Origin-Destination
Employment Statistics version 7 (LODES7) Workplace Area
Characteristics (WAC) 2010 (for jobs)

Algorithm: First, a CBD polygon file must be generated for all EUC's primary cities. A discussion of this file's creation is included in Section 6.3.2. The number of jobs in each block in the EUC's CBD were then added together. The number of jobs in each block in the EUC were also summed. The CBD sum was divided by the EUC sum and multiplied by 100 to calculate the percentage of employment in the CBD.

Differences between this and Ewing et al. metric: There was some difference in how CBDs were created, which is discussed in detail in Section 6.3.2. Also, Ewing et al.'s metric included other employment sub-centers besides the CBD. These were not used in this study because smaller urban areas have fewer or no sub-centers and more importantly sub-centers can be considered sprawl in their own right and this study is designed to eliminate these from skewing results.

6.1.4 Street Accessibility Metrics

The final group of metrics used to calculate the Sprawl Score deal with the transportation network. The other three factors focused only on people and places, but the only way to connect these things is with a transportation network. There are two main ways to get at this relationship. First is to look at the block layout of the EUC. Blocks are defined by the census bureau based on the space between roads, so naturally they would help in defining street accessibility. Smaller blocks, and shorter blocks general mean more connectivity and more compactness. The second way to assess street connectivity is with street data, looking primarily at the layout of intersections and lack of things like cul-de-sacs. These areas make up the five metrics which combine to measure street accessibility for the EUC. Each of these five is discussed below.

6.1.4.1 Percentage of small urban blocks

Description: This metric gives the percentage of blocks in the EUC that are less than one hundredth of a square mile, which most urban blocks are. Blocks with an area greater than 1 square mile were excluded from this percentage.

Units: percentage (number of small blocks over total number of blocks)

Relationship: Higher values equate to less sprawl

Datasets Used

Census 2010 block data including demographic data (for areas and block counts)

Algorithm: The areas of each block were checked to see if they were less than 0.01 miles. In this case, they were added to the count of small blocks. Each block area was also checked to see if it was less than or equal to one mile, and in this case they were added to the count of normal blocks. After the completion of both summing operations, the number of small blocks in the EUC was divided by the number of normal blocks in the EUC and multiplied by 100 to give the percentage of small blocks in the EUC.

Differences between this and Ewing et al. metric: N/A

6.1.4.2 Average block size

Description: This metric gives the average size of blocks in the EUC and is a measure of total area of blocks in the EUC divided by the number of blocks. Blocks with an area greater than 1 square mile were excluded from this calculation.

Units: square miles per block

Relationship: Higher values equate to more sprawl

Datasets Used:

Census 2010 block data including demographic data (for areas and block counts)

Algorithm: The areas of each block were checked to see if they were less than or equal to one mile, and in this case they were added to the count of normal blocks and their areas were summed. After the completion of both summing operations, the area of normal blocks in the EUC was divided by the number of normal blocks in the EUC to give the average block size in the EUC.

Differences between this and Ewing et al. metric: N/A

6.1.4.3 Average block length

Description: This metric gives the average length of a block in the EUC and is a measure of total road length divided by number of road segments in the EUC.

Units: miles

Relationship: Higher values equate to more sprawl

Datasets Used:

Census 2010 block data including demographic data (for clipping regions)
Statewide All Roads Layer, Michigan Geographic Framework,
version 2014a [105]

Algorithm: The Michigan roads dataset is broken up into segments. Each segment of road is the edge of one block in the EUC. Therefore, all that is needed for this algorithm is to sum up the lengths of all of the roads in the EUC and also count the number of segments. Dividing the sum of segment lengths by number of segments gives the average block length.

Differences between this and Ewing et al. metric: This metric uses the Michigan roads dataset from the Michigan Geographic Framework as opposed to the one defined by the Census Bureau. This was done as the Michigan dataset already broke roads into segments based on blocks, making the calculation more convenient. Also, road lengths were taken across the entire EUC and not just the urbanized portion of the region, although Ewing et al. considered UAs, which are not too dissimilar to EUCs.

6.1.4.4 Intersection Density

Description: This metric gives the total number of intersections per square mile in the urban/suburban portion of the EUC. Only blocks with population densities of at least 50 persons per square mile were considered.

Units: intersections per square mile

Relationship: Higher values equate to less sprawl

Datasets Used:

Census 2010 block data including demographic data (for clipping region,
populations, and areas)
University of Utah Metropolitan Research Center (MRC) Intersection
Dataset [106] (for intersection counts)

Algorithm: This algorithm was significantly simplified by acquiring the intersections data file for all EUCs thanks to University of Utah's Metropolitan Research Center. This algorithm is described in Section 6.3.1. The intersection file includes points for every intersection in the EUC. First, all blocks with population densities over 50 (urban/suburban densities) within the EUC are selected from the block file. Then, all intersections within the selected blocks are selected. The total number of intersections are then counted and divided by the total land area of the high density blocks in the EUC to solve for intersection density.

Differences between this and Ewing et al. metric: Rural blocks were excluded rather than tracts, and only blocks with less than 50 people per square mile as opposed to 100 people per square mile were excluded.

6.1.4.5 Percentage of 4-or-more-way intersection

Description: This metric gives the percentage of 4-or-more-way intersections in the EUC compared to the total number of intersections in the EUC. Intersections with more than three entries offer more accessibility and choice in direction.

Units: percentage (number of 4-or-more-way intersections over total number of intersections)

Relationship: Higher values equate to less sprawl

Datasets Used:

Census 2010 block data including demographic data (for clipping region)
University of Utah Metropolitan Research Center (MRC) Intersection Dataset
(for intersection counts)

Algorithm: This algorithm was significantly simplified by acquiring the intersections data file for all EUCs thanks to University of Utah's Metropolitan Research Center. This algorithm is described in Section 6.3.1. The intersection file includes points for every intersection in the EUC, each of which had an attribute giving the number of directions entering the intersection. First, all intersections within the EUC were selected and the total number of intersections was counted. From this set of intersections, those with a count of intersections of 4 or more were selected and counted. The number of 4-or-more-way intersections was divided by the total number of intersections and multiplied by 100 to give the percentage of 4-or-more-way intersections.

First, all blocks with population densities over 50 (urban/suburban densities) within the EUC are selected from the block file. Then, all intersections within the selected blocks are selected. The total number of intersections are then counted and divided by the total land area of the blocks in the EUC to solve for intersection density.

Differences between this and Ewing et al. metric: N/A

6.2 Python Script: EUC_Metrics.py

Although many of the algorithms for calculating the 21 metrics are relatively simply to determine, some are more complex and when grouped together this can be large undertaking if done manually. For this reason, a script was created which processed the datasets as needed. Various support scripts were also used, and these are discussed in Section 6.3.

The benefits of a script include

- reduction of calculation time and repetition
- elimination of human error in processing all of the complex selections
- consistent application of the algorithm
- ease of change to the algorithm
- ease of expansion to a broader study area

The script to calculate the metrics for the EUCs was created with Python, which is the primary programming software used with ArcGIS. The full script is included in Appendix C. An overview of the script's operation is included in the following subsections.

6.2.1 Files needed for EUC_Metrics.py

The main file needed to run the metric calculator script is the EUC shapefile created by the EUC Creator program (described in Chapter 5). This script also currently requires some additional preprocessing, which is described in Section 6.3.1.

Beyond this file preparation, the script requires a seven additional files to operate (eight total). These are listed in Table 6.2.

Table 6.2: Files needed to run EUC_Metrics.py Python script

File Name	File Scope	Filename	File Developer	Vintage
Extended Urban Clusters*	State	EUC_All.shp	Riehl	2010
Blocks*	State	tl_2010_26_tabblock10.shp	US Census Bureau	2010
Central Business District Points	State	CBD_Points.shp	Riehl	2014
Central Business District Polygons	State	CBD_polygons_all.shp	Riehl#	2015
National Land Cover Database	National	nlcd_2011.img	MRLC Consortium	2011
NLCD Processed Table^	State	NLCD_table.dbf	Riehl	2011
Statewide All Roads Layer	State	allroads_miv14a.shp	MI DTMB MGF	2014
Intersection Points~	EUC	intersections_MI_EUC.shp	Utah MRC	2006

*Some pre-joining required. See Section 6.3.1 for details

#Main script was written by Riehl for this project. Two CBDs were appended from Hamidi [107]

^File created using techniques described in Section 6.3.1

~File is national, but file that was used was pre-clipped to the Michigan EUC boundaries. See Section 6.3.1 for more details

Some files are standard files that are freely available to the public and updated with the decennial census and others are created using methods discussed in Section 6.3. The only proprietary file is the intersection file. It is recommended that an intersection algorithm be developed as this is the only proprietary dataset needed to develop the Sprawl Score.

6.2.2 Overview of EUC_Metrics.py

The Python script is relatively basic in its operation, although many support operations need to be completed for many of the basic steps which makes the syntax and operation somewhat hard to follow. Although programmers will be able to follow along with the generous commenting throughout the script, those with only a basic knowledge of ArcGIS will benefit from a broader overview, which is provided in this subsection.

The basic flowchart for the main operation of the script is included in Figure 6.1.

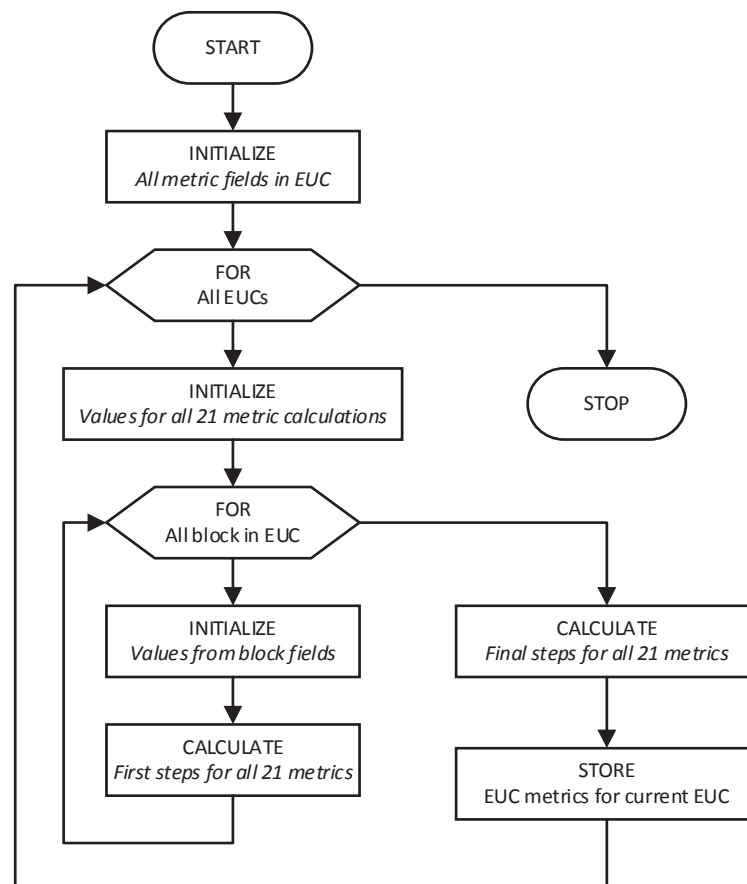


Figure 6.1: Basic flowchart for EUC metric calculation

This script will actually calculate all metrics for all EUAs and EUCs, although only the 42 pilot EUCs are needed for this study. As stated previously, this program calculates

the 21 metrics developed by Ewing et al. in 2014. The EUC metric calculation algorithm is explained at a high level for non-programmers in the sections below.

The EUC metric calculation algorithm was run for each Urban Area in Michigan. This was done to help troubleshoot and verify the algorithm to eliminate all minor flaws in the calculation of the metrics.

Additional information for non-programmers is included in Appendix C.

6.2.3 General Notes for Improving Algorithm

It was found that this algorithm was satisfactory with only a few differences from Ewing et al.'s algorithms. A few ideas were identified for extending this implementation to other problems. These are listed by metric.

- *popden* is not normalized, but it would be a better metric if it was
- *lt1500* and *gt12500* are not normalized to population, but they should be to provide for a better comparison between cities
- *urbdn* is not the best metric as it should be changed to be population density per unit area of urban land, not an urban land percentage
- *dgcent*, *popdcen*, *empdcen*, *dgrad*, *popcen*, *empcen* are all set to -1 for the 5 EUCs that do not have their CBD in Michigan
- *avgblkln* algorithm uses slightly different dataset and slightly different method from Ewing et al.

6.3 Support Scripts

In order for the metrics script to run correctly, several items needed to be completed before running the program. First, the EUC and block files needed to be prepared, as discussed in Section 6.3.1. Also, two support scripts were written, one to create CBD polygons (discussed in Section 6.3.2) and one to calculate Walk Scores (discussed in Section 6.3.3). These last two support scripts should be useful programs for future applications.

6.3.1 Preparation Work for EUC and Block Shapefiles

Various preparation work was done and columns were merged into the EUC and block file. These preparations are discussed in the following subsections.

6.3.1.1 Employment Data

Employment data was gathered from the Longitudinal Employer Household Dynamics (LEHD) Origin-Destination Employment Statistics version 7 (LODES7) Workplace Area Characteristics (WAC) 2010 data file. Specifically, the file used included all primary jobs located in each block, as opposed to all jobs which includes secondary jobs that are not an individual's primary work responsibility. Although there are various reasons a person may select a second job, the primary job is the one listed by each person as more prominent on the survey used to compile this data.

This data file comes in the form of a CSV file, specifically *mi_wac_S000_JT01_2010.csv* was used. The reason this file was not added to the block file programmatically is that not all blocks were included in this file. Specifically, blocks with no primary jobs were not included. Rather than running a complex sorting join in the program, the columns listed below were merged from this file to the block file manually. It is recommended that this join be added programmatically to the EUC metrics script.

- C000 – All primary jobs in the block
- CNS07 – Retail jobs
- CNS09 – Information services jobs
- CNS10 – Financial jobs
- CNS11 – Real estate jobs
- CNS12 – Technical service jobs
- CNS15 – Education jobs
- CNS16 – Health service jobs
- CNS17 – Entertainment jobs
- CNS18 – Hotel and food service jobs
- CNS19 – Other service jobs

EUC_Metrics.py brings in all of these fields and uses them in various calculations. C000 is used by many metrics, whereas the rest are used only by the degree of job mixing metric.

6.3.1.2 Land Use / Land Cover Data

Land cover data is needed for the urban density metric. This data was downloaded from the Multi-Resolution Land Characteristics (MRLC) Consortium's National Land Cover Database (NLCD) 2011 edition. The NLCD dataset is a raster dataset that separates land coverage by category. The categories needed for the urban density metrics were the four developed land categories, classes 21-24 (Developed, Open Space; Developed, Low Intensity; Developed, Medium Intensity; and Developed, High Intensity).

The zonal Spatial Analyst tool called *TabulateArea* was used in ArcGIS to tabulate all of the categories for each EUC. Specifically, the command uses the EUC file as the zone feature, the FID as the zone field, the NLCD file as the class file, the *NLCD_2011* field as the class field, *NLCD_table.dbf* as the output table, and 30 as the processing cell size.

The output file then includes a column for each NLCD category with areas given for each EUC row. The four columns of interest are added together and this column is added to the EUC file as *Devel_Lnd*.

Note that this algorithm should be added to the EUC Metrics script and the algorithm should be updated to sum the populations over the urban areas as well. This was not included in the version of the Sprawl Score, but should be included in an updated version.

6.3.1.3 Central Business District Point File

Central Business District center points are needed for two metrics: estimated density at the center of the metro area and density gradient moving outward from the CBD. The points are also used for support to decide whether or not the CBD of an EUC is within the state of Michigan or not. The five CBDs in UAs outside the state were not included in the final CBD point file.

The CBD points file denotes the center of the CBD of the primary city in the EUC for each of the EUCs. Originally, an algorithm was developed to select the points programmatically. Such ideas as selecting a civic location (post office, courthouse, city hall, etc.) were postulated, but some of these offices have moved to non-central locations. Even a weighted average of these locations was not sufficient. Thus, the 117 locations were manually determined using a combination of remote sensing, Google Street View [108], and the author's judgment.

6.3.1.4 Intersection Point File

An intersection points file was also created that includes a point for every intersection in the state. The only column of information that needs to be included in this file needs to be the count of entrances into this intersection. This file is used for the intersection density and 4-or-more-way intersection percentage metrics.

The file was obtained from the University of Utah's Metropolitan Research Center (MRC). It is recommended that this algorithm be programmed and stored as a separate support program for this software package. The dataset used as a base for the algorithm is TomTom's 2006 national road dataset which is included with ESRI's ArcGIS install package. An updated dataset should be used to attain more accurate metrics. The algorithm is summarized below, from Ewing et al.'s description [3]:

- Filter out freeways, unpaved tracks, and other roadways that don't function as pedestrian routes using Census Feature Class Code (CFCC) values
- Merge pairs of parallel centerline segments into single centerlines.
- Trim / extend streets intersecting the original centerlines to the merged segments
- Split each centerline at each intersection with side streets
- Locate roundabout centroids and assign a count of four incoming streets
- Delete endpoints of incoming streets into roundabouts
- Generate point features at both endpoints of each street segment in the file
- Delete points that are closer than 12m to each other
- Count number of points coinciding at any location and store count for each point
- Discard point counts of one or two from the file

6.3.2 Python Script: *CBD_Creator.py*

A Central Business District polygon file must be created in order to calculate four metrics: gross population density and employment density of the CBD in the EUC and percentage of the EUC's population and employment in the CBD. This algorithm was created as a separate program, and the program is currently in beta phase.

Currently, the program calculates CBDs for many of the EUCs using an algorithm similar to that in Ewing et al. However, many EUCs have CBDs with populations too small to be recognized by this algorithm. Therefore, after the program is executed, two CBDs are appended from Ewing et al.'s CBD polygon file [107] and the remaining CBDs were added manually using a combination of remote sensing, Google Street View [108], and the author's judgment.

The program *CBD_Creator.py* is included in Appendix D. The program requires the output file of *EUC_Creator.py* (the EUC shapefile), the CBD point shapefile, and the block shapefile in order to execute correctly. The basic algorithm is included in Figure 6.2. Essentially, the algorithm uses the Clusters and Outlier Analysis (Anselin Local Morans I [109]) Spatial Statistics tool in ArcGIS to calculate z-values for each block. Significant z-values are selected to be part of the CBD polygon and polygons are ensured to not have donuts. These CDBs are joined to create a CBD polygon shapefile.

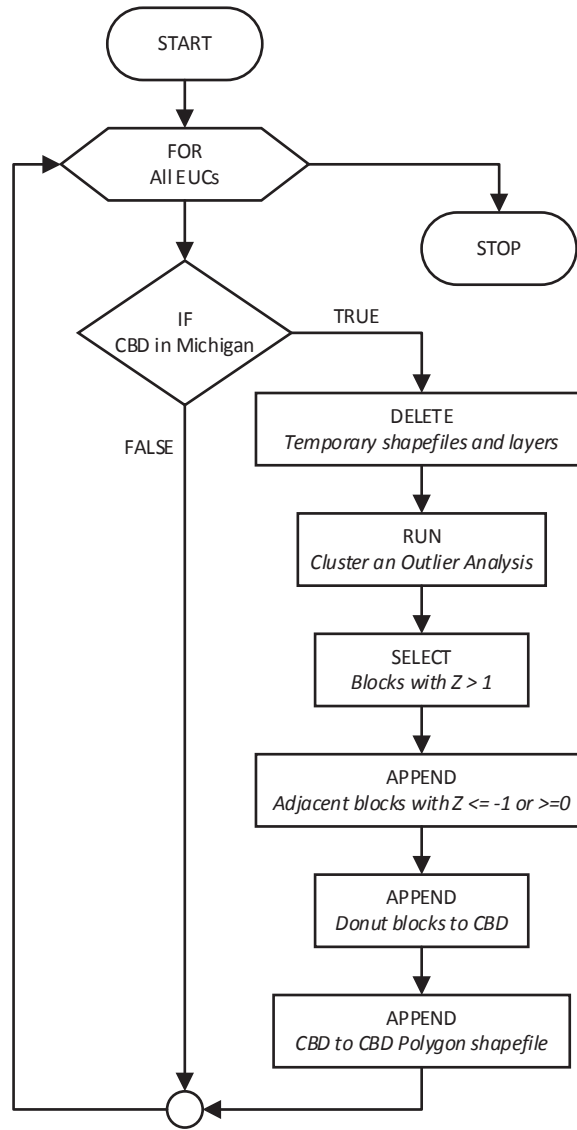


Figure 6.2: Basic flowchart for CBD polygon creation

6.3.3 Python Script: WalkScore_Locator.py

The Walk Score is used as one of the three land use mix metrics. Acquiring Walk Score data can be accomplished by querying the Walk Score website with a location and receiving back a Walk Score. As there were nearly 200,000 blocks that are part of the EUC areas that needed Walk Scores, a programmatic approach was required. A script was created to query the website for the values.

This script uses the generic Walk Score hyperlink that accepts latitude/longitude inputs:

<https://www.walkscore.com/score/loc/lat=LATNUM/lng=LONNUM>
where LATNUM and LONNUM are latitude and longitude values in decimal degrees

A flowchart for this step is presented in Figure 6.3 and the full script is included in Appendix E. The algorithm is more complicated than the figure suggests, so it is recommended that programmers review the comments in the script to fully understand its functionality. In basic terms, the program loops through block values querying the Walk Score website as shown above. The website returns the source code and the source code is parsed to obtain the Walk Score value. The Walk Scores are stored in a list and after all are found, they are written to a new file that includes the block's geographical identification, the latitude, the longitude, and the Walk Score.

This program should be executed before EUC_Metrics.py. Note that it is useful to create a block file with only the blocks of interest being each call to the website takes roughly one second to complete, which means 100,000 blocks would take about a day to run.

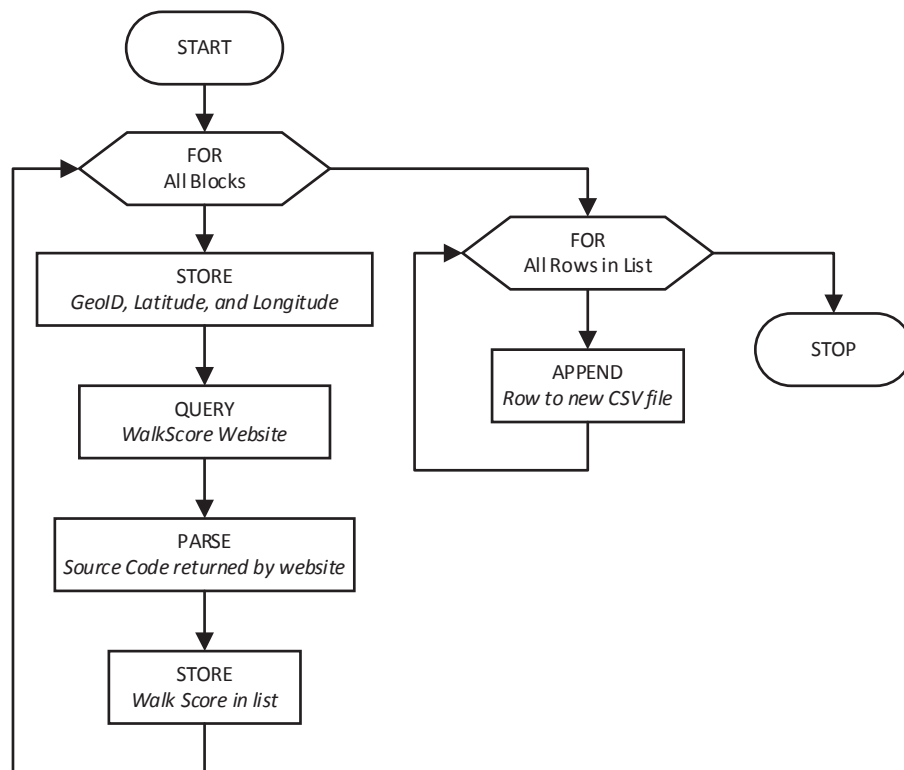


Figure 6.3: Basic flowchart for Walk Score locator program

Chapter 7: Sprawl Score

After calculating the metrics for the pilot cities, the scores for each of the four factor categories were determined using principal components analysis (PCA). These four factor categories (development density, land use mix, activity centering, and street accessibility) summarize the scores for the given community based on a weighted average of the metrics associated with each factor category. The extraction of principal components is discussed in Section 7.1.

Once Principal Components Analysis is completed, the four factor category scores are scaled to a system based upon a similar scale to an intelligence quotient (IQ) test, where a 100 is the average and the standard deviation is 25. Thus a score of 75 would be a more sprawling community and a 125 would be a more compact community based on the given factor category. These four scores, along with the 21 metric scores, can help a community narrow down their sprawl characteristics and better understand the level of sprawl in the community. The scores are calculated as part of the Sprawl Score calculator Python program as discussed in Section 7.2, with the full script included in Appendix F. The Python script file is also included in Appendix H, the enclosed DVD that is included in the media pocket on the inside back cover of this dissertation.

Once all four factor category scores are calculated, their values are combined programmatically into a final Sprawl Score. The final Sprawl Score for each community is scaled to the same extent as the four factor category scores. This score provides an overall sprawl rating for the entire EUC. The algorithm for the Sprawl Score is discussed in Section 7.2. Section 7.3 lists the Sprawl Scores, factory category scores, and original metric scores for all 42 pilot cities. Finally, Section 7.4 includes a discussion of the rankings and a sample community Sprawl Scorecard. Sprawl Scorecards for all 42 pilot cities are included in Appendix G. High Resolution scorecards are included in Appendix H.

7.1 Principal Components Analysis of Metrics

With 21 metrics, understanding how to correctly combine these into four summary scores is a challenge, especially due to the fact that there is certainly overlap in the variables. Although a simple weighted average would arrive at a score, manually selecting weights is a biased approach. Both factor analysis and principal components analysis are statistical algorithms used to simplify confounding datasets and cull out relevant summary values. This is done by taking uncorrelated linear combinations of the variables to arrive at fewer, less confounded variables.

The choice between factor analysis and principal components analysis depends on the datasets and the relationships between variables. When merely looking to reduce the number of components for variables that are highly correlated, as many in this study are, principal components analysis is preferred. Principal Components Analysis

explains the amount of variance in the variables, and for highly correlated variables, one principal component is usually all that is needed [110]. An added benefit is that Ewing et al. also used Principal Components Analysis for this step, so better comparisons can be made.

The steps necessary to complete the principal components analysis and produce a score for each of the four factor categories is summarized in the following section.

7.2 Python Script: EUC_Sprawl_Score.py

A script was written to calculate the sprawl scores and is included in Appendix F. This script is broken into three parts: data preparation, principal components analysis, and sprawl score calculation. These steps are shown in the basic flowchart in Figure 7.1 below and each is discussed in the following sections. There are a few minor errors that should be fixed for the next implementation of the script, and these are pointed out in the script comments.

The following sections are again written for non-programmers. The most complicated part about this program is actually understanding the statistics followed by understanding how to use arrays in Python. The statistics are discussed below, but the details of the Python programming are included in Appendix F.

7.2.1 Data preparation for script

After the EUC metrics script is executed, 21 raw metric scores are calculated for each EUC. These values need to be preprocessed in order to run principal components analysis and determine a score for each of the four factor categories.

It is important to note that PCA could be performed on the entire group of cities, just the 42 pilot cities, or the pilot cities by category. The approach to score pilot cities by category was chosen, even though error is increased at smaller sample sizes. The fundamental theory behind this is that cities can most accurately be compared to cities of similar size, and that the error produced by comparing, for example, Detroit to Saint Ignace, would be much larger than the errors associated with small sample size.

The main preprocessing step was to put all of the scores in each factor category into a large matrix which can be processed for PCA. These steps are outlined in the script in Appendix F.

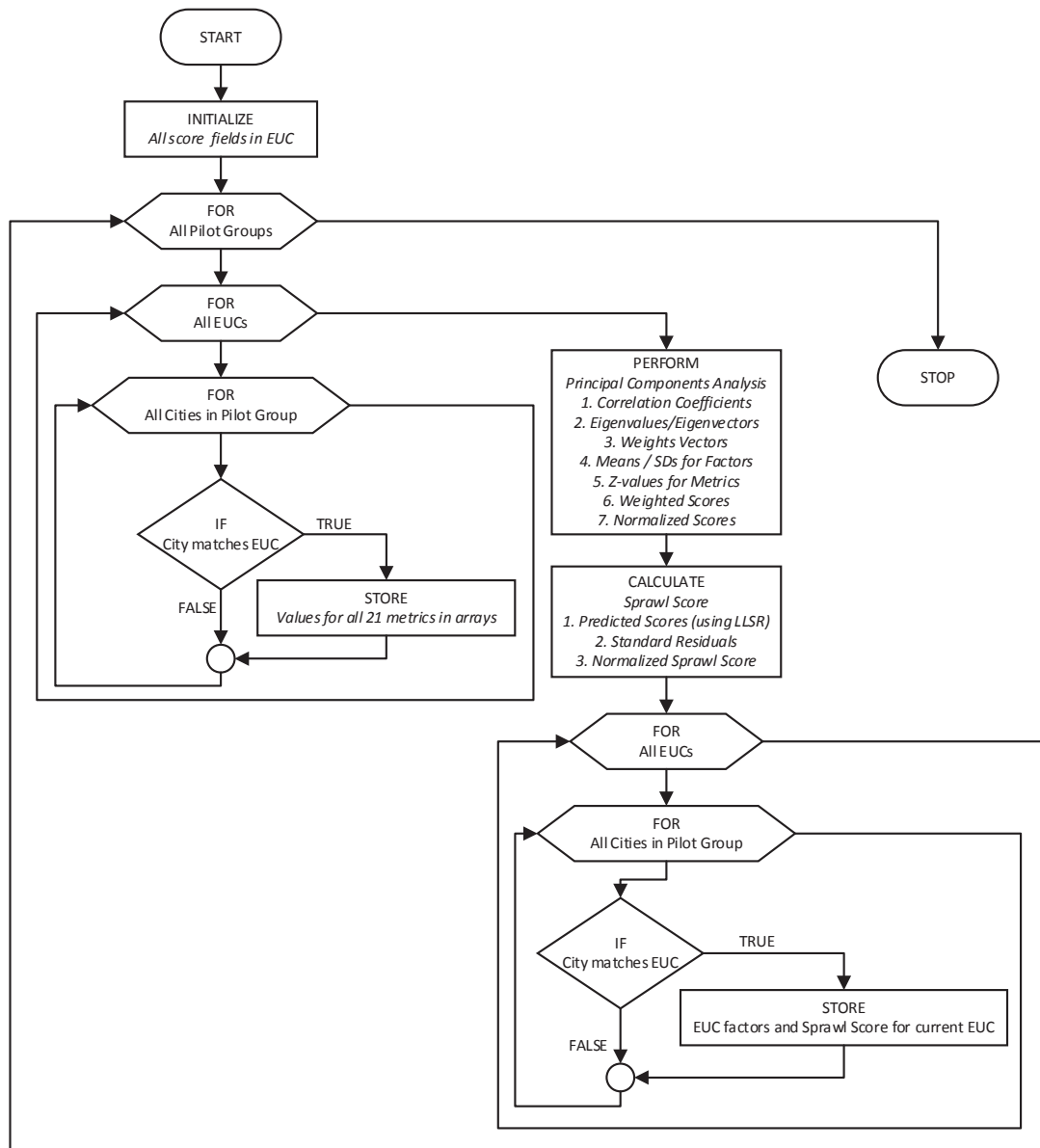


Figure 7.1: Basic flowchart for Sprawl Score calculator program

7.2.2 Principal Components Analysis Algorithm

Once all metric values are stored by group in the matrix, PCA is performed. Within each factor category, the following algorithm is used for PCA [111]:

- Determine a correlation matrix for all metrics in the category
- Calculate eigenvalues and eigenvectors for each factor category
- Calculate a weights vector to weigh importance of metrics for principal component calculation
- Calculate means and standard deviations for each factor category
- Calculate standardized value (z-value) for each metric in each factor category
- Calculate weighted score for each metric and sum for each factor category
- Normalize weighted average score for final factor category score for each of four factor categories

By performing the PCA, the metrics in each factor category is condensed into one overall value. Due to the statistical methods involved in PCA, the metric scores are not equally weighted in the calculation of their respective factor category score or the overall Sprawl Score. The exact weightings are calculated programmatically in the Sprawl Score Python script. The final weights are shown in Table 7.1.

Most of the metrics have an inverse relationship with the Sprawl Score, meaning that if the metric increases, the Sprawl Score will decrease. The only metrics with a direct relationship are: the percentage of persons living at less than 1,500 persons per square mile, the density gradient moving outward from the CBD, the average block size, and the average block length.

The percentage explained by the principal component score (i.e., the factor category score for each factor) is determined by the eigenvalue calculations in the PCA. These percentages are shown in Table 7.2. This table shows that all factor category scores explain at least 40% of the variability of the metrics contained in the scores.

Table 7.1: Metric Weights as Determined by Principal Components Analysis

		Metric Weights			
		Mid-Size Regional Cities	Small Regional Cities	Large Subregional Cities	Mid-Size Subregional Cities
Development Density	Population Den	0.415	0.423	0.486	0.503
	Employ Den	0.404	0.420	0.429	0.511
	Density at < 1.5K	-0.379	-0.389	-0.418	-0.398
	Density at > 25K	0.424	0.008	0.226	0.119
	Urban Density	0.247	0.444	0.478	0.501
	Density (Center)	0.231	0.315	0.269	0.203
	Pop Density (CBD)	0.272	0.320	0.050	0.140
	Emp Density (CBD)	0.391	0.307	0.225	0.034
Land Use Mix	Job / Pop Balance	0.553	0.564	0.583	0.627
	Job Mixing	0.722	0.576	0.649	0.665
	Walk Score	0.416	0.591	0.489	0.405
Activity Centering	Pop Variation	0.320	0.176	0.144	0.588
	Emp Variation	0.428	0.019	0.254	0.207
	Density Gradient	-0.573	-0.434	-0.352	-0.457
	Pop % (CDB)	0.492	0.594	0.610	0.472
	Employ % (CBD)	0.380	0.654	0.648	0.424
Street Accessibility	% Small Blocks	0.452	0.460	0.487	0.486
	Avg Block Size	-0.424	-0.479	-0.540	-0.463
	Avg Block Length	-0.467	-0.433	-0.539	-0.498
	Intersec Density	0.299	0.467	0.409	0.412
	% 4+ Intersections	0.556	0.391	0.112	0.363

Table 7.2: Metric Weights as Determined by Principal Components Analysis

		Metric Weights			
		Mid-Size Regional Cities	Small Regional Cities	Large Subregional Cities	Mid-Size Subregional Cities
Development Density		52	54	49	44
Land Use Mix		62	65	74	66
Activity Centering		61	42	43	41
Street Accessibility		61	79	60	68

7.2.3 Sprawl Score Algorithm

The result of the PCA is a normalized score for each of the four factor categories. The four results from the PCA are fed into the Sprawl Score algorithm. The Sprawl Score algorithm is relatively simple. First, a raw score is calculated as the sum of all four factor category PCA scores. Then a predicted sprawl score is calculated using a least-squares linear regression of the raw total score versus the natural logarithm of population of the primary city of each EUC. This is done to normalize the score based on population as a city with a larger population, other factors being equal, generally has more sprawl [3]. Modification to the normalization procedure should be considered for the next implementation of the script.

Next, the predicted value is compared to the expected (raw) value and a standard residual is calculated. This standard residual is normalized to an IQ-like scale and the resultant score is the Sprawl Score. This algorithm is explained in more detail in the script comments in Appendix F.

7.3 Rank of Pilot Cities with Sprawl Score

After completing the EUC Sprawl Score calculator program, each EUC has 21 component metric scores based on the units given in Chapter 6, four normalized factor category scores, and a normalized Sprawl Score. The final Sprawl Scores (including four factor category scores and 21 metric scores) are listed in Tables 7.1 through 7.4, for the four city size categories including mid-size regional (Table 7.1), small regional (Table 7.2), large subregional (Table 7.3), and mid-size subregional (Table 7.4).

Because the five composite scores (the four factor category scores and the Sprawl Score) are calculated using PCA for only the cities in the group, it should be understood that the ratings have most meaning when compared directly to peer cities in the group. A score of 100 for each of the five composite scores means that the EUC is average in the group. A score lower than 100 means that the EUC is below average and a score above 100 means that the EUC is above average, again when compared to peers in the group. However, the metrics are raw scores, thus using metrics cities can be compared within the group or across groups.

Table 7.3: Sprawl Scores (including all metrics) for mid-size regional cities

		Adrian	Marquette	Mount Pleasant	Traverse City
Sprawl Score		70	106	130	94
Development Density	Overall	120	48	161	71
	Population Den	385	201	351	285
	Employ Den	125	101	163	137
	Density at < 1.5K	58	69	61	74
	Density at > 25K	1.17	0.84	3.40	0.36
	Urban Density	19	11	16	18
	Density (Center)	800	390	570	640
	Pop Density (CBD)	10	15	22	2
	Emp Density (CBD)	73	106	1775	23
Land Use Mix	Overall	78	126	111	85
	Job / Pop Balance	0.21	0.23	0.18	0.21
	Job Mixing	138	150	166	146
	Walk Score	25	40	34	21
Activity Centering	Overall	60	125	89	126
	Pop Variation	1.4	1.7	2.3	1.9
	Emp Variation	6	10	14	7
	Density Gradient	0.8	0.7	1.0	0.6
	Pop % (CDB)	1.0	11.3	1.0	12.5
	Employ % (CBD)	11	53	26	52
Street Accessibility	Overall	84	113	97	106
	% Small Blocks	44	50	52	48
	Avg Block Size	0.09	0.08	0.09	0.07
	Avg Block Length	250	300	290	250
	Intersec Density	15.3	18.0	12.7	16.3
	% 4+ Intersections	23	33	38	16

Table 7.4: Sprawl Scores (including all metrics) for small regional cities

		Alma	Alpena	Big Rapids	Cadillac	Coldwater	Escanaba	Hillsdale	Houghton	Ionia	Lapeer	Ludington	Owosso	Sault Sainte Marie	Sturgis
	Sprawl Score	101	102	84	68	89	158	67	90	89	85	123	112	99	134
Development Density	Overall	119	72	58	74	88	118	52	60	101	99	150	141	87	182
	Population Den	345	163	168	183	288	274	179	93	204	343	402	393	207	532
	Employ Den	139	78	64	94	131	118	68	34	71	141	193	139	81	256
	Density at < 1.5K	57	65	81	66	68	58	77	67	70	68	55	49	59	49
	Density at > 25K	1.11	0.38	0.51	0.13	0.43	0.51	0.47	2.07	1.01	0.40	0.72	1.57	1.54	1.52
	Urban Density	18	15	12	17	17	20	14	6	12	21	31	21	15	27
	Density (Center)	710	460	290	420	390	990	420	280	240	360	560	920	470	380
	Pop Density (CBD)	8	12	32	5	9	2	3	73	69	6	3	3	27	36
	Emp Density (CBD)	26	301	1540	127	68	10	111	1049	3991	25	21	23	190	655
Land Use Mix	Overall	98	112	114	59	123	109	109	72	120	81	109	90	83	120
	Job / Pop Balance	0.13	0.19	0.18	0.13	0.23	0.19	0.22	0.12	0.28	0.20	0.18	0.16	0.16	0.17
	Job Mixing	148	133	144	106	143	121	133	107	132	129	129	108	111	119
	Walk Score	24	28	26	25	23	31	23	31	19	16	29	33	29	38
Activity Centering	Overall	76	97	132	90	89	203	83	123	111	76	75	70	113	62
	Pop Variation	1.5	1.4	3.3	1.5	2.6	1.5	1.9	2.5	1.8	1.6	1.3	1.2	1.4	1.6
	Emp Variation	7	12	16	16	5	8	5	13	17	8	18	6	6	13
	Density Gradient	1.6	1.0	0.9	1.1	1.2	0.9	1.1	0.6	1.1	1.3	1.5	1.7	0.7	2.5
	Pop % (CDB)	1.5	1.6	2.0	0.9	0.9	19.9	0.1	2.7	0.7	0.7	1.0	0.8	3.1	1.6
	Employ % (CBD)	10	19	42	15	8	81	8	26	38	5	9	11	28	17
Street Accessibility	Overall	111	126	34	68	55	163	40	100	35	80	169	125	118	177
	% Small Blocks	59	59	47	48	42	63	44	60	48	55	62	60	64	69
	Avg Block Size	0.08	0.05	0.11	0.09	0.09	0.04	0.10	0.06	0.11	0.07	0.05	0.07	0.08	0.04
	Avg Block Length	260	300	340	280	270	210	320	300	310	250	200	220	270	190
	Intersec Density	20.5	23.8	10.1	15.5	16.6	23.3	11.8	21.4	11.5	11.2	27.5	19.8	22.6	33.3
	% 4+ Intersections	33	36	27	27	21	42	26	24	21	22	43	32	32	30

Table 7.5: Sprawl Scores (including all metrics) for large subregional cities

		Caro	Gaylord	Greenville	Hastings	Houghton Lake	Manistee	Petoskey
Sprawl Score		71	78	118	135	76	106	116
Development Density	Overall	66	46	162	147	69	90	119
	Population Den	106	96	393	312	119	117	224
	Employ Den	48	37	203	148	23	41	158
	Density at < 1.5K	77	88	53	48	79	69	79
	Density at > 25K	0.21	0.00	0.00	0.00	0.00	0.23	0.37
	Urban Density	11	14	23	20	14	13	18
	Density (Center)	220	130	390	490	620	250	230
	Pop Density (CBD)	9	4	7	6	1	9	11
	Emp Density (CBD)	74	11	20	32	0	241	61
Land Use Mix	Overall	104	79	114	101	43	93	166
	Job / Pop Balance	0.21	0.18	0.23	0.20	0.16	0.15	0.28
	Job Mixing	148	131	138	130	97	132	181
	Walk Score	20	16	25	26	13	30	28
Activity Centering	Overall	89	158	56	125	73	122	78
	Pop Variation	5.2	2.2	1.2	1.0	1.4	1.6	2.1
	Emp Variation	8	9	7	5	6	12	5
	Density Gradient	1.4	0.6	2.5	1.5	0.6	0.8	1.0
	Pop % (CDB)	1.7	15.7	1.4	11.9	0.3	3.4	1.6
	Employ % (CBD)	14	64	5	66	0	47	11
Street Accessibility	Overall	23	59	126	144	127	116	106
	% Small Blocks	40	45	55	71	51	62	57
	Avg Block Size	0.17	0.10	0.06	0.05	0.05	0.08	0.07
	Avg Block Length	380	360	210	230	260	290	250
	Intersec Density	8.8	14.1	20.1	18.6	33.3	24.6	15.9
	% 4+ Intersections	32	18	32	38	21	29	17

Table 7.6: Sprawl Scores (including all metrics) for mid-size subregional cities

		Bad Axe	Charlevoix	Cheboygan	East Tawas	Fremont	Gladwin	Grayling	Harrison	Hart	Iron River	Kalkaska	Manistique	Munising	Newberry	Rogers City	Sandusky	Saint Ignace
Sprawl Score		97	89	100	95	95	105	60	45	99	84	90	110	107	151	118	139	116
Development Density	Overall	82	64	84	95	76	71	38	60	76	91	115	111	104	212	104	172	143
	Population Den	203	125	301	220	142	149	59	159	190	219	434	447	505	1699	236	744	729
	Employ Den	160	57	136	120	84	67	24	20	108	96	256	163	203	661	88	679	291
	Density at < 1.5K	72	80	70	60	72	67	89	80	74	59	67	65	66	43	46	38	66
	Density at > 25K	0.41	0.00	0.00	0.28	0.30	0.44	0.09	0.00	0.38	0.24	0.83	0.49	0.56	0.00	0.56	0.00	0.00
	Urban Density	18	15	31	19	13	16	12	16	18	17	34	34	25	74	17	50	64
	Density (Center)	120	210	170	410	290	120	60	270	100	370	110	170	150	100	210	100	190
	Pop Density (CBD)	28	24	1	1	16	7	1	1	30	3	20	35	18	12	32	19	34
Land Use Mix	Emp Density (CBD)	11	101	17	65	354	16	68	0	70	39	2250	154	30	40	172	197	69
	Overall	121	99	99	91	117	136	62	58	125	46	59	98	66	142	102	180	99
	Job / Pop Balance	0.31	0.22	0.21	0.17	0.26	0.29	0.16	0.19	0.27	0.13	0.13	0.18	0.17	0.42	0.19	0.40	0.15
	Job Mixing	158	137	131	127	138	163	95	92	149	73	76	123	76	155	107	185	148
Activity Centering	Walk Score	20	23	26	26	29	30	19	13	30	19	27	31	26	21	37	39	27
	Overall	114	165	70	78	132	125	158	91	110	74	100	97	142	53	79	41	70
	Pop Variation	1.7	1.9	1.4	1.2	1.6	1.5	2.6	1.4	1.8	1.2	1.7	1.4	1.6	0.9	0.9	0.8	1.2
	Emp Variation	6	5	4	5	5	6	11	6	7	6	9	4	4	4	8	3	9
	Density Gradient	1.3	0.9	2.0	1.5	1.5	1.9	0.7	1.3	1.7	1.8	2.4	2.0	2.2	2.1	1.9	2.5	2.2
	Pop % (CDB)	8.8	16.6	0.1	0.2	10.8	16.1	0.3	0.3	6.1	0.3	1.5	6.4	23.4	0.9	3.2	1.6	0.7
Street Accessibility	Employ % (CBD)	5	70	5	8	74	63	6	1	11	7	44	55	86	6	31	9	4
	Overall	64	42	111	116	35	78	34	31	100	129	111	115	117	185	167	130	137
	% Small Blocks	48	48	68	67	49	68	47	41	67	73	58	61	69	76	79	65	66
	Avg Block Size	0.08	0.07	0.04	0.05	0.14	0.10	0.08	0.09	0.06	0.03	0.03	0.03	0.03	0.02	0.02	0.04	0.02
	Avg Block Length	270	330	220	190	330	290	400	350	260	180	200	200	190	140	210	160	160
	Intersec Density	18.3	22.1	21.0	26.8	9.0	14.0	26.6	18.2	17.3	42.7	30.3	40.3	17.2	77.2	107.9	51.7	72.5
	% 4+ Intersections	32	20	34	36	35	34	24	27	36	31	36	32	33	50	37	33	28

7.4 Analysis of Rankings

An analysis of the values in Tables 7.3 through 7.6 showed that there were no calculation errors as numbers matched expected estimates. In general, most scores make sense when thinking about the area under consideration for each scores. Scores that do not immediately make sense can generally be explained with some context. This shows that the scores are at least somewhat representative of sprawl in those areas and the variation in scores shows that sprawl can indeed be measured for small cities.

Although scores were determined, rankings are never perfect and there is always room for improvement. Although future research might reduce this problem, these rankings are still rather useful for cities looking to make improvements and create more livable cities.

7.4.1 Overall Winners and Losers Based on Factor Categories

As with any score system, it is first useful to look at the winners and losers. Each of the groups is broken down by winners and losers by category in Table 7.7. Note that the most compact EUC in each group is Mount Pleasant, Escanaba, Manistee, and Newberry. The most sprawling EUCs in each group are Marquette, Owosso, Greenville, and Harrison.

Table 7.7: Best and worst EUCs based on factor categories and Sprawl Scores

Category	Mid-Size Regional	Small Regional	Large Subregional	Mid-Size Subregional
Dense	Mount Pleasant	Sturgis	Greenville	Newberry
Mixed	Marquette	Coldwater	Petoskey	Sandusky
Centered	Traverse City	Escanaba	Gaylord	Charlevoix
Accessible	Marquette	Sturgis	Hastings	Newberry
Compact	Mount Pleasant	Escanaba	Hastings	Newberry
Sparse	Marquette	Hillsdale	Gaylord	Grayling
Separated	Adrian	Cadillac	Houghton Lake	Iron River
Scattered	Adrian	Sturgis	Greenville	Sandusky
Inaccessible	Adrian	Big Rapids	Caro	Harrison
Sprawling	Adrian	Hillsdale	Caro	Harrison

Observations from Table 7.7 include:

- Multiple cities including Marquette, Sturgis, Gaylord, and Greenville appear in both the best and worst for certain factor categories underpinning the idea that the categories are distinct
- Marquette outperforms Mount Pleasant in number of best categories, but suffers due to a very low density score which gives Mount Pleasant the best score in the group
- Hastings only outperforms its peers on street accessibility, but it's other ratings are all on the compact side leading to the best overall score in the group
- Although Sandusky is very scattered, it performs well in all other areas and ultimately is measured to be very compact
- Although Coldwater does well on mixed use, it suffers in all other categories and ultimately is measured to be more sprawling

Some additional analyses could also be completed as future work. For example, Mount Pleasant does well compared to the other three cities in its group, but looking at its performance on the individual metrics versus other groups, the city is not necessarily doing well overall. Those scores which are lower compared to other groups should be addressed, even if they are higher than average within the peer group. Looking forward, more analysis could be done to not only compare cities within their groups, but perhaps a multifaceted sprawl score between groups would also be useful.

Another more complicated conclusion involves Newberry, which does well in all categories except activity centering. Newberry is rated as the most compact, but those who are familiar with Newberry know that many businesses exist outside of the central business district along the M-28 corridor. Because of the EUC algorithm, Newberry's gap in population between M-28 and the CBD caused the M-28 section not to be selected as part of the EUC. This does severely skew the results and should be noted by Newberry officials as well as peer cities. Although this is clearly an issue that needs to be addressed in subsequent studies, the scores still hold merit. For instance, when reviewing a high score for Newberry, careful attention should be paid to whether or not one could assume that score would change significantly based on updated EUC boundaries. For this reason, the EUC boundaries are placed on the Sprawl Scorecards.

In addition, a Sprawl Score less than 100 does not necessarily mean that the city is suffering from negative effects of sprawl, it simply means that there are more areas where improvement is needed than not. Cities should interpret the data by narrowing down to the metrics that are causing the negative scores. All cities have room for improvement, as not all cities have positive scores for every metric. The only time a city should be completely alarmed is when all of their four factor category scores are negative.

An example Sprawl Scorecard is shown below in Figure 7.2 and Sprawl Scorecards for each of the pilot cities are included in Appendix G. The street map in Figure 7.2 and the figures in Appendix G is sourced from ESRI's World Street Map dataset which is available for academic use [112]. Population numbers are from the 2010 Census [7]. The rest of the data in the figures were compiled from the datasets used to derive the Sprawl scores [79,101,102,105,106].

On the individual Sprawl Scorecards, the overall Sprawl Score as well as the four factor category scores are listed at the top. All values on the sheet are color-coded, with dark red representing significant sprawl to dark green representing very compact patterns. To determine the precise color for the Sprawl Score and factor category scores, standardized residual scores are used. For the rest of the 21 raw metric scores, z-scores are used to determine the precise color.

Peer cities are also listed on the scorecard for easier comparison. Below the summary are the four factor categories broken into the total of 21 metrics used for the scores. Each metric shows the overall raw score, the number of standard deviations away from the mean score in the group, the percentile rank of the city for that metric, and the best and worst performers among peer group members for that metric with their corresponding raw score.

From these color-coded Sprawl Scorecards, cities and regions can get a quick idea of where their city is doing well and where the city is suffering from negative effects of sprawl. These ratings can be compared using the summary scoresheet in Tables 7.3 through 7.6 or through a comparison with the Sprawl Scorecards of peer communities in Appendix G.

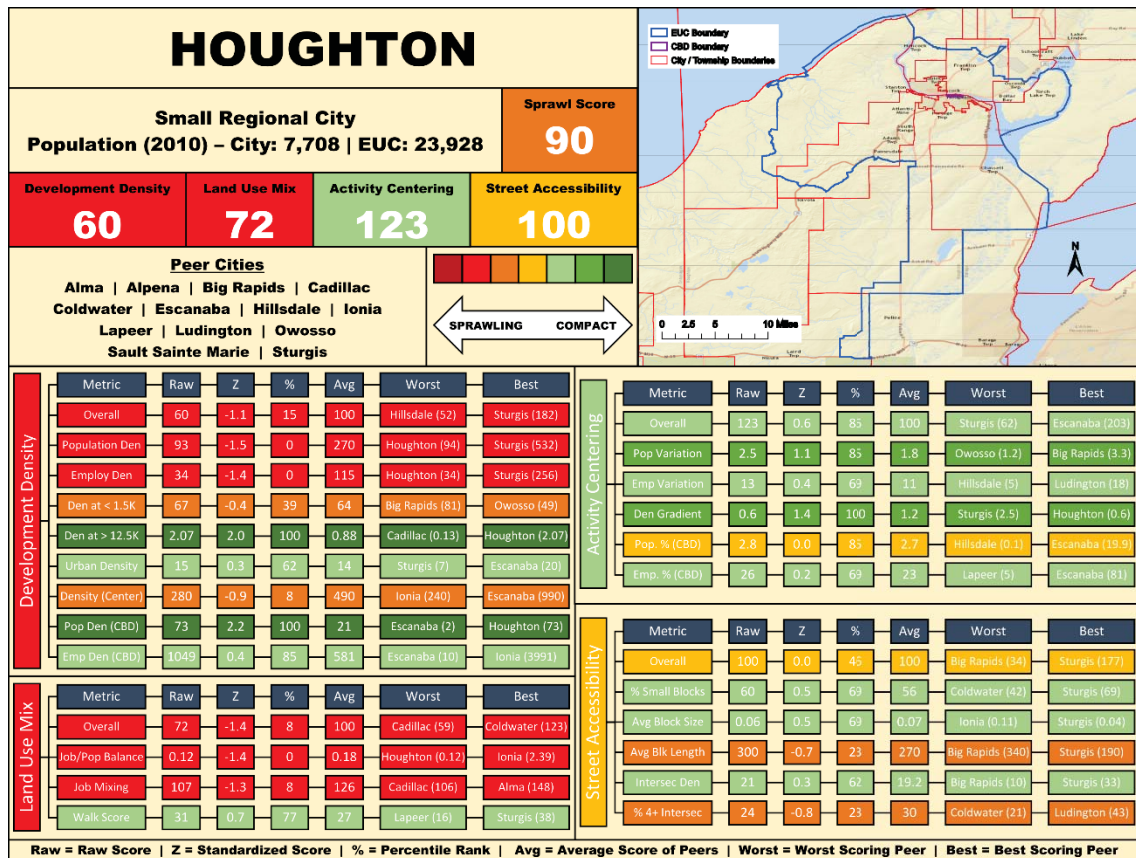


Figure 7.2: Example Sprawl Scorecard for Houghton, Michigan
Source for map is ESRI's World Street Map dataset [112]

7.4.2 How Communities Should Use the Sprawl Scorecard

To show how communities can use the Sprawl Scorecards to help in planning, an example study for Houghton is presented. Broad analyses of factor category scores and more pointed analyses of specific metrics within cities can both be very useful. Looking at the Houghton Sprawl Scorecard, there are some findings worthy of consideration.

Although Houghton has an overall negative development density score, this is primarily due to low population and employment density. Looking at the EUC boundaries and thinking about the large area of influence and small population of the community, this is not inherently a problem and probably does not need to be addressed. Especially when directly comparing this to all of the positive centering scores, which shows that Houghton has a vibrant and functional CBD. Anyone who lives within the boundaries of the Houghton EUC could tell you that the downtown is vibrant, and it is good to see that the centering numbers reflect this.

Even with the relative high score for activity centering however, there is still room for improvement for Houghton if the city wants an even more vibrant downtown. The two lowest scores in the category are percentage of housing and employment in the downtown. Upon thinking about the population and employment in the region, one could see that this is likely because of all of the employment on the hill outside downtown south on M-26 towards South Range. To improve this, Houghton could either incentivize more downtown living and working, or disincentivize building out the hill, or both. The Sprawl Score does not seek to tell cities like Houghton what to do, but is meant to be a tool to begin the process of narrowing down the question set to be asked and beginning to look at possible ways forward.

One final piece of information regarding the Houghton case study that cannot be ignored is Houghton's poor scores on land use mix. A planner's first question here should be, what specific metrics are causing this? The second question should then be, why is this the case and is this actually a problem? And finally if they deem this a problem, what could be done, if anything, to address the problem? These are not easy questions to answer, but the score should help supplement their knowledge of the area and draw attention to the issue.

To imagine the types of conversations that could occur, consider the following hypothetical scenario. Perhaps the main reason job mixing is so poor in Houghton is found to be the fact that all education jobs are a mile to the east at Michigan Tech. Perhaps this is not a problem because that is considered walkable. They could address the problem directly by adding more businesses along the corridor and growing the CBD. Or perhaps they could help address the problem indirectly by adding a fixed bus line between the school and downtown, which might not change the score, but would address the issue and make for a more livable city.

These types of conversations can be started based upon a quick analysis of the Sprawl Scorecards by local and regional officials. This shows the relevance of the scores and how the score can help small cities draw direct conclusions and make improvements to their cities.

Chapter 8: Conclusions and Recommendations

Conclusions are presented in Section 8.1, including a review of the results and a discussion of the answers to the original research questions and those that still remain. Section 8.2 presents a discussion of how the Sprawl Score can be used by city planners and regional planning agencies in order to begin adjusting their policies to mitigate sprawl in their communities. Finally, Section 8.3 discusses future research opportunities for advancing this work.

8.1 Conclusions

The goal of this study was to develop a method for measuring sprawl using a case study of small cities in Michigan. A scoring mechanism was developed and results were presented for the 42 small cities in the study. For each city, an overall Sprawl Score, four factor category scores for development density, land use mix, activity centering, and street accessibility, as well as 21 metric sub-scores were calculated. Although many metrics and algorithms were presented throughout the literature review, Ewing et al.'s 2014 list of metrics used for large cities was determined to be most applicable and was used for this research.

8.1.1 Research Question Summary

A primary research question and three additional questions were presented in Chapter 1. This section summarizes the findings of this research with respect to addressing these questions.

What metrics are best for quantifying sprawl?

Although the original goal of this research was to determine the best metrics to quantify sprawl, this objective was not fully reached although progress was made. The metrics selected for use were those vetted by the Ewing et al.'s 2014 scoring mechanism, which were released after this research was already underway. These metrics were referenced by the authors to be their best summarizing scores at the time of publication. Therefore, these were chosen for the analysis in this work.

The calculated scores do begin to show what elements are more important and how these elements interact to measure sprawl. Especially at the smaller population of the pilot cities, it becomes clear in comparison why certain metrics change based on small developmental differences in the cities. For instance, overall population density does not seem to be as important as percentage density within the CBD, based on the principal components analysis. Some metrics could be removed from the score in future work.

What measures have been used consistently to measure sprawl?

This question was addressed and summarized in the literature review. Although density seems to be the most common measure, the next five most used are accessibility, centrality, continuity, porosity, and mixed use.

It should also be noted here that researches have chosen a variety of metrics to assess each of the most used measures. Although most researchers have been using census or similar collected data, many researches have begun incorporating remote sensing and digital image analysis techniques to assess these measures.

Does sprawl exist at small scales?

With the differences in scores and the conclusions drawn from these values, it is clear that sprawl not only exists at small scales, but it can be identified and measured. Although the effects may not be as damning for small cities as large cities, small cities have noticeable, measureable sprawl. In fact, with the sensitivity to city size selection, there is a compelling argument that sprawl is easier to measure at small scales.

How do various measures of sprawl compare for the pilot cities of this study?

The answer to this question is summarized in Table 7.1. and Sprawl Scorecards show comparisons for 42 cities in the pilot group.

8.1.2 Other Conclusions

The geographical area selected around the core of the city is critical to many of the scores and the results show how sensitive the metrics are to city size selection. For larger cities, this is not as noticeable, but for smaller cities, the metrics can be skewed significantly by which blocks are included or not in the city definition.

This is actually a positive finding, as the fact that sprawl metrics in smaller cities are very sensitive to the area means that sprawl measured in smaller cities is easier to distinguish than for larger cities. As sprawl is easier to distinguish, small cities may be the best places to focus further research.

8.2 How the Sprawl Score can be used

The Sprawl Scorecards for each community along with the overall results could be sent to the mayor's office of each city along with an executive summary of this study. The aim is to help planning professionals and elected representatives better understand sprawl and focus their efforts to affect positive change for their community.

Following are a list of ideas for utilizing the Sprawl Score for those involved with city planning.

- Review the Sprawl Scorecard and understand the Sprawl Score, the four component scores, and the 21 metrics mean and how they are measured
- Review the best and the worst scores and compare these scores to other cities in the size range and across the study
- Host a brainstorming charrette with important players in the city's planning and development (e.g., local planners, government officials, developers, business leaders, and community leaders)
 - In this session, ideas should be drawn up as to why the city is the best and worse in the given categories
 - Acknowledge problem areas as all cities will have some problems to address; this is okay as the first step to solving problems is admitting they are problems
- Use session results to frame sprawl-related issues and pinpoint ideas for change
 - Take pride in areas with high scores
 - Begin addressing problem areas using specific programs
- Subsequent work and meetings should select specific metrics and relevant, realistic objectives to fix problem areas or highlight areas where score are strong
 - Remember that not all problems can be fixed at once
 - Any work in this regard will help make the city a better place for all residents

8.3 Recommendations for Future Research

This research marks the beginning of examining sprawl in small cities and using this as a way to identify the components of sprawl.

Future research should branch from and build off this foundation ultimately leading to the development of more precise measures for larger cities with the end goal of leading to a consensus in measuring sprawl. Consensus will provide policy makers and planners a better understanding of sprawl and will enable the pursuit of more logical development programs for 21st century cities.

There are several ways that this research could be extended, and a few directions are discussed in this section. Many of these items could be the focus future research.

8.3.1 Perform In-Depth Analysis of Sprawl Score for Pilot Cities

Although the Sprawl Score is best used as a practical tool for city planners, future research could be done by running an in-depth analysis of all pilot cities and component scores. Drawing more conclusions from this data could help communities as well as help to focus further research.

8.3.2 Extend the Scope of the Sprawl Score

This study should be extended to more cities around the world to help others to begin addressing their sprawl problems as well as to help further the research in identifying the best set of sprawl component metrics.

The Sprawl Score could be extended to EUCs in other states across the United States. With some minor changes to the data collection, this could be done relatively easily and would provide a much broader picture of the quality of these metrics.

The Sprawl Score could also be extended to all EUAs as well to see how effective it is with larger cities and to validate the metrics. Metrics for all EUCs and EUAs in Michigan have actually already been calculated along with the 42 pilot cities in this study, so there is already precedence and support for extending these algorithms to other EUAs.

Yet another possibility would be extending the Sprawl Score by focusing on different sized urban areas. For instance, scores could be looked at for core based statistical areas (CBSAs) or perhaps the U.S. Bureau of Economic Analysis's old Economic Areas. Smaller scales could be used as well including urban areas or even cities themselves, and perhaps even tracts or block groups within cities.

Finally, a culmination of these projects would be to compare metrics for cities using different geographical area units to draw conclusions about sprawl within the city. This could offer interesting insights into the location as well as causes and effects of sprawl.

8.3.3 Run Sensitivity Testing to Improve Geographical Area Selection

Future research should focus on sensitivity testing for even better city size selection. Size of the area of influence could start at the size of the Central Business District and increase until the appropriate size is determined.

Another method would be to do sensitivity testing in tandem with calculating metrics. This could be completed by looking at each individual metric and varying the size of the city scope block-by-block to look at effects on the metric score.

Either of these studies would not only help narrow down metrics, but they would help in understanding exactly what types of city elements improve or worsen scores. Through careful selection and analysis of area, not only will sprawl measures be more precise, but it will be possible to examine exactly what dynamics cause the metrics to change.

8.3.4 Reduce the Number of Metrics

With the components measured and presented in this research, the data can be mined to remove redundant metrics and list the most important metrics in identifying sprawl. Studies of differing scopes and the improvement of EUC and CBD algorithms will also support this endeavor.

Another route would be to begin by experimenting with more metrics, comparing each metric across cities in the research area and narrowing to only the most significant metrics.

Although this project focused on a primary group of 21 metrics, future research should focus on paring down the number until a fundamental grouping is achieved. The goal should be to reduce the number of metrics to as few as possible to simplify the modeling of the sprawl system. A simpler system will make the measure more transparent and more usable. With a simplified system, true scientific experiments, with only one independent variable changing, can be performed. That being said, enough variables should still be used to make sure the model is robust and effective.

8.3.5 Improve Tools and Extend Tools to Other Areas

The EUC selection algorithm, although a good start, could be improved. Some cities had too much area selected (e.g., Houghton) and some had too little (e.g., Newberry), which skewed some metric calculations. An improved algorithm will help make for more precise, meaningful scores

Central Business Districts are also areas of extreme importance. In developing the algorithms for the Sprawl Score, there were few examples of selecting CBD center points or polygons. The CBD center point creation in this research was a manual process and the polygon creation was not complete. CBD analysis plays an important role in the sprawl and other urban studies.

The Walk Score location tool has many applications beyond just sprawl measurements to allow for the quick amalgamation of Walk Scores. This algorithm could be used by cities and regions looking to assess walkability, real estate agents to collect walkability for their current offerings, businesses to assess walkability of all of their locations, as well as researchers utilizing the scores in mass for any reason.

8.3.6 Address Other Research Realms of Sprawl

Although this research primarily focused on determining the components of sprawl, the research also overlaps with better understanding the causes and consequences of sprawl. Initially, these metrics should also be applied to studies of causes and consequences to gauge their effectiveness.

As components of sprawl usually occur for a specific reason, these reasons can be addressed to determine the root causes of sprawl. Small city research is where reasons will be much clearer to identify as the links between causes, components, and consequences are less convoluted.

These numbers could also be used as part of the growing body of research on the consequences of sprawl, as only larger cities have been examined thus far. Although consequences may not be as severe for these smaller cities, results would still be intriguing. Research in both of these areas could also help support research on larger cities.

8.3.7 Perform Historical Comparisons with Sprawl Score

Although this research focused primarily on the present time period, many researchers have begun to compare sprawl across time as a metric for sprawl. These metrics should be calculated for each of the past three census periods and compared.

Also, understanding the historical zoning and master plans will also shed light on the sprawl development in the city. These components should be considered in any historical analysis of sprawl.

Understanding how sprawl organically grew in each city could show smaller cities how best to grow or even if growth is the best option.

8.3.8 Make the Sprawl Score more Accessible

A user-friendly Sprawl Score web interface could be developed. The Sprawl Score could be used by the general public as well as professionals in a manner similar to Walk Score. Although the Walk Score algorithm is well accepted, the most successful part of the Walk Score is the accessibility of the score to the public through the website. A website is useful for increasing the prominence of Sprawl Score.

8.3.9 Analyze Inter-Urban Influence

Cities do not exist in a vacuum. Just because a person uses a city for a certain purpose does not mean the person uses the city for all purposes. In other words, cities influence other cities, and this influence needs to be understood in order to best interpret the results of a Sprawl Score as well as to improve the Sprawl Score.

All cities in this study have unique issues. For instance, Sault Sainte Marie is certainly influenced by the larger population in Sault Sainte Marie, Canada. Some urban clusters have multiple cores, such as Houghton's separate, but related (and adjacent) core of Hancock. Others, such as Alma and Mount Pleasant, share a border. And many of the UCs are influenced by larger UAs due to proximity.

These issues could be analyzed to help better understand a given city, to better understand the metrics, and to improve the metrics.

References

- [1] R. Bruegmann, *Sprawl: A Compact History*, Chicago, IL: University of Chicago Press, 2006.
- [2] J. L. Renne, "Smart Growth and Transit-Oriented Development at the State Level: Lessons from California, New Jersey, and Western Australia," *Journal of Public Transportation*, vol. 11, no. 3, pp. 77-108, 2008.
- [3] S. Hamidi and R. Ewing, "Measuring Sprawl and Its Impacts: An Update," *Journal of Planning Education and Research*, 2015.
- [4] R. Ewing and S. Hamidi, "Measuring Urban Sprawl and Validating Sprawl Measures," Metropolitan Research Center at the University of Utah, Salt Lake City, UT, 2014.
- [5] D. Owen, "Everywhere else should be more like New York," *The New Yorker*, 2004.
- [6] H. E. Nasser and P. Overberg, "A comprehensive look at sprawl in America," *USA Today*, 2001.
- [7] "United States Census Bureau," [Online]. Available: <http://www.census.gov>. [Accessed 2014 13 January].
- [8] A. Frenkel and M. Ashkenazi, "Measuring urban sprawl: how can we deal with it?," *Environmental and Planning B: Planning and Design*, vol. 35, no. 1, pp. 56-79, 2008.
- [9] G. Galster, R. Hanson, M. R. Ratcliffe, H. Wolman, S. Coleman and J. Freihage, "Wrestling sprawl to the ground: Defining and measuring an elusive concept," *Housing Policy Debate*, vol. 12, no. 4, pp. 681-717, 2001.
- [10] G. Squires, Ed., *Urban Sprawl: Causes, Consequences, & Policy Responses*, Washington, DC: Urban Institute Press, 2002.
- [11] J. Jacobs, *The Death and Life of Great American Cities*, New York, NY: Random House, 1961.
- [12] M. Miller, "What Causes Sprawl?," National Center for Policy Analysis, Dallas, TX, 2003.
- [13] K. T. Jackson, *Crabgrass Frontier: The Suburbanization of the United States*, New York, NY: Oxford University Press, 1987.
- [14] The Nielson Company, "Millennials Prefer Cities to Suburbs, Subways to Driveways," New York, NY, 2014.
- [15] M. E. Lewyn, "Sprawl in Europe And America," *San Diego Law Review*, vol. 46, no. 1, pp. 85-112, 2009.
- [16] A. Duany, E. Plater-Zyberk and J. Speck, *Suburban Nation: The Rise of Sprawl and the Decline of the American Dream*, New York, NY: North Point Press, 2010.

- [17] A. Norman, *Slam-Dunking Wal-Mart: How You Can Stop Superstore Sprawl in Your Hometown*, St. Johnsbury, VT: Brigantine Media, 1999.
- [18] LandWatch Monterey County, "Beyond Sprawl: New Patterns of Growth to Fit the New California," Salinas, CA, 1995.
- [19] Sprawl Retrofit Initiative, "The unbearable costs of sprawl," The Congress for the New Urbanism, Chicago, IL, 2011.
- [20] C. Zimmerman and L. Sobel, "The Fiscal Implications of Development Patterns," Smart Growth America and RCLCO, Washington, DC, 2015.
- [21] M. Munasinghe, "Environmental Economics and Sustainable Development," in *UN Earth Summit, Rio de Janeiro, Environment Paper No.3, World Bank*, Washington, DC, 1992.
- [22] M. E. Kahn, "The Quality of Life in Sprawled versus Compact Cities," in *OECD ECMT Regional Round*, Berkeley, CA, 2006.
- [23] Towers Watson, "Workplace stress leads to less productive employees: Excessive pressure leads to low engagement and high absenteeism," [Online]. Available: <http://www.towerswatson.com/en/Press/2014/09/Workplace-stress-leads-to-less-productive-employees>. [Accessed 3 April 2015].
- [24] B. Bhatta, *Analysis of Urban Growth and Sprawl from Remote Sensing*, New York, NY: Springer, 2010.
- [25] R. W. Wassmer, "An Economic Perspective on Urban Sprawl: With an Application to the American West and a Test of the Efficacy of Urban Growth Boundaries," California State University, Sacramento, 2002.
- [26] R. Burchell, A. Downs, S. Mukherji and B. McCann, *Sprawl Costs: Economic Impacts of Unchecked Development*, Washington, DC: Island Press, 2005.
- [27] R. W. Burchell and D. Listokin, "Land, Infrastructure, Housing Costs, and Fiscal Impacts Associated with Growth: The Literature on the Impacts of Traditional versus Managed Growth," in *Alternatives to Sprawl Conference*, Washington, DC, 1995.
- [28] R. Ewing, R. Pendall and D. Chen., "Measuring Sprawl and Its Transportation Impacts," *Transportation Research Record*, vol. 1832, pp. 175-183, 2003.
- [29] M. J. Trowbridge and N. C. McDonald., "Urban Sprawl and Miles Driven Daily by Teenagers in the United States," *American Journal of Preventive Medicine*, vol. 34, no. 3, pp. 202-206, 2008.
- [30] N. McDonald and M. Trowbridge, "Does the built environment affect when American teens become drivers? Evidence from the 2001 National Household Travel Survey," *Journal of Safety Research*, vol. 40, no. 3, pp. 177-183, 2009.
- [31] E. J. Zolnik, "The effect of sprawl on private-vehicle commuting outcomes," *Environment and Planning A*, vol. 43, no. 8, pp. 1875-1893, 2011.
- [32] Y. Fan and Y. Song., "Is Sprawl Associated with a Widening Urban-Suburban Mortality Gap?," *Journal of Urban Health: Bulletin of the New York Academy of Medicine*, vol. 86, no. 5, pp. 708-728, 2009.

- [33] D. Nguyen, "Evidence of the impacts of urban sprawl on social capital," *Environment and Planning B: Planning and Design*, vol. 37, no. 4, pp. 610-627, 2010.
- [34] R. Ewing, T. Schmid, R. Killingsworth, A. Zlot and S. Raudenbush, "Relationship Between Urban Sprawl and Physical Activity, Obesity, and Morbidity," *American Journal of Health Promotion*, vol. 18, pp. 47-57, 2003.
- [35] A. Kelly-Schwartz, J. Stockard, S. Doyle and M. Schlossberg, "Is Sprawl Unhealthy? A Multilevel Analysis of the Relationship of Metropolitan Sprawl to the Health of Individuals," *Journal of Planning Education and Research*, vol. 24, pp. 184-196, 2004.
- [36] R. Sturm and D. Cohen, "Suburban Sprawl and Physical and Mental Health," *Public Health*, vol. 118, no. 7, pp. 488-496, 2004.
- [37] S. Cho, Z. Chen, S. Yen and D. Eastwood, "The Effects of Urban Sprawl on Body Mass Index: Where People Live Does Matter," in *The 52nd Annual ACCI Conference*, Baltimore, MD, 2006.
- [38] S. Doyle, A. Kelly-Schwartz, M. Schlossberg and J. Stockard, "Active Community Environments and Health: The Relationship of Walkable and Safe Communities to Individual Health," *Journal of the American Planning Association*, vol. 72, no. 1, pp. 19-31, 2006.
- [39] R. Ewing, R. Brownson and D. Berrigan, "Relationship Between Urban Sprawl and Weight of U.S. Youth," *American Journal of Preventive Medicine.*, vol. 31, pp. 464-474, 2006.
- [40] D. Kim, S. Subramanian, S. Gortmaker and I. Kawachi, "U.S. state-and county-level social capital in relation to obesity and physical inactivity: a multilevel, multivariable analysis," *Social science & medicine*, vol. 63, no. 4, pp. 1045-1059, 2006.
- [41] A. Plantinga and S. Bernell, "The Association between Urban Sprawl and Obesity: Is It a Two-Way Street?," *Journal of Regional Science*, vol. 47, no. 5, pp. 857-879, 2007.
- [42] C. E. Joshu, T. K. Boehmer, R. C. Brownson and R. Ewing, "Personal, neighbourhood and urban factors associated with obesity in the United States," *Journal of Epidemiology and Community Health*, vol. 62, pp. 202-208, 2008.
- [43] I. Lee, R. Ewing and H. Sesso, "The Built Environment and Physical Activity Levels: The Harvard Alumni Health Study," *American Journal of Preventive Medicine*, vol. 37, no. 4, pp. 293-298, 2009.
- [44] B. Griffin, C. Eibner, C. Bird, A. Jewell, K. Margolis, R. Shih and J. Escarce, "The Relationship between Urban Sprawl and Coronary Heart Disease in Women," *Health and Place*, vol. 20, pp. 51-61, 2013.
- [45] R. Ewing, G. Meakins, S. Hamidi and A. C. Nelson, "Relationship between urban sprawl and physical activity, obesity, and morbidity – Update and refinement," *Health and Place 26(2014) 118-126.*, vol. 26, pp. 118-126, 2014.

- [46] R. Ewing, R. Schieber and C. Zegeer, "Urban Sprawl as a Risk Factor in Motor Vehicle Occupant and Pedestrian Fatalities," *American Journal of Public Health*, vol. 93, pp. 1541-1545, 2003.
- [47] M. J. Trowbridge, M. J. Gurka and R. O'Connor, "Urban Sprawl and Delayed Ambulance Arrival in the United States," *American Journal of Preventive Medicine*, vol. 37, no. 5, pp. 428-432, 2009.
- [48] R. Ewing, J. Kostyack, D. Chen, B. Stein and M. Ernst, "Endangered by Sprawl: How Runaway Development Threatens America's Wildlife," National Wildlife Federation, Smart Growth America, and NatureServe, Washington, DC, 2005.
- [49] D. Czamanski, I. Benenson, D. Malkinson, M. Marinov, R. Roth and L. Wittenberg, "Urban Sprawl and Ecosystems—Can Nature Survive?," *International Review of Environmental and Resource Economics*, vol. 2, no. 4, pp. 321-366, 2008.
- [50] B. Stone, "Urban Sprawl and Air Quality in Large U.S. Cities," *Journal of Environmental Management*, vol. 86, pp. 688-698, 2008.
- [51] L. Schweitzer and J. Zhou, "Neighborhood Air Quality Outcomes in Compact and Sprawled Regions," *Journal of the American Planning Association*, vol. 76, no. 3, pp. 363-371, 2010.
- [52] B. Bereitschaft and K. Debbage, "Urban Form, Air Pollution, and CO2 Emissions in Large U.S. Metropolitan Areas," *The Professional Geographer*, vol. 65, no. 4, pp. 612-635, 2013.
- [53] Western Resource Advocates, Smart Water: A Comparative Study of Urban Water Use Efficiency Across the Southwest, Chapter 4, Boulder, CO: Western Resource Advocates, 2003.
- [54] R. Ewing and F. Rong, "Impact of Urban Form on U.S. Residential Energy Use," *Housing Policy Debate*, vol. 19, pp. 1-30, 2008.
- [55] B. Stone, J. Hess and H. Frumkin, "Urban Form and Extreme Heat Events: Are Sprawling Cities More Vulnerable to Climate Change than Compact Cities?," *Environmental Health Perspectives*, vol. 118, no. 10, pp. 1425-1428, 2010.
- [56] R. Ewing, R. Pendall and D. Chen, "Measuring sprawl and its impact, Volume 1," Smart Growth America, Washington, DC, 2002.
- [57] M. E. Kahn, "Does sprawl reduce the black/white housing consumption gap?," *Housing Policy Debate*, vol. 12, no. 1, pp. 77-86, 2001.
- [58] R. Lopez, "Thirty years of urban sprawl in metropolitan America," Fannie Mae Foundation, Washington, DC, 2004.
- [59] Metro, "Urban Growth Boundary," [Online]. Available: http://www.oregonmetro.gov/sites/default/files/UGB_080814.pdf. [Accessed 16 July 2014].
- [60] R. Burchell, N. A. Shad, D. Listokin, H. Phillips, A. Downs, S. Seskin, J. S. Davis, T. Moore, D. Helton and M. Gall, "TCRP Report 39: The costs of sprawl – revisited," Transportation Research Board, Washington, DC, 1998.

- [61] A. Brown, C. Collins, T. Frank, K. Haddow, B. Hitchings, S. Parry, G. Vanderpool and L. Wormser, "Sprawl: The dark side of the American dream.," Sierra Club, San Francisco, CA, 1998.
- [62] A. Hahs and M. J. McDonnell, "Selecting independent measures to quantify Melbourne's urban-rural gradient," *Landscape and Urban Planning*, vol. 78, no. 4, pp. 435-448, 2006.
- [63] A. Schneider and C. E. Woodcock, "Compact, dispersed, fragmented, extensive? A comparison of urban growth in twenty-five global cities using remotely sensed data, pattern metrics and census information.," *Urban Studies*, vol. 45, no. 3, p. 659, 2008.
- [64] J. Huang, X. X. Lu and J. M. Sellers, "A global comparative analysis of urban form: Applying spatial metrics and remote sensing," *Landscape and Urban Planning*, vol. 82, no. 4, pp. 184-197, 2007.
- [65] S. Angel, J. Parent and D. Civco, "Urban sprawl metrics: an analysis of global urban expansion using GIS," in *In Proceedings of the 2007 American Society of Photogrammetry and Remote Sensing Annual Convention*, Tampa, FL, 2007.
- [66] D. Arribas-Bel, P. Nijkamp and H. Scholten, "Multidimensional Urban Sprawl in Europe: A Self-Organizing Map Approach," *Computers, Environment, and Urban Systems*, vol. 35, no. 4, pp. 263-275, 2011.
- [67] J. H. Lowry and M. B. Lowry, "Comparing Spatial Metrics That Quantify Urban Form," *Computers, Environment, and Urban Systems*, vol. 44, pp. 59-67, 2014.
- [68] E. Glaeser, M. Kahn and C. Chu, "Job sprawl: Employment location in U.S. Metropolitan Areas," Center for Urban & Metropolitan Policy, The Brookings Institution, Washington, DC, 2011.
- [69] H. Wolman, G. Galster, R. Hanson, M. Ratcliffe, K. Furdell and A. Sarzynski, "The Fundamental Challenge in Measuring Sprawl: Which Land Should be Considered?," *Professional Geographer*, vol. 57, no. 1, pp. 94-105, 2005.
- [70] J. Jaeger, R. Bertiller, C. Schwick and F. Kienast, "Suitability Criteria for Measures of Urban Sprawl," *Ecological Indicators*, vol. 10, no. 2, pp. 397-406, 2010.
- [71] J. A. G. Jaeger, R. Bertiller, C. Schwick, D. Cavens and F. Kienast, "Urban Permeation of Landscapes and Sprawl per Capita: New Measures of Urban Sprawl," *Ecological Indicators*, vol. 10, no. 2, pp. 427-441, 2010.
- [72] P. Zito and G. Salvo, "Toward an Urban Transport Sustainability Index: an European Comparison," *European Transport Research Review*, vol. SP, no. 3, pp. 179-195, 2011.
- [73] N. Schwarz, "Urban form Revisited--Selecting Indicators for Characterising European Cities," *Landscape and Urban Planning*, vol. 96, no. 1, pp. 29-47, 2010.
- [74] P. C. Sutton, "A Scale-Adjusted Measure of Urban Sprawl" Using Nighttime Satellite Imagery," *Remote Sensing of Environment*, vol. 86, no. 3, p. 353, 2003.

- [75] W. Ji, J. Ma, R. W. Twibell and K. Underhill, "Characterizing Urban Sprawl using Multi-Stage Remote Sensing Images and Landscape Metrics," *Computers, Environment, and Urban Systems*, vol. 30, no. 6, pp. 861-879, 2006.
- [76] E. H. Wilson, J. D. Hurd, D. L. Civco, M. P. Prisloe and C. Arnold, "Development of a Geospatial Model to Quantify, Describe and Map Urban Growth," *Remote Sensing of Environment*, vol. 86, no. 3, p. 275, 2003.
- [77] M. Herold, J. Scepan and K. C. Clarke, "The Use of Remote Sensing and Landscape Metrics to Describe Structures and Changes in Urban Land Uses," *Environment and Planning A*, vol. 34, no. 8, pp. 1443-1458, 2002.
- [78] B. Bhatta, S. Saraswati and D. Bandyopadhyay, "Urban Sprawl Measurement from Remote Sensing Data," *Applied Geography*, vol. 30, no. 4, pp. 731-740, 2010.
- [79] "Walk Score," Redfin, [Online]. Available: <http://www.walkscore.com>. [Accessed 13 December 2013].
- [80] Economist Intelligence Unit, "Best Cities Ranking and Report," The Economist, London, UK, 2012.
- [81] Mercer, LLC., "2014 Quality of Living Worldwide City Rankings - Mercer Survey," Mercer, LLC., New York, NY, 2014.
- [82] "AARP Liveable Communities," AARP, [Online]. Available: <http://www.aarp.org/liveable-communities>. [Accessed 27 May 2015].
- [83] J. Tratalos, R. A. Fuller, P. H. Warren, R. G. Davies and K. J. Gaston, "Urban form, Biodiversity Potential and Ecosystem Services," *Landscape and Urban Planning*, vol. 83, no. 4, pp. 308-317, 2007.
- [84] Center for Water and Society, "Huron Creek Watershed Management Plan," Michigan Technological University, Houghton, MI, 2009.
- [85] A. H. Maslow, "A Theory of Human Motivation," *Psychological Review*, vol. 50, pp. 370-396, 1943.
- [86] M. A. Bragg, "Couple Completes Visits to Lower 48's 3,108 Counties with Trip to Nantucket," *Cape Cod Times*, p. 1, 11 March 2015.
- [87] D. Farr, *Sustainable Urbanism: Urban Design With Nature*, Hoboken, NJ: John Wiley and Sons, 2007.
- [88] "2014 TIGER/Line Shapefiles (machine-readable data files)," Prepared by the U.S. Census Bureau, Washington, DC, 2014.
- [89] ESRI, "ArcGIS Version 10.1," Environmental Systems Research Institute, Redlands, CA, 2012.
- [90] "Python 2.7," Python Software Foundation, Beaverton, OR, 2010.
- [91] "2010 Census Urban and Rural Classification and Urban Area Criteria," U.S. Census Bureau, Washington, DC, 2010.
- [92] "Regional Prosperity Initiative," Michigan Department of Technology, Management, and Budget, Lansing, MI, 2013.

- [93] U.S. Census Bureau, "Standard Hierarchy of Census Geographic Entities," [Online]. Available: <http://www2.census.gov/geo/pdfs/reference/geodiagram.pdf>. [Accessed 30 November 2014].
- [94] Bureau of Economic Analysis (BEA), "BEA Economic Areas (EAs)," United States Department of Commerce, 2004, Washington, DC.
- [95] USDA Economic Research Service (ERS), "2013 Rural-Urban Continuum Codes," United States Department of Agriculture, Washington, DC, 2013.
- [96] "World Imagery," ESRI, DigitalGlobe, Earthstar Geographics, CNES/Airbus DS, GeoEye, USDA FSA, USGS, Getmapping, Aerogrid, IGN, IGP, swisstopo, and the GIS User Community, Redlands, CA, 2015.
- [97] J. Cutsinger, G. Galster, H. Wolman, R. Hanson and D. Towns, "Verifying the Multi-Dimensional Nature of Metropolitan Land Use: Advancing the Understanding and Measurement of Sprawl," *Journal of Urban Affairs*, vol. 27, no. 3, p. 235, 2005.
- [98] J. Shlens, "A Tutorial on Principal Component Analysis," Systems Neurobiology Laboratory, Salk Institute for Biological Studies and Institute for Nonlinear Science, University of California, San Diego, La Jolla, CA, 2005.
- [99] USDA Economic Research Service (ERS), "Rural-Urban Commuting Area Codes," United States Department of Agriculture, Washington, DC, 2013.
- [100] USDA Economic Research Service (ERS), "Rural-Urban Commuting Area Codes Documentation," United States Department of Agriculture, Washington, DC, 2010.
- [101] "Longitudinal Employer Household Dynamics (LEHD) Origin-Destination Employment Statistics version 7 (LODES7) Workplace Area Characteristics (WAC) 2010," U.S. Census Bureau, Washington, DC, 2010.
- [102] C. Homer, J. Dewitz, L. Yang, S. Jin, P. Danielson, G. Xian, J. Coulston, N. Herold, J. Wickham and K. Megown, "Completion of the 2011 National Land Cover Database for the Conterminous United States-Representing a Decade of Land Cover Change Information," *Photogrammetric Engineering and Remote Sensing*, vol. 81, no. 5, pp. 345-354, 2015.
- [103] "Geographic Terms and Concepts - Geographic Area Attributes," U.S. Census Bureau, [Online]. Available: https://www.census.gov/geo/reference/gtc/gtc_area_attr.html. [Accessed 17 July 2014].
- [104] A. Hedges, "Finding Distances Based on Latitude and Longitude," [Online]. Available: <http://andrew.hedges.name/experiments/haversine/>. [Accessed 19 March 2015].
- [105] Center for Shared Solutions and Technology Partnerships, "Michigan Geographic Data Library - MI Geographic Framework All Roads (v14a)," Michigan Department of Technology, Management, and Budget, Lansing, MI, 2014.

- [106] University of Utah Metropolitan Research Center (MRC), "Intersection Shapefile," University of Utah Metropolitan Research Center, Salt Lake City, UT, 2014.
- [107] S. Hamidi, "Central Business Districts Shapefile," University of Utah Metropolitan Research Center, Salt Lake City, UT, 2014.
- [108] Google Maps, "Google Street View," Google, Inc., Mountain View, CA, 2014.
- [109] L. Anselin, "Local Indicators of Spatial Association—LISA.," *Geographical Analysis*, vol. 27, no. 2, pp. 93-115, 1995.
- [110] D. Suhr and M. Shay, "Guidelines for Reliability, Confirmatory and Exploratory Factor Analysis," University of Northern Colorado, Greeley, CO, 2008.
- [111] S. Raschka, "Implementing a Principal Component Analysis (PCA) in Python Step by Step," 2014. [Online]. Available: http://sebastianraschka.com/Articles/2014_pca_step_by_step.html. [Accessed 25 April 2015].
- [112] "World Street Map," ESRI, HERE, DeLorme, USGS, Intermap, iPC, NRCAN, OpenStreetMap Contributors, and the GIS User Community, Redlands, CA, 2015.

Appendix A: Copyright Permission Form for Figure 3.1



Michigan Technological University

Department of Engineering Fundamentals

112 Grover C. Dillman Hall
1400 Townsend Drive
Houghton, MI 49931-1295
906-487-3057 • Fax 906-487-1620

July 24, 2015

Alex S. Mayer
630 Dow Environmental Sciences
1400 Townsend Drive
Houghton, MI 49931

Dear Alex Mayer:

I am completing a doctoral dissertation at Michigan Technological University entitled "Development of a Fundamental Rating System for Identifying Sprawl: A Case Study Utilizing Small Cities in Michigan." I would like your permission to reprint in my dissertation figures from the following:

Center for Water and Society, "Huron Creek Watershed Management Plan," Michigan Technological University, Houghton, MI, 2009.

The figures to be reproduced are: Figure 2.1 and Figure 2.7.

The requested permission extends to any future revisions and editions of my dissertation, including non-exclusive world rights in all languages, and to the prospective publication of my dissertation by ProQuest through its ProQuest® Dissertation Publishing business. ProQuest may produce and sell copies of my dissertation on demand and may make my dissertation available for free internet download at my request. These rights will in no way restrict republication of the material in any other form by you or by others authorized by you. Your signing of this letter will also confirm that you own or Michigan Technological University owns the copyright to the above-described material.

If these arrangements meet with your approval, please sign this letter where indicated below and return it to me as a scanned email. Thank you very much.

Sincerely,

Signature Redacted 7/24/15

Jonathan Riehl

PERMISSION GRANTED FOR THE USE REQUESTED ABOVE:

Signature Redacted

Alex Mayer

Date: 7/24/15

www.mtu.edu

We prepare students to create the future.

Michigan Technological University is an equal opportunity educational institution/equal opportunity employer.

Appendix B: EUC_Creator.py – Overview Full Python Script

B.1 Overview for Programmers

This section is meant to be an addendum to Chapter 5, Section 2 for those well versed in programming. The overall algorithm for determining EUCs seems relatively straightforward, but there are many points where the bits and pieces of this algorithm becomes tricky to program. Hence, this section is provided to make it easier to replicate and extend this program for other applications.

This section is intended to guide those well versed in computer programming to be able to use and modify the EUC creation Python script for their work. To begin, it is recommended that Section 5.2 is read first to understand the basic overview of the software. This section will cover some of the nuances that were not covered in that section.

In order to execute this script, a copy of ESRI ArcGIS version 10.0 or higher with Python 2.6 or higher is recommended. When the script is executed, many support shapefiles are created and the script will take between a few hours and a couple of days to execute, depending on processor speed. It is recommended that this program is run on a workstation with at least 1GB of memory and 5GB of free disk drive space. In order to run the package, the following files must be in the current directory: *euc_creator.py*, *MI_UAC_2010.shp*, *ruca2010Michigan.csv*, *tl_2010_26_bg10.shp*, and *tl_2010_26_tabblock10.shp*.

The script utilizes the extensive ArcPy library as well as the csv library (for dealing with RUCA file processing). The script has two built-in functions, *delete_file* and *create_layer*. *Delete_file* works for both layers and shapefiles, checking to see if the file exists, and if so deleting the file. *Create_layer* first deletes the layer using *delete_file* and then makes a new layer. Both functions are used extensively, as many shapefiles and layers are created and deleted throughout the program.

It should also be noted that a fair amount of SQL statements are used throughout the script to perform complex selections or removals from selections. Although a working knowledge of SQL will help, advanced ArcGIS users should be familiar with the syntax of these commands.

Comments are included throughout the script to help explain the functionality of the steps in the program. Block comments separate the program into sections and short comments discuss the functionality of individual lines of code.

Although many of the comments are self-explanatory, some of the broader concepts will be discussed in the following subsections including reasoning and methodology of the loops in the program, the donut addition algorithm, and the RUCA algorithm. The final subsection will address some of the issues that occurred as the script was being developed to help achieve a better understanding and appreciation for why certain pieces of the algorithm were developed.

B.1.1 Reasoning and Methodology of Loops in Algorithm

There are a total of 10 for loops and one while loops in the script. Each is described briefly below (tab spacing shows level of loop in the script):

- **for loop 1 – Main loop (1st UAC loop):** This loops through each UAC in order of decreasing population density, each time creating a new EUC, appending appropriate block groups and blocks, and adding it to the EUC shapefile.
 - **for loop 2 – 2nd UAC loop:** This loops through each UAC for each run of the main loop in order to remove block groups that overlap any other UAC (i.e., no block group that intersects more than one UAC will be fully included in and EUC).
 - **for loop 3 – 1st EUC loop:** This loops through EUCs created in prior runs of loop 1 in order to remove block groups that overlap any larger EUC (i.e., larger EUCs get preference on block groups that could otherwise be included in multiple EUCs).
 - **while loop – Block while loop:** Each run through this loop will add on additional outlying blocks to the current EUC. This loop will run indefinitely, until there are no more outlying blocks that meet the criteria for addition to the EUC. The while flag will be set to 1 to stop the loop only when no blocks are available.
 - **for loop 4 – Block for loop:** This loops through each outlying block to the current boundaries of the EUC (these are working boundaries which are growing slowly until the EUC is fully created). Each run through the loop determines if the block should be included or not in the EUC.
 - **for loop 5: RUCA tract loop:** This loops through each RUCA tract in the RUCA file, to see if the current block is in the RUCA tract.
 - **for loop 6: RUCA code loop:** This loops through each RUCA code to see if the current block's tract is acceptable or not.
 - **for loop 7: 3rd UAC loop:** This loops through each UAC in order to remove blocks that overlap any other UAC. This process is very similar to the 2nd UAC loop except it is checking blocks instead of block groups

- **for loop 8: 2nd EUC loop:** This loops through EUCs created in prior runs of loop 1 in order to remove blocks that overlap any larger EUC. This process is very similar to the 1st EUC loop except it is checking blocks instead of block groups.
- **for loop 9: 1st “dummy” loop:** This loop actually doesn’t really loop at all (it loops through one value, the current EUC being worked on). The reason for this is that the EUC needs to be named and this seemed like the easiest method to add the name to the EUC. This may be made simpler in a future version of the script.
- **for loop 10: 2nd “dummy” loop:** This loop is identical to loop 9, except it is executed in the case that the EUC file is already created (everything but the first run of loop 1)

B.1.2 Donut Addition Algorithm

Throughout the algorithm that adds block groups and blocks to each EUC, there is the chance that “donuts” will be created. A donut is an area where block groups or blocks were added to the EUC that ended up fully enclosing another block group or block. For continuity purposes, these need to be added into the EUC. There are two times where the current EUC is checked for donuts.

The donut algorithm works similarly in each case it is used. First, a *union* operation is performed on the current EUC with the “NO_GAPS” option selected, which assigns an FID of -1 to all gap (i.e., donut) block groups (and blocks, depending on the algorithm). All of these gap block groups (and blocks) are selected and any with zero land area are removed from the selection. The selected block groups (and blocks) are then appended to the current EUC shapefile.

B.1.3 RUCA Algorithm

The purpose of the RUCA algorithm is to verify that a block is indeed within a census tract that has at least 30 percent of workers commuting to the UAC. Ideally, this information would be available at the block level, but the only dataset that is maintained with this information is the RUCA dataset from the United States Department of Agriculture’s Economic Research Service.

The complete definition of RUCA codes from the USDA-ERS website is:

The rural-urban commuting area (RUCA) codes classify U.S. census tracts using measures of population density, urbanization, and daily commuting. The most recent RUCA codes are based on data from the 2010 decennial census and the 2006-10 American Community Survey. The classification contains two levels. Whole numbers (1-10) delineate metropolitan, micropolitan, small town, and rural commuting areas based on the size and direction of the primary (largest)

commuting flows. These 10 codes are further subdivided based on secondary commuting flows, providing flexibility in combining levels to meet varying definitional needs and preferences [99].

A table of RUCA codes used in the 2010 RUCA file are included above in Table 5.3. Further documentation can be obtained at the USDA-ERS website [100].

The RUCA algorithm is executed for each set of new outlying blocks being checked to be added to the EUC (during the while loop). For loop 4 loops through every outlying block in the current iteration of the while loop and for each block, loops through the rows in the RUCA table (loop 5). Each row in the RUCA table includes one census tract along with its RUCA code. The RUCA file is a comma-separated value (CSV) file, with the following columns:

- State-County FIPS Code
- State
- County
- State-County-Tract FIPS Code
- Primary RUCA Code 2010
- Secondary RUCA Code, 2010
- Tract Population, 2010
- Land Area (square miles), 2010
- Population Density (per square mile), 2010

Only columns 3 and 5 (starting at a count of 0) are stored, which are the Tract FIPS Code and the Secondary RUCA Code. The Tract FIPS Code for the current row in the RUCA table is compared to the Tract FIPS Code of the current block in question. Once the loop locates a match (the correct row in the table is found), loop 6 is entered which loops through all acceptable RUCA codes for the given type of UA (UZA, large UC, or small UC). If the stored secondary RUCA code matches an acceptable code, then the RUCA flag is set to 1, meaning that the block is acceptable. If after execution of loop 5 the RUCA flag was never raised, the block is removed from consideration. Otherwise, the block is accepted.

B.1.4 Issues and Fixes

Throughout the development of the script, many errors were encountered and fixed. Some of these fixes involved convoluting the program, which is why they are discussed here to aid the user in understanding the algorithm.

- Many EUCs contained blocks over water, specifically for Michigan being one of the Great Lakes. Clearly water is not a place where most people will decide to build (except for extreme circumstances a la Dubai's Palm Islands or Mumbai's filling in of islands to create one mega-land mass). To fix this problem, there

are many places in the algorithm where block groups and blocks with 0 land area are removed from the selection before sets are added to the EUC.

- There were situations where EUCs would overlap each other, which is clearly not desirable. These situations are complicated, as these are times where the OMB would combine sections into CSAs, and they will offer limitations in developing the sprawl score which will be discussed in that chapter. However, at this point, overlaps should be removed. To do this, a few checks throughout the script to eliminate overlaps. Specifically, loops 2, 3, 7, and 8 were created for this purpose.
- Some situations arose where certain blocks were not being added when they should have been (they met the criteria for addition). This was due to two distinct problems. The first was simply an error in the algorithm where block groups were being checked when block checks were called for. The second was more critical. Datasets from different years were being used (specifically 2010 for those datasets only updated on a decennial basis and 2013 for datasets updated on a yearly basis). To fix this problem, only datasets from 2010 were used, even if updated datasets were available. This is important to note when extending this algorithm. 2010 datasets should be used at least until all the 2020 datasets are released. Note that RUCA datasets seem to be the latest released, as the 2010 RUCA dataset was released December 2013.
- For some EUCs, not all of the UAC was included. This was not acceptable, as the smallest a EUC can be is the size of the UAC that it contains. This happened because some block groups were removed from consideration because they were either included in a bordering EUC or a bordering UAC. To fix this problem, code was added to add back in any blocks missing from the EUC after all block groups were processed and before the block while loop was executed. This section of code is found under the block comment “Add UAC intersecting blocks to the current EUC.”
- An issue existed where block groups or blocks on the border of a EUC were not selected when they should have been if the borders exactly lined up, but did not cross. To fix this for block groups, search distances of -1 feet were added to block group intersection commands. For block selection commands, a series of a “boundary touches” location selection followed by a “within” location removal command were executed.

As discussed earlier, some block groups that were added resulted in unreasonably large EUCs, especially in more rural areas such as the Upper Peninsula. To fix this problem, a check was added to make sure that block groups over 5x the size of the original UAC area were not added to the EUC.

B.2 EUC_Creator.py – Full Python Script

```
#####EUC_Creator.py#####
##This program creates a shapefile including all Extended Urban Clusters for##
##urban areas in Michigan. This uses a modified approach to ##
##Galster, et. al's "Which Land Should be Considered?" ##
## 1. The EUC consists of the Census Bureau defined urban cluster boundary ##
## 2. Boundaries are modified to follow census block group boundaries ##
## 3. "Outlying" blocks are added that ##
## 3a. contain 60 or more dwelling units per sq. mile ##
## 3b. and are located in a census tract from which at least ##
## 30 percent of the workers commute to the UC ##
#####

#Import Libraries
import arcpy
import csv

#####
##Function Declarations##
#####

def delete_file(file,type):
    '''This function checks if a file exists and if so, deletes the file'''
    if arcpy.Exists(file) == 1:
        arcpy.Delete_management(file,type)
    return

def create_layer(fromfile,newfile):
    '''This function deletes an old layer and makes a new layer'''
    delete_file(newfile,"LAYER")
    arcpy.MakeFeatureLayer_management(fromfile,newfile)
    return

#####
##Initializations##
#####

#Initialize folder and file locations
arcpy.env.workspace = "H:/PHD/EUC_Creator/"
bg_file = "tl_2010_26_bg10.shp"
blk_file = "tl_2010_26_tabblock10.shp"
uac_file = "MI_UAC_2010.shp"
ruca_file = "H:/PHD/EUC_Creator/ruca2010Michigan.csv"

#Initialize names that will be used throughout the program
geoid_field = "GEOID10" #Used in block layer
name_field = "NAME10" #Used in the UAC layer
area_field = "ALAND10" #Used in the UAC layer
pop_field = "POP10" #Used in the UAC layer

#Initialize constants that will be used throughout the program
m2_to_mi2 = pow(100,2)/pow(2.54,2)/pow(12,2)/pow(5280,2)
ruca_accept_UA = ['1.0','1.1','2.0','2.1','4.1','5.1','7.1','8.1','10.1']
ruca_accept_UCL = ['4.0','5.0','7.2','8.2','10.2']
ruca_accept_UCS = ['7.0','8.0','10.3']

#####
##Main Loop##
#####

#Initialize loop counter for main loop
loop_count = 1;

#Create a cursor to go through each UAC sorted by population in decending order
create_layer(uac_file,"uac_lyr")
uac_cursor = arcpy.SearchCursor("uac_lyr","","","POP10 D")
```

```

#Loop through each UAC to create each EUC
for current_UAC in uac_cursor:

    #Initializations for current iteration of loop
    current_UAC_name = current_UAC.getValue(name_field)
    city_sql = ' "NAME10" = \'%s\' ' %(current_UAC_name) #Used w/ bg layer
    current_UAC_size = current_UAC.getValue(area_field)
    bg_cutoff_size = current_UAC_size * 5 #Size limit for including bgs - 5x
    size_sql = ' "ALAND10" > %s ' %(bg_cutoff_size)
    current_UAC_pop = current_UAC.getValue(pop_field)
    while_flag = 0 #Flag variable for exiting block while loop

    #Make new layers for block group and block files
    create_layer(bg_file,"bg_lyr")
    create_layer(blk_file,"blk_lyr")

    #####
    ##Select block groups from UAC to include in current EUC##
    #####

    #Select the current UAC from the UAC file
    arcpy.SelectLayerByAttribute_management("uac_lyr","NEW_SELECTION",city_sql)

    #Perform an intersection to include all BGs that intersect the UC,
    #-1 will make sure not to include ones that just have a coincident line
    arcpy.SelectLayerByLocation_management("bg_lyr","INTERSECT","uac_lyr",
        selection_type="NEW_SELECTION",search_distance="-1 Feet")

    #Remove any block groups that have no land area
    arcpy.SelectLayerByAttribute_management("bg_lyr","REMOVE_FROM_SELECTION",
        ' "ALAND10" = 0')

    #Remove block groups that at larger than cutoff amount of original UAC area
    arcpy.SelectLayerByAttribute_management("bg_lyr","REMOVE_FROM_SELECTION",
        size_sql)

    #Create a cursor to go through each UAC a second time
    create_layer(uac_file,"UAC_lyr2")
    uac_cursor_2 = arcpy.SearchCursor("UAC_lyr2")

    #Loop through each UAC and remove block groups that are within any other UAC
    for current_UAC_2 in uac_cursor_2:

        #Store current inner loop UAC name and FID
        current_UAC_2_name = current_UAC_2.getValue(name_field)
        current_UAC_2_fid = current_UAC_2.getValue("FID")

        if current_UAC_name <> current_UAC_2_name:

            #Make temporary UAC layer
            create_layer(uac_file,"UAC_lyr_temp")

            #Select only the current inner loop UAC from the file
            fid_uac2_sql = ' "FID" = %s' %(current_UAC_2_fid)
            arcpy.SelectLayerByAttribute_management("UAC_lyr_temp",
                "NEW_SELECTION",fid_uac2_sql)

            #Select all block groups that intersect the current inner loop UAC
            #and make a layer
            arcpy.Intersect_analysis(["bg_lyr","UAC_lyr_temp"],
                "UAC_BG_intersect","ALL",-1 FEET,"INPUT")
            arcpy.MakeFeatureLayer_management("UAC_BG_intersect.shp",
                "UAC_BG_intersect_lyr")

            #Remove all intersecting block groups from the current outer loop
            #bg selection
            arcpy.SelectLayerByLocation_management("bg_lyr","CONTAINS",
                "UAC_BG_intersect_lyr",selection_type="REMOVE_FROM_SELECTION")

```



```

        #Clean up temporary shapefiles and layers for inner loop
        delete_file("UAC_BG_intersect.shp","SHAPEFILE")
        delete_file("UAC_BG_intersect_lyr","LAYER")
        delete_file("UAC_lyr_temp","LAYER")

#Only run next loop if EUC file has been created
#(iteration 2 or beyond of main loop)
if loop_count <> 1:

    #Create a new EUC layer
    create_layer("EUC_All.shp","EUC_All_lyr")

    #Create a cursor to go through each EUC
    euc_cursor = arcpy.SearchCursor("EUC_All_lyr")

    #Loop through each EUC and remove bgs that are within any other EUC
    for current_EUC in euc_cursor:

        #Store current inner loop UAC FID
        current_EUC_fid = current_EUC.getValue("FID")

        #Create a temporary EUC layer
        create_layer("EUC_All.shp","EUC_All_lyr_temp")

        #Select only the current inner loop EUC from the file
        fid_euc_sql = ' "FID" = %s'%(current_EUC_fid)
        arcpy.SelectLayerByAttribute_management("EUC_All_lyr_temp",
            "NEW_SELECTION",fid_euc_sql)

        #Select all block groups that intersect the current inner loop EUC
        #and make a layer
        arcpy.Intersect_analysis(["bg_lyr","EUC_All_lyr_temp"],
            "EUC_BG_intersect","ALL",-1 FEET,"INPUT")
        arcpy.MakeFeatureLayer_management("EUC_BG_intersect.shp",
            "EUC_BG_intersect_lyr")

        #Remove all intersecting bgs from current outer loop bg selection
        arcpy.SelectLayerByLocation_management("bg_lyr","CONTAINS",
            "EUC_BG_intersect_lyr",selection_type="REMOVE_FROM_SELECTION")

        #Clean up temporary shapefiles and layers for inner loop
        delete_file("EUC_BG_intersect.shp","SHAPEFILE")
        delete_file("EUC_BG_intersect_lyr","LAYER")
        delete_file("EUC_All_lyr_temp","LAYER")

#Create an empty current EUC temp file and append all selected block groups
arcpy.Select_analysis("blk_lyr","current_EUC_temp",' "FID" = -1 ')
arcpy.Append_management("bg_lyr","current_EUC_temp.shp","NO_TEST")

#####
##Add "donut" block groups to the current EUC - completes BG selection##
#####
#An example is a block group that is surrounded by selected block groups
#that is not yet included in the EUC... it should be for continuity

#Select all block groups that border the current set and make this a layer
#The union function sets FIDs of all gap block groups to -1
arcpy.Union_analysis("current_EUC_temp.shp","bg_donuts",gaps="NO_GAPS")
create_layer("bg_donuts.shp","bg_donuts_lyr")

#Select all gap block groups from the donut layer
arcpy.SelectLayerByAttribute_management("bg_donuts_lyr","NEW_SELECTION",
    ' "FID_curren" = -1 ')

#Select all gap block groups from the block group layer
arcpy.SelectLayerByLocation_management("bg_lyr","WITHIN","bg_donuts_lyr",
    selection_type="NEW_SELECTION")

```

```

#Remove any block groups that have no area
arcpy.SelectLayerByAttribute_management("bg_lyr", "REMOVE_FROM_SELECTION",
    ' "ALAND10" = 0')

#Append gap block groups to current EUC temp file
arcpy.Append_management("bg_lyr", "current_EUC_temp.shp", "NO_TEST")

#####
##Add UAC intersecting blocks to the current EUC ##
#####
#Would occur if BG was removed due to intersection with another existing EUC

#Dissolve current EUC temp file and create layer for dissolved feature
arcpy.Dissolve_management("current_EUC_temp.shp", "current_EUC_temp_diss")
create_layer("current_EUC_temp_diss.shp", "current_EUC_temp_diss_lyr")

#Select only the current UAC from the UAC file
arcpy.SelectLayerByAttribute_management("uac_lyr", "NEW_SELECTION", city_sql)

#Select all blocks within current UAC and remove blocks within current EUC
arcpy.SelectLayerByLocation_management("blk_lyr", "WITHIN", "uac_lyr",
    selection_type="NEW_SELECTION")
arcpy.SelectLayerByLocation_management("blk_lyr", "WITHIN",
    "current_EUC_temp_diss_lyr", selection_type="REMOVE_FROM_SELECTION")

#Append all remaining blocks to the current EUC
arcpy.Append_management("blk_lyr", "current_EUC_temp.shp", "NO_TEST")

#Note: The following only works for tabblock files if the file has
#housing and population prejoined

#####
##Add outlying blocks to the current EUC ##
#####

#While there are still blocks to add, run through procedure to add blocks
while while_flag == 0:

    #####
    ##Select outlying blocks that contain 60 or more dwellings per mile##
    #####

    #Dissolve current EUC temp file and create layer for dissolved feature
    delete_file("current_EUC_temp_diss.shp", "SHAPEFILE")
    arcpy.Dissolve_management("current_EUC_temp.shp",
        "current_EUC_temp_diss")
    create_layer("current_EUC_temp_diss.shp", "current_EUC_temp_diss_lyr")

    #Select all blocks that border current EUC and remove those within EUC
    arcpy.SelectLayerByLocation_management("blk_lyr", "BOUNDARY_TOUCHES",
        "current_EUC_temp_diss_lyr", selection_type="NEW_SELECTION")
    arcpy.SelectLayerByLocation_management("blk_lyr", "WITHIN",
        "current_EUC_temp_diss_lyr", selection_type="REMOVE_FROM_SELECTION")

    #Remove all blocks with no land area or no houses
    arcpy.SelectLayerByAttribute_management("blk_lyr",
        "REMOVE_FROM_SELECTION", ' "ALAND10" = 0 OR "HOUSING10" = 0')

    #Remove all blocks that have under 60 dwellings per square mile
    remove_condition = "\"HOUSING10\"/(\"ALAND10\"**s) < 60\"%m2_to_mi2
    arcpy.SelectLayerByAttribute_management("blk_lyr",
        "REMOVE_FROM_SELECTION", remove_condition)

    #####
    ##Remove outlying blocks that do not contain at least 30 percent##
    ##of workers commuting to the UAC - Uses RUCA file and codes#####
    #####

```

```

#Create a cursor to go through each selected block
blk_cursor = arcpy.SearchCursor("blk_lyr")

#Decide which RUCA codes are acceptable based on population of UAC
#50,000+ is an Urbanized Area
#10,000-49,999 is a large Urban Cluster
#2,500-9,999 is a small Urban Cluster
if current_UAC_pop >= 50000:
    ruca_codes_accept = ruca_accept_UA
elif current_UAC_pop >=10000:
    ruca_codes_accept = ruca_accept_UCL
else:
    ruca_codes_accept = ruca_accept_UCS

#Loop through each selected block and remove if necessary
for blk_row in blk_cursor:

    #Store current blocks GeoID and parse out the tract ID
    current_geoid_blk = blk_row.getValue(geoid_field)
    current_geoid_tract = current_geoid_blk[:11]

    #Read in csv file (with statement makes sure file will close after
    #completion of this block)
    with open(ruca_file, 'rb') as ruca_csv: #Read in csv file

        #Store data array from csv file
        ruca_array = csv.reader(ruca_csv, delimiter=',')

        #Loop through each row of the data array file
        for ruca_row in ruca_array:

            #Reset flag used in RUCA code analysis
            #0 means unacceptable, 1 means acceptable
            ruca_flag = 0

            #Store the RUCA tract and RUCA code for the current row
            ruca_tract = ruca_row[3]
            ruca_code = ruca_row[5]

            #If current block's tract matches the current RUCA tract,
            #loop through each acceptable ruca code and check to see if
            #the current block's tract has an acceptable RUCA code
            if current_geoid_tract == ruca_tract:
                for acceptable_ruca in ruca_codes_accept:
                    if ruca_code == acceptable_ruca:
                        ruca_flag = 1

            #If flag never raised (no acceptable code found for
            #current block), then remove the block from selection
            if ruca_flag == 0:
                fid_blk = str(blk_row.getValue("FID"))
                fid_blk_sql = "FID = %s"%(fid_blk)
                arcpy.SelectLayerByAttribute_management("blk_lyr",
                    "REMOVE_FROM_SELECTION",fid_blk_sql)

#Create a cursor to go through each UAC a third time
create_layer(uac_file,"UAC_lyr3")
uac_cursor_3 = arcpy.SearchCursor("UAC_lyr3","","","","POP10 D")

#Loop through each UAC and remove blocks that are within any other UAC
for current_UAC_3 in uac_cursor_3:

    #Store current inner loop UAC name and FID
    current_UAC_3_name = current_UAC_3.getValue(name_field)

```

```

if current_UAC_name <> current_UAC_3_name:

    #Make temporary UAC layer
    create_layer(uac_file, "UAC_lyr_temp")

    #Select only the current inner loop UAC from the file
    name_uac3_sql = ' "NAME10" = \'%s\''%(current_UAC_3_name)
    arcpy.SelectLayerByAttribute_management("UAC_lyr_temp",
        "NEW_SELECTION",name_uac3_sql)

    #Remove all intersecting blocks from the current outer loop
    #block selection
    arcpy.SelectLayerByLocation_management("blk_lyr","WITHIN",
        "UAC_lyr_temp",selection_type="REMOVE_FROM_SELECTION")

#Only run next loop if EUC file has been created
#(iteration 2 or beyond of main loop)
if loop_count <> 1:

    #Create a new EUC layer
    create_layer("EUC_All.shp","EUC_All_lyr2")

    #Create a cursor to go through each EUC
    euc_cursor_2 = arcpy.SearchCursor("EUC_All_lyr2")

    #Loop through each EUC and remove blks that are within any other EUC
    for current_EUC_2 in euc_cursor_2:

        #Store current inner loop UAC FID
        current_EUC_2_fid = current_EUC_2.getValue("FID")

        #Create a temporary EUC layer
        create_layer("EUC_All.shp", "EUC_lyr_temp")

        #Select only the current inner loop EUC from the file
        fid_euc2_sql = ' "FID" = %s'%(current_EUC_2_fid)
        arcpy.SelectLayerByAttribute_management("EUC_lyr_temp",
            "NEW_SELECTION",fid_euc2_sql)

        #Remove all intersecting blks from current blk selection
        arcpy.SelectLayerByLocation_management("blk_lyr","WITHIN",
            "EUC_lyr_temp",selection_type="REMOVE_FROM_SELECTION")

    #If blocks are selected, add blocks to the current EUC temp file
    #Otherwise set while flag to kick out of loop next iteration
    blk_selected = int(arcpy.GetCount_management("blk_lyr").getOutput(0))
    if blk_selected <> 0:

        #Create a layer including all blocks to add
        arcpy.Select_analysis("blk_lyr","blks_add")
        create_layer("blks_add.shp","blks_add_lyr")

        #Create temporary EUC layer
        create_layer("current_EUC_temp.shp","current_EUC_temp_lyr")

        #Merge the blocks add layer with the current EUC and remake file
        arcpy.Merge_management(["blks_add_lyr","current_EUC_temp_lyr"],
            "current_EUC_merge")
        delete_file("current_EUC_temp.shp","SHAPEFILE")
        arcpy.Copy_management("current_EUC_merge.shp",
            "current_EUC_temp.shp")
        delete_file("current_EUC_merge.shp","SHAPEFILE")

    else:
        while_flag = 1

```

```
#####
##Add "donut" blks to current EUC - completes current blk selection##
#####

#Select all blocks that border the current set and make this a layer
#The union function sets FIDs of all gap blocks to -1
arcpy.Union_analysis("current_EUC_temp.shp","blk_donuts",gaps="NO_GAPS")
create_layer("blk_donuts.shp","blk_donuts_lyr")

#Select all gap blocks from the donut layer
arcpy.SelectLayerByAttribute_management("blk_donuts_lyr",
    "NEW_SELECTION",' "FID_curren" = -1 ')

#Select all gap block groups from the block group and block layers
arcpy.SelectLayerByLocation_management("bg_lyr","WITHIN",
    "blk_donuts_lyr",selection_type="NEW_SELECTION")
arcpy.SelectLayerByLocation_management("blk_lyr","WITHIN",
    "blk_donuts_lyr",selection_type="NEW_SELECTION")

#Remove any blocks and block groups that have no area
arcpy.SelectLayerByAttribute_management("bg_lyr",
    "REMOVE_FROM_SELECTION",' "ALAND10" = 0')
arcpy.SelectLayerByAttribute_management("blk_lyr",
    "REMOVE_FROM_SELECTION",' "ALAND10" = 0')

#Create a temporary block group and block shapefile and layer
arcpy.Select_analysis("bg_lyr", "bg_donuts_temp")
create_layer("bg_donuts_temp.shp","bg_donuts_temp_lyr")
arcpy.Select_analysis("blk_lyr", "blk_donuts_temp")
create_layer("blk_donuts_temp.shp","blk_donuts_temp_lyr")

#Merge the blks and bgs with the current EUC and remake file
arcpy.Merge_management(["blk_donuts_temp_lyr","bg_donuts_temp_lyr",
    "current_EUC_temp.shp"],"current_EUC_merge")
delete_file("current_EUC_temp.shp","SHAPEFILE")
arcpy.Copy_management("current_EUC_merge.shp","current_EUC_temp.shp")

#Clean up temporary shapefiles and layers for while loop
delete_file("blks_add.shp","SHAPEFILE")
delete_file("current_EUC_merge.shp","SHAPEFILE")
delete_file("bg_donuts_temp.shp","SHAPEFILE")
delete_file("blk_donuts_temp.shp","SHAPEFILE")
delete_file("current_EUC_merge.shp","SHAPEFILE")
delete_file("blk_donuts.shp","SHAPEFILE")

#####
##Create final dissolved EUC shapefile##
#####

#If first time through loop, create new EUC file
if loop_count == 1:

    #Clean up old files
    delete_file("EUC_All.shp","SHAPEFILE")

    #Create a new EUC file (dissolved) with the first EUC
    arcpy.Dissolve_management("current_EUC_temp.shp","EUC_All")

    #Add in a name field to the file
    arcpy.AddField_management("EUC_All.shp","EUC_Name","TEXT","","",128)

    #Create an EUC layer
    create_layer("EUC_All.shp","EUC_All_lyr")

    #Create a dummy cursor and use it to loop through EUCs
    #Will actually only loop once to set current EUC name
    dummy_cursor = arcpy.UpdateCursor("EUC_ALL_lyr")
```

```

        for dummy_value in dummy_cursor:
            dummy_value.setValue("EUC_Name",current_UAC_name)
            dummy_cursor.updateRow(dummy_value)

#If EUC All file has already been created
#(iteration 2 or beyond of main loop)
else:
    #Create a new EUC file (dissolved) with the current EUC
    arcpy.Dissolve_management("current_EUC_temp.shp","current_EUC")

    #Add in a name field to the file
    arcpy.AddField_management("current_EUC.shp","EUC_Name","TEXT","","",128)

    #Create an EUC layer for current EUC
    create_layer("current_EUC.shp","current_EUC_lyr")

    #Create a dummy cursor and use it to loop through EUCs
    #Will actually only loop once to set current EUC name
    dummy_cursor = arcpy.UpdateCursor("current_EUC_lyr")
    for dummy_value in dummy_cursor:
        dummy_value.setValue("EUC_Name",current_UAC_name)
        dummy_cursor.updateRow(dummy_value)

    #Append the current EUC to the EUC All file and delete the current EUC
    arcpy.Append_management("current_EUC.shp","EUC_All.shp")
    delete_file("current_EUC.shp","SHAPEFILE")

#####
##File cleanup for outer loop##
#####

delete_file("bg_lyr","LAYER")
delete_file("blk_lyr","LAYER")
delete_file("UAC_lyr2","LAYER")
delete_file("UAC_BG_intersect.shp","SHAPEFILE")
delete_file("UAC_BG_intersect_lyr","LAYER")
delete_file("UAC_lyr_temp","LAYER")
delete_file("EUC_All_lyr","LAYER")
delete_file("EUC_BG_intersect.shp","SHAPEFILE")
delete_file("EUC_BG_intersect_lyr","LAYER")
delete_file("EUC_All_lyr_temp","LAYER")
delete_file("current_EUC_temp.shp","SHAPEFILE")
delete_file("current_EUC_temp_lyr","LAYER")
delete_file("bg_donuts.shp","SHAPEFILE")
delete_file("bg_donuts_lyr","LAYER")
delete_file("current_EUC_temp_diss.shp","SHAPEFILE")
delete_file("current_EUC_temp_diss_lyr","LAYER")
delete_file("UAC_lyr3","LAYER")
delete_file("EUC_All_lyr2","LAYER")
delete_file("blks_add.shp","SHAPEFILE")
delete_file("blks_add_lyr","LAYER")
delete_file("current_EUC_merge.shp","SHAPEFILE")
delete_file("blk_donuts.shp","SHAPEFILE")
delete_file("blk_donuts_lyr","LAYER")
delete_file("bg_donuts_temp.shp","SHAPEFILE")
delete_file("bg_donuts_temp_lyr","LAYER")
delete_file("blk_donuts_temp.shp","SHAPEFILE")
delete_file("blk_donuts_temp_lyr","LAYER")
delete_file("current_EUC.shp","SHAPEFILE")
delete_file("current_EUC_lyr","LAYER")

#Increase loop counter for next iteration
loop_count = loop_count + 1

#Final file cleanup
delete_file("uac_lyr","LAYER")

```

Appendix C: EUC_Metrics.py – Full Python Script

```
#####EUC_Metrics.py#####
##This program appends 21 metrics to the Extended Urban Clusters    ##
##shapefile which was created using EUC_Creator.py.  The approach used to##
##calculate each metric is outlined in the dissertation Chapter 6, but    ##
##there should be enough comments here to follow along                ##
#####

##Basic Notes for running the program
#1 Python editor must be exited after running program for EUC file to
#   update properly
#2 The following pre-processing steps must be done before running program
#   A Jobs data (from LEHD LODES7 WAC) must be merged with the block file
#   B WalkScore_Locator.py must be run and the resulting Walk Scores column
#     must be merged with the block file
#   C NLCD data must be merged with the EUC file using NLCD_table.dbf
#     (described in section 6.3.1 of the dissertation)
#   D A CBD points file must be created (described in section 6.3.1 of the
#     dissertation)
#   E A CBD polygon file must be created by running CBD_polygon_creator.py
#     and completing some additional steps as described in section 6.3.2
#     of the dissertation
#   F An intersections file must be obtained from University of Utah's
#     Metropolitan Research Center (MRC) or developed in some other manner
#     They clipped their file to Riehl's EUC file
#3 There are three additional metrics calculated (prefixed EX).  These
#   can be ignored as these are just alternates to the EW metrics
#4 Notes for next version update
#   A popden - not normalized
#   B lt1500, gt12500 - should these be normalized to population?
#   C urbden - should be changed to be population density, not an area
#     percentage
#   D dgcent, popdcen, empdcen, dgrad, popcen, empcen are set to -1 for
#     the 5 EUCs that do not have their CBD in Michigan
#   E avgbklng - Algorithm uses slightly different dataset and slightly
#     different method form Ewing et al.

#Import Libraries and checkout extensions
import arcpy
import csv
import numpy
import math
import urllib2

arcpy.CheckOutExtension("Spatial") #Check out spatial visualization license

#####
##Initializations##
#####

#Initialize folder and file locations
arcpy.env.workspace = "H:\PHD\EUC_Metrics" #Set location of working folder
EUC_file = "EUC_All.shp"
blk_file = "tl_2010_26_tabbblock10.shp"
CBD_point_file_short = "CBD_Points_UA_Named_City_Only"
CBD_point_file = CBD_point_file_short + ".shp"
CBD_poly_file = "CBD_polygons_all.shp"
NLCD_file = "nlcd_2011.img"
NLCD_table_file = "NLCD_table.dbf"
road_file = "allroads_miv14a.shp"
intersection_file = "intersections_MI_EUC.shp"
```

```

#Initialize names that will be used throughout the program
name_field = "EUC_Name" #Set name of name field to be used in the EUC layer
employ_field = "C000"
retail_field = "CNS07"
info_field = "CNS09"
finance_field = "CNS10"
realest_field = "CNS11"
techserve_field = "CNS12"
edu_field = "CNS15"
health_field = "CNS16"
entertain_field = "CNS17"
hotelfood_field = "CNS18"
othservices_field = "CNS19"
ws_field = "WalkScore"

#Initialize constants that will be used throughout the program
m2_to_mi2 = pow(100,2)/pow(2.54,2)/pow(12,2)/pow(5280,2)
R = 3959 #radius of the earth in miles

#####
##Create New Fields##
#####

#Add in some fields for base metrics
arcpy.AddField_management("EUC_All.shp", "pop", "DOUBLE")
arcpy.AddField_management("EUC_All.shp", "CBD_x", "DOUBLE")
arcpy.AddField_management("EUC_All.shp", "CBD_y", "DOUBLE")

#Add in fields for Ewing's metrics (EW = Ewing et al.)
# (EX = Additional ones) to EUC file
#Could add in a check for each field to see if it exists first
#Labels (e.g., D1, are used throughout the program to refer to sections
# that are used to help calculate the labeled metric

#D1 - Gross population density of all census blocks in EUC in persons per
# square mile
arcpy.AddField_management("EUC_All.shp", "EWpopden", "DOUBLE")

#D2 - Gross employment density of all census blocks in EUC in jobs per
# square mile
arcpy.AddField_management("EUC_All.shp", "EWempden", "DOUBLE")

#D3 - percentage of the population living at low suburban densities
# (less than 1,500 persons per square mile)
arcpy.AddField_management("EUC_All.shp", "EWlt1500", "DOUBLE")

#D4 - percentage of the population living at medium to high urban densities
# (greater than 12,500 persons per square mile - transit-supportive)
arcpy.AddField_management("EUC_All.shp", "EWgt12500", "DOUBLE")

#D5 - gross population density of urban lands - currently this is actually
# the percentage of urban area to total area
arcpy.AddField_management("EUC_All.shp", "EWurbden", "DOUBLE")

#D6 - estimated density at the center of the metro area derived from a
# negative exponential density function
arcpy.AddField_management("EUC_All.shp", "EWdgcent", "DOUBLE")

#D7 - Gross population density of CBD in EUC in persons per square mile
# (not weighted)
arcpy.AddField_management("EUC_All.shp", "EWpopdcen", "DOUBLE")

#D8 - Gross employment density of CBD in EUC in jobs per square mile
# (not weighted)
arcpy.AddField_management("EUC_All.shp", "EWempdcen", "DOUBLE")

```



```

#M1 - job-population balance (using blocks instead of 1-mile ring block
#       groups)
arcpy.AddField_management("EUC_All.shp", "EWjobpop", "DOUBLE")

#M2 - degree of job mixing (entropy) (using blocks instead of 1-mile
#       ring block groups)
arcpy.AddField_management("EUC_All.shp", "EWjobmix", "DOUBLE")

#M3 - weighted average Walk Score
arcpy.AddField_management("EUC_All.shp", "EWwalkscor", "DOUBLE")

#C1- coefficient of variation in census block population densities
arcpy.AddField_management("EUC_All.shp", "EWvarpop", "DOUBLE")

#C2- coefficient of variation in census block employment densities
arcpy.AddField_management("EUC_All.shp", "EWvaremp", "DOUBLE")

#C3 - density gradient moving outward from the CBD
arcpy.AddField_management("EUC_All.shp", "EWdgrad", "DOUBLE")

#C4 - percentage of EUC population in CBD (using EUC instead of MSA
#       and CBD only without subcenters)
arcpy.AddField_management("EUC_All.shp", "EWpopcen", "DOUBLE")

#C5 - percentage of EUC employment in CBD (using EUC instead of MSA
#       and CBD only without subcenters)
arcpy.AddField_management("EUC_All.shp", "EWempcen", "DOUBLE")

#S1 - percentage of small urban blocks
arcpy.AddField_management("EUC_All.shp", "EWsmlblk", "DOUBLE")

#S2 - average block size
arcpy.AddField_management("EUC_All.shp", "EWavgblksz", "DOUBLE")

#S3 - average block length
arcpy.AddField_management("EUC_All.shp", "EWavgblkln", "DOUBLE")

#S4 - intersection density - removes blocks with < 50 popden
#       (half of EW's suggested 100)
arcpy.AddField_management("EUC_All.shp", "EWintden", "DOUBLE")

#S5 - percentage of 4-or-more-way intersections
arcpy.AddField_management("EUC_All.shp", "EW4way", "DOUBLE")

#S1a - percentage of small urban blocks without removing large blocks
arcpy.AddField_management("EUC_All.shp", "EXsmlblk", "DOUBLE")

#S2a - average block size without removing large blocks
arcpy.AddField_management("EUC_All.shp", "EXavgblksz", "DOUBLE")

#S4a - intersection density - does not removes blocks with < 50 popden
arcpy.AddField_management("EUC_All.shp", "EXintden", "DOUBLE")

#####
##Make Layers for Processing##
#####

#Make EUC a layer
if arcpy.Exists("EUC_lyr") == 1:
    arcpy.Delete_management("EUC_lyr", "LAYER")
arcpy.MakeFeatureLayer_management(EUC_file, "EUC_lyr")

#Make CBD points a layer
if arcpy.Exists("CBD_point_lyr") == 1:
    arcpy.Delete_management("CBD_point_lyr", "LAYER")
arcpy.MakeFeatureLayer_management(CBD_point_file, "CBD_point_lyr")

```

```

#Make CBD polygons a layer
if arcpy.Exists("CBD_poly_lyr") == 1:
    arcpy.Delete_management("CBD_poly_lyr", "LAYER")
arcpy.MakeFeatureLayer_management(CBD_poly_file, "CBD_poly_lyr")

#Make intersections a layer
if arcpy.Exists("intersection_lyr") == 1:
    arcpy.Delete_management("intersection_lyr", "LAYER")
arcpy.MakeFeatureLayer_management(intersection_file, "intersection_lyr")

#####
##Join CBD points to EUC file##
#####

# Join the EUC and CBD Shapefiles
arcpy.AddJoin_management("EUC_lyr", "EUC_Name", "CBD_point_lyr", "Name")

#Setup CBD sql statements for CalculateField function
CBD_sql_x = "[%s.POINT_X]"%(CBD_point_file_short)
CBD_sql_y = "[%s.POINT_Y]"%(CBD_point_file_short)

#Copy CBD x and CBD y points
arcpy.CalculateField_management("EUC_lyr", "EUC_All.CBD_x", CBD_sql_x)
arcpy.CalculateField_management("EUC_lyr", "EUC_All.CBD_y", CBD_sql_y)

#Clean up join
arcpy.RemoveJoin_management("EUC_lyr", CBD_point_file_short)

#####
##Main Loop - for each EUC##
#####

#Create a cursor to go through each EUC and loop through each EUC
EUC_cursor = arcpy.UpdateCursor("EUC_lyr")
for current_EUC in EUC_cursor:

    #####
    ##Initializations for current iteration of loop##
    #####

    #Initializations needed for many metrics
    sum_aland = 0.0
    sum_pop = 0.0
    sum_emp = 0.0

    #Get values from current EUC
    current_EUC_name = current_EUC.getValue(name_field)
    current_EUC_cbd_x_coord = current_EUC.getValue("CBD_x")
    current_EUC_cbd_y_coord = current_EUC.getValue("CBD_y")
    current_EUC_developed_land = current_EUC.getValue("Devel_Lnd") #D5

    #Convert EUC CBD coords to radians
    lon_EUC_r = math.radians(current_EUC_cbd_x_coord)
    lat_EUC_r = math.radians(current_EUC_cbd_y_coord)

    #Initializations needed for specific metrics (labeled)
    blk_count = 0.0 #D3/D4/S1a/S2a
    lt1500_count = 0.0 #D3
    gt12500_count = 0.0 #D4
    d = numpy.array([]) #D6/C3
    D = numpy.array([]) #D6/C3
    jobpop = 0.0 #M1
    J = numpy.array([]) #M1/M2
    P = numpy.array([]) #M1/M2
    TJ = 0.0 #M1/M2
    TP = 0.0 #M1/M2
    jobmix_j_vec = numpy.array([]) #M2
    jobmix = 0.0 #M2

```

```

ws_sum = 0.0 #M3
popden_vec = numpy.array([]) #C1
empden_vec = numpy.array([]) #C2
normal_blk_count = 0.0 #S1/S2
sml_blk_count = 0.0 #S2
sum_aland_normal_block = 0.0 #S2
sum_road_len = 0.0 #S3
road_count = 0.0 #S3
sum_aland_high_density = 0.0 #S4

#Make a fresh block layer
if arcpy.Exists("blk_lyr") == 1:
    arcpy.Delete_management("blk_lyr", "LAYER")
arcpy.MakeFeatureLayer_management(blk_file, "blk_lyr")

#Select only blocks in the current EUC by selecting the current EUC and
# then the blocks within the EUC
euc_select_sql = ' "EUC_Name" = \'%s\' ' %(current_EUC_name)
arcpy.SelectLayerByAttribute_management("EUC_lyr", "NEW_SELECTION",
    euc_select_sql)
arcpy.SelectLayerByLocation_management("blk_lyr", "WITHIN", "EUC_lyr",
    selection_type="NEW_SELECTION")

#####
##Loop 2 - for each block in the EUC##
#####

#Create a cursor to go through each block in the EUC
blk_cursor = arcpy.SearchCursor("blk_lyr")
for current_blk in blk_cursor:

    #####
    ##Initializations for current iteration of loop 2##
    #####

    #Get and store values from current block
    blk_aland = current_blk.getValue("ALAND10")*m2_to_mi2
    blk_pop = current_blk.getValue("POP10")
    blk_employ = current_blk.getValue(employ_field)
    blk_x_coord = float(current_blk.getValue("INTPTLON10"))
    blk_y_coord = float(current_blk.getValue("INTPTLAT10"))

    #Calculate the population and employment densities; 0 if no land
    if blk_aland > 0:
        blk_popden = blk_pop / blk_aland
        blk_empden = blk_employ / blk_aland
    else:
        blk_popden = 0.0 #might need to change to -1
        blk_empden = 0.0

    #Convert block longitude and latitude to radians
    lon_blk_r = math.radians(blk_x_coord)
    lat_blk_r = math.radians(blk_y_coord)

    #Sum land area, population, and employment for all blocks in EUC
    sum_aland += blk_aland
    sum_pop += blk_pop
    sum_employ += blk_employ

    #####
    ##Specific Metric Calculations##
    #####

    ##D3/D4/S1a/S2a##
    #increment block counter
    blk_count += 1.0

```

```

##D3 - 1/2##
#count if density less than 1500 for lt1500 category
if blk_popden < 1500 and blk_popden >= 0:
    lt1500_count += 1.0

##D4 - 1/2##
#count if density greater than 12500 for gt12500 category
if blk_popden > 12500:
    gt12500_count += 1.0

##D6/C3 - 1/2##
#If CBD exists, calculate D and d for equation  $D = D_0 * e^{(-b*d)}$ 
if current_EUC_cbd_x_coord <> 0:

    #Calculate difference in lat/lon between blk and EUC
    dlat = lat_blk_r - lat_EUC_r
    dlon = lon_blk_r - lon_EUC_r

    #Calculate current d value using Haversine formula
    a = math.pow(((math.sin(dlat/2)),2) + math.cos(lat_blk_r)
        * math.cos(lat_EUC_r) * math.pow((math.sin(dlon/2)),2))
    c = 2 * math.atan2( math.sqrt(a), math.sqrt(1-a) )
    di = R * c #(where R is the radius of the Earth)

    #Append current distance and current density to d/D lists
    d = numpy.append(d,di)
    D = numpy.append(D,blk_popden)

##M1/M2 - 1/2##
#Vectors created here, calculated after the loop due to need for
#EUC densities (not calculated until after loop runs through all
#blocks in EUC)
if blk_popden >= 100:

    #Append employment and population values onto lists
    J = numpy.append(J,blk_employ)
    P = numpy.append(P,blk_pop)

    #Add employment and population to total counts
    TJ = TJ + blk_employ
    TP = TP + blk_pop

    ##M2##
    num_sectors = 5.0

    #Store all job counts for current block
    jobs_retail = current_blk.getValue(retail_field)
    jobs_entertainment = current_blk.getValue(entertain_field)
    jobs_health = current_blk.getValue(health_field)
    jobs_education = current_blk.getValue(edu_field)
    jobs_personalservices = (current_blk.getValue(info_field) +
        current_blk.getValue(finance_field) +
        current_blk.getValue(realest_field) +
        current_blk.getValue(techserve_field) +
        current_blk.getValue(hotelfood_field) +
        current_blk.getValue(othservices_field))
    jobs_5_total = (jobs_retail + jobs_entertainment +
        jobs_health + jobs_education + jobs_personalservices)

    #As long as jobs exist, run jobmix calculations
    if jobs_5_total <> 0:

        #Calculate percentages for all five job categories
        P_retail = 100.0 * jobs_retail / jobs_5_total
        P_entertainment = 100.0 * jobs_entertainment / jobs_5_total
        P_health = 100.0 * jobs_health / jobs_5_total
        P_education = 100.0 * jobs_education / jobs_5_total

```

```

P_personalservices = (100.0 * jobs_personalservices /
jobs_5_total)

#As long as percentages are not 0 calculate mixing factor
#Otherwise, set it to 0
if P_retail <> 0:
    jobmix_blk_retail = (P_retail * math.log(P_retail) /
        math.log(num_sectors))
else:
    jobmix_blk_retail = 0.0
if P_entertainment <> 0:
    jobmix_blk_entertainment = (P_entertainment *
        math.log(P_entertainment) / math.log(num_sectors))
else:
    jobmix_blk_entertainment = 0.0
if P_health <> 0:
    jobmix_blk_health = (P_health * math.log(P_health) /
        math.log(num_sectors))
else:
    jobmix_blk_health = 0.0
if P_education <> 0:
    jobmix_blk_education = (P_education *
        math.log(P_education) / math.log(num_sectors))
else:
    jobmix_blk_education = 0.0
if P_personalservices <> 0:
    jobmix_blk_personalservices = (P_personalservices *
        math.log(P_personalservices) /
        math.log(num_sectors))
else:
    jobmix_blk_personalservices = 0.0

#Sum mix factors
jobmix_j = (jobmix_blk_retail + jobmix_blk_entertainment +
    jobmix_blk_health + jobmix_blk_education +
    jobmix_blk_personalservices)
else:
    jobmix_j = 0

#Append current job mix factor to job mix factor list
jobmix_j_vec = numpy.append(jobmix_j_vec,jobmix_j)

##M3 - 1/2##
#Note: Currently WalkScore tabulation occurs in WalkScoreLocator.py
#Get block walkscore, weight it based on population and employment,
# and add to EUC WalkScore sum
walkscore_raw = current_blk.getValue(ws_field)
walkscore_weighted = walkscore_raw * (blk_pop + blk_employ)
ws_sum = ws_sum + walkscore_weighted

##C1 - 1/2##
popden_vec = numpy.append(popden_vec,blk_popden)

##C2 - 1/2##
empden_vec = numpy.append(empden_vec,blk_empden)

##S1/S2 - 1/2##
#count small blocks if land area less than 0.01
# normal blocks if land area less than 1
# and sum area of normal blocks
if blk_aland < 0.01:
    sml_blk_count += 1.0
if blk_aland <= 1:
    normal_blk_count += 1.0
    sum_aland_normal_block += blk_aland

```

```

##S4 - 1/2##
#Sum area of all blocks that are high density (greater than 50)
if blk_popden >= 50:
    sum_aland_high_density += blk_aland

#####
##End Loop 2##
#####

##D1 - 1/1##
#Calculate overall population density for entire EUC
EWpopden = sum_pop / sum_aland

##D2 - 1/1##
#Calculate overall employment density for entire EUC
EWempden = sum_emp / sum_aland

##D3 - 2/2##
EWlt1500 = 100.0 * lt1500_count / blk_count

##D4 - 2/2##
EWgt12500 = 100.0 * gt12500_count / blk_count

##D5 - 1/1##
EWurbden = 100.0 * current_EUC_developed_land * m2_to_mi2 / sum_aland

##D6/C3 - 2/2##
#If CBD exists, calculate D and d in equation  $D = D_0 \cdot e^{(-b \cdot d)}$ 
if current_EUC_cbd_x_coord <> 0: #Make sure CBD exists

    #Take the natural log of d and D values
    for i in range(0,len(d),1):
        if (d[i] and D[i]) <> 0:
            d[i] = math.log(d[i]) #should not take log of this?
            D[i] = math.log(D[i])

    #Fit a polynomial to D versus d; equation is  $D = D_{cent} \cdot \exp(-grad \cdot d)$ 
    coefficients = numpy.polyfit(d,D,1)
    grad = coefficients[0]
    #2nd argument comes back as log(Dcent), exp removes log
    Dcent = math.exp(coefficients[1])
    EWdgcen = Dcent
    EWdgrad = -grad
else:
    EWdgcen = -1 #Store as -1 if there is no CBD in Michigan
    EWdgrad = -1

##D7/D8/C4/C5 Calculations - 1/1##
#Initializations for these calculations
sum_aland_cbd = 0.0
sum_pop_cbd = 0.0
sum_emp_cbd = 0.0
sum_pop_cbd_weighted = 0.0
sum_emp_cbd_weighted = 0.0

#If the CBD exists, run calculations for D7/D8/C4/C5
if current_EUC_cbd_x_coord <> 0: #Make sure CBD exists

    #Create a 2nd block layer used for only CBD blocks
    if arcpy.Exists("blk_cbd_lyr") == 1:
        arcpy.Delete_management("blk_cbd_lyr", "LAYER")
        arcpy.MakeFeatureLayer_management(blk_file, "blk_cbd_lyr")

    #Select only blocks in the current CBD by selecting the current CBD
    # and then the blocks within the CBD
    cbd_select_sql = ' "Name" = \'' + current_EUC_name + '\''
    arcpy.SelectLayerByAttribute_management("CBD_poly_lyr",
        "NEW_SELECTION",cbd_select_sql)

```

```

arcpy.SelectLayerByLocation_management("blk_cbd_lyr","WITHIN",
    "CBD_poly_lyr",selection_type="NEW_SELECTION")

#Create a cursor to go through each block in the CBD
blk_cursor_cbd = arcpy.SearchCursor("blk_cbd_lyr")
for current_blk_cbd in blk_cursor_cbd:

    #Get and store values from current block
    blk_aland_cbd = current_blk_cbd.getValue("ALAND10")*m2_to_mi2
    blk_pop_cbd = current_blk_cbd.getValue("POP10")
    blk_employ_cbd = current_blk_cbd.getValue(employ_field)

    #Add values from current block to CBD sums
    sum_aland_cbd += blk_aland_cbd
    sum_pop_cbd += blk_pop_cbd
    sum_employ_cbd += blk_employ_cbd
    sum_pop_cbd_weighted += (blk_pop_cbd * (blk_pop_cbd +
        blk_employ_cbd))
    sum_employ_cbd_weighted += (blk_employ_cbd * (blk_pop_cbd +
        blk_employ_cbd))

    #As long as sum not 0, calculate D7/D8 values, otherwise set to 0
    if sum_aland_cbd <> 0:
        EWpopdcen = sum_pop_cbd_weighted / (sum_aland_cbd *
            (sum_pop + sum_employ))
        EWempdcen = sum_employ_cbd_weighted / (sum_aland_cbd *
            (sum_pop + sum_employ))
    elif sum_cbd_aland == 0:
        EWpopdcen = 0.0
        EWempdcen = 0.0

    #Calculate C4/C5 values
    EWpopcen = 100.0 * sum_pop_cbd / sum_pop
    EWempcen = 100.0 * sum_employ_cbd / sum_employ

#If no cbd's in Michigan, set D7/D8/C4/C5 values to -1
else:
    EWpopdcen = -1
    EWempdcen = -1
    EWpopcen = -1
    EWempcen = -1

##M1/M2 - 2/2##
#Calculate Job to Population Ratio
JP = TJ / TP

#Loop through all values in J vector and calculate jobpop/jobmix values
#Based on Ewing et al. equations
for i in range(0,len(J),1):
    jobpop_blk = ((1 - abs(J[i] - JP * P[i]) / (J[i] + JP * P[i])) *
        (J[i] + P[i]) / (TJ + TP))
    jobpop += jobpop_blk
    jobmix_blk = jobmix_j_vec[i] * (J[i] + P[i]) / (TJ + TP)
    jobmix += jobmix_blk
EWjobpop = jobpop
EWjobmix = jobmix

##M3 - 2/2##
EWwalkscor = ws_sum / (sum_pop + sum_employ)

##C1 - 2/2##
EWvarpop = numpy.std(popden_vec)/numpy.mean(popden_vec)

##C2 - 2/2##
EWvaremp = numpy.std(empden_vec)/numpy.mean(empden_vec)

```

```

##S1/S2a - 2/2##
EWsmlblk = 100.0 * sml_blk_count / normal_blk_count
EXsmlblk = 100.0 * sml_blk_count / blk_count

##S2/S2a - 2/2##
EWavgblkksz = sum_aland_normal_block / normal_blk_count
EXavgblkksz = sum_aland / blk_count

##S3 - 1/1##
#Create a road layer
if arcpy.Exists("road_lyr") == 1:
    arcpy.Delete_management("road_lyr","LAYER")
arcpy.MakeFeatureLayer_management(road_file, "road_lyr")

#Select all roads in the current EUC (EUC selection done above loop)
arcpy.SelectLayerByLocation_management("road_lyr","WITHIN","EUC_lyr",
    selection_type="NEW_SELECTION",search_distance="1 Meters")

#Create a cursor to go through each road segment in the EUC
road_cursor = arcpy.SearchCursor("road_lyr")
for current_road in road_cursor:

    #Store length of current road segment
    road_len = current_road.getValue("LENGTH")

    #Sum road segment lengths and road count
    sum_road_len = sum_road_len + road_len
    road_count = road_count + 1.0

#Calculate average road length which is the average block length
EWavgblkln = sum_road_len / road_count

##S4 - 2/2 and S5/S4a - 1/1##
#Select all intersections within EUC and count the number
# of intersections selected for S5 and S4a
arcpy.SelectLayerByLocation_management("intersection_lyr","INTERSECT",
    "blk_lyr",selection_type = "NEW_SELECTION")
intersections_total = len(
    arcpy.Describe("intersection_lyr").FIDSet.split('; '))

#Select all 4-or-more-way intersections for S5
arcpy.SelectLayerByAttribute_management("intersection_lyr",
    "SUBSET_SELECTION",' "ICOUNT" >= 4')
intersections_4way = len(
    arcpy.Describe("intersection_lyr").FIDSet.split('; '))

#Select high density blocks, select all intersections located within
# these blocks, and then count the number of high density intersections
# for S4
arcpy.SelectLayerByAttribute_management("blk_lyr","SUBSET_SELECTION",
    ' "POPDEN" >= 50')
arcpy.SelectLayerByLocation_management("intersection_lyr","INTERSECT",
    "blk_lyr",selection_type = "NEW_SELECTION")
intersections_highdensity = len(
    arcpy.Describe("intersection_lyr").FIDSet.split('; '))

#S4 Final Calculation
EWintden = intersections_highdensity / sum_aland_high_density

#S5 Final Calculation
EW4way = 100.0 * intersections_4way / intersections_total

#S4a Final Calculation
EXintden = intersections_total / sum_aland

```



```

#####
##Store all metrics for the current EUC##
#####

current_EUC.setValue("pop",sum_pop)

current_EUC.setValue("EWpopden",EWpopden) #D1
current_EUC.setValue("EWempden",EWempden) #D2
current_EUC.setValue("EWlt1500",EWlt1500) #D3
current_EUC.setValue("EWgt12500",EWgt12500) #D4
current_EUC.setValue("EWurbden",EWurbden) #D5
current_EUC.setValue("EWdgcent",EWdgcent) #D6
current_EUC.setValue("EWpopdcen",EWpopdcen) #D7
current_EUC.setValue("EWempdcen",EWempdcen) #D8

current_EUC.setValue("EWjobpop",EWjobpop) #M1
current_EUC.setValue("EWjobmix",EWjobmix) #M2
current_EUC.setValue("EWwalkscor",EWwalkscor) #M3

current_EUC.setValue("EWvarpop",EWvarpop) #C1
current_EUC.setValue("EWvaremp",EWvaremp) #C2
current_EUC.setValue("EWdgrad",EWdgrad) #C3
current_EUC.setValue("EWpopcen",EWpopcen) #C4
current_EUC.setValue("EWempcen",EWempcen) #C5

current_EUC.setValue("EWsmlblk",EWsmlblk) #S1
current_EUC.setValue("EWavgblksh",EWavgblksh) #S2
current_EUC.setValue("EWavgblkln",EWavgblkln) #S3
current_EUC.setValue("EWintden",EWintden) #S4
current_EUC.setValue("EW4way",EW4way) #S5

current_EUC.setValue("EXsmlblk",EXsmlblk) #S1a
current_EUC.setValue("EXavgblksh",EXavgblksh) #S2a
current_EUC.setValue("EXintden",EXintden) #S4a

#Update the EUC row
EUC_cursor.updateRow(current_EUC)

#####
##End Main Loop##
#####

```

Appendix D: CBD_Creator.py – Full Python Script

```
#####CBD_Creator.py#####
##This program creates CBDs for all of Michigan's EUC's principal cities ##
##This program should be run on the shapefile which was created using ##
##EUC_Creator.py. This program should be executed before EUC_Metrics.py ##
##The approach used to create these polygons is outlined in the section ##
##6.3.2, but there should be enough comments here to follow along. ##
##The output of this program is the file "CBD_polygons.shp." Currently ##
##this file does not create all CBDs so post processing must be done to ##
##complete this and prepare "CBD_polygons_all.shp" for EUC_Metrics.py. ##
#####

#Import Libraries
import arcpy
import os

#####
##Initializations##
#####

#Set location of working folder
home_folder = "H:\PHD\Metrics"
arcpy.env.workspace = home_folder

#Initialize folder and file locations
EUC_file = "EUC_All.shp"
CBD_points_file = "CBD_Points_UA_Named_City_Only.shp"
blk_file = "t1_2010_26_tabblock10.shp"

#Initialize names that will be used throughout the program
employ_field = "C000"
name_field = "EUC_Name"

#Initialize loop counter - because first loop needs to be distinguished
loop_count = 1

#####
##Make Layers for Processing##
#####

#Make EUC a layer
if arcpy.Exists("EUC_lyr") == 1:
    arcpy.Delete_management("EUC_lyr", "LAYER")
arcpy.MakeFeatureLayer_management(EUC_file, "EUC_lyr")

#Make block a layer
if arcpy.Exists("blk_lyr") == 1:
    arcpy.Delete_management("blk_lyr", "LAYER")
arcpy.MakeFeatureLayer_management(blk_file, "blk_lyr")

#Make CBD points a layer
if arcpy.Exists("CBD_Points_lyr") == 1:
    arcpy.Delete_management("CBD_Points_lyr", "LAYER")
arcpy.MakeFeatureLayer_management(CBD_points_file, "CBD_Points_lyr")

#####
##Main Loop - for each EUC##
#####

#Create a cursor to go through each EUC and loop through each EUC
EUC_cursor = arcpy.UpdateCursor("EUC_lyr")
for current_EUC in EUC_cursor:

    #Store name for current EUC
    current_EUC_name = current_EUC.getValue(name_field)
```

```

#Do not create CBD Polygons for five EUCs that have primary cities
# outside of Michigan
if (current_EUC_name <> "Toledo, OH--MI" and
    current_EUC_name <> "South Bend, IN--MI" and
    current_EUC_name <> "Elkhart, IN--MI" and
    current_EUC_name <> "Michigan City--La Porte, IN--MI" and
    current_EUC_name <> "Marinette--Menominee, WI--MI"):

#####
##Delete files from previous loops if they exist##
#####

#Delete temporary EUC block layer if it already exists
# (will if it was last run)
if arcpy.Exists("EUC_blks_temp.shp") == 1:
    arcpy.Delete_management("EUC_blks_temp.shp", "SHAPEFILE")

#Delete Spatial Weights Matrix if it already exists
# (will if it was last run)
if os.path.isfile(home_folder + "\current_EUC_weights.swm"):
    os.remove(home_folder + "\current_EUC_weights.swm")

#Delete Morans I output shapefile if it already exists
# (will if it was last run)
if arcpy.Exists("current_EUC_ClusterOutlier.shp") == 1:
    arcpy.Delete_management("current_EUC_ClusterOutlier.shp",
        "SHAPEFILE")

#Delete Morans I output layer if it already exists
# (will if it was last run)
if arcpy.Exists("current_EUC_ClusterOutlier_lyr") == 1:
    arcpy.Delete_management("current_EUC_ClusterOutlier_lyr",
        "LAYER")

#Delete temporary current CBD shapefile if it already exists
# (will if it was last run)
if arcpy.Exists("current_CBD_temp.shp") == 1:
    arcpy.Delete_management("current_CBD_temp.shp", "SHAPEFILE")

#Delete temporary current CBD layer if it already exists
# (will if it was last run)
if arcpy.Exists("current_CBD_temp_lyr") == 1:
    arcpy.Delete_management("current_CBD_temp_lyr", "LAYER")

#Delete donut blocks shapefile and layer if either already exist
# (will if it was last run)
if arcpy.Exists("blk_donuts.shp") == 1:
    arcpy.Delete_management("blk_donuts.shp", "SHAPEFILE")
if arcpy.Exists("blk_donuts_lyr") == 1:
    arcpy.Delete_management("blk_donuts_lyr", "LAYER")

#Delete dissolved temporary current CBD shapefile if it already
# exists (will if it was last run)
if arcpy.Exists("current_CBD_temp_diss.shp") == 1:
    arcpy.Delete_management("current_CBD_temp_diss.shp",
        "SHAPEFILE")

#Delete dissolved temporary current CBD layer if it already
# exists (will if it was last run)
if arcpy.Exists("current_CBD_temp_diss_lyr") == 1:
    arcpy.Delete_management("current_CBD_temp_diss_lyr", "LAYER")

#Delete current CBD shapefile if it already exists
# (will if it was last run)
if arcpy.Exists("current_CBD.shp") == 1:
    arcpy.Delete_management("current_CBD.shp", "SHAPEFILE")

```

```

#Delete current CBD layer if it already exists
# (will if it was last run)
if arcpy.Exists("current_CBD_lyr") == 1:
    arcpy.Delete_management("current_CBD_lyr", "LAYER")

#####
##Run Cluster and Outlier Analysis (Local Morans I) on EUC##
#####

#Select all blocks in current EUC and create a temporary shapefile
# to use for SWM
euc_select_sql = ' "EUC_Name" = \'%s\' ' %(current_EUC_name)
arcpy.SelectLayerByAttribute_management("EUC_lyr", "NEW_SELECTION",
    euc_select_sql)
arcpy.SelectLayerByLocation_management("blk_lyr", "WITHIN",
    "EUC_lyr", selection_type="NEW_SELECTION")
arcpy.Select_analysis("blk_lyr", "EUC_blks_temp")

#Create Spatial Weights Matrix for Morans I Calculations
arcpy.GenerateSpatialWeightsMatrix_stats("EUC_blks_temp.shp", "id",
    "current_EUC_weights.swm", "INVERSE_DISTANCE", "EUCLIDEAN", 1,
    "#", "#", "ROW_STANDARDIZATION")

#Run Cluster and Outlier Analysis (Local Morans I)
arcpy.ClustersOutliers_stats("EUC_blks_temp.shp", employ_field,
    "current_EUC_ClusterOutlier.shp",
    "GET_SPATIAL_WEIGHTS_FROM_FILE", "#", "#", "#",
    "current_EUC_weights.swm")

#Make Cluster/Outlier file a layer
arcpy.MakeFeatureLayer_management("current_EUC_ClusterOutlier.shp",
    "current_EUC_ClusterOutlier_lyr")

#####
##Select blocks with appropriate z-values to be included in CBD##
##and append to current CBD file                                     ##
#####

#Select all blocks with z value >= 1 and create new shapefile
arcpy.Select_analysis("current_EUC_ClusterOutlier_lyr",
    "current_CBD_temp", ' "LMiZScore" >= 1 ')

#Make temporary current CBD file a layer
arcpy.MakeFeatureLayer_management("current_CBD_temp.shp",
    "current_CBD_temp_lyr")

#Select all blocks touching a current CBD block
# and remove low-z blocks (z > -1 and < 0)
arcpy.SelectLayerByLocation_management(
    "current_EUC_ClusterOutlier_lyr", "BOUNDARY_TOUCHES",
    "current_CBD_temp_lyr", selection_type="NEW_SELECTION")
arcpy.SelectLayerByLocation_management(
    "current_EUC_ClusterOutlier_lyr", "ARE_IDENTICAL_TO",
    "current_CBD_temp_lyr", selection_type="REMOVE_FROM_SELECTION")
arcpy.SelectLayerByAttribute_management(
    "current_EUC_ClusterOutlier_lyr", "REMOVE_FROM_SELECTION",
    ' "LMiZScore" > -1 AND "LMiZScore" < 0 ')

#Append all new blocks to current CBD file
arcpy.Append_management("current_EUC_ClusterOutlier_lyr",
    "current_CBD_temp.shp", "NO_TEST")

#####
##Add donut blocks to current CBD##
#####

arcpy.Union_analysis("current_CBD_temp.shp", "blk_donuts",
    gaps="NO_GAPS")

```

```

arcpy.MakeFeatureLayer_management("blk_donuts.shp",
    "blk_donuts_lyr")
arcpy.SelectLayerByAttribute_management("blk_donuts_lyr",
    "NEW_SELECTION",' "FID_curren" = -1 ')
arcpy.SelectLayerByLocation_management("blk_lyr","WITHIN",
    "blk_donuts_lyr",selection_type="NEW_SELECTION")
arcpy.Append_management("blk_lyr","current_CBD_temp.shp","NO_TEST")

#####
##Prepare CBD polygon to be added to the CBD polygons shapefile##
#####

#Dissolve CBD temp file
arcpy.Dissolve_management("current_CBD_temp.shp",
    "current_CBD_temp_diss","", "", "SINGLE_PART", "")

#Make temporary dissolved current CBD file a layer
arcpy.MakeFeatureLayer_management("current_CBD_temp_diss.shp",
    "current_CBD_temp_diss_lyr")

#Select only polygon for principle city and create new shapefile
arcpy.SelectLayerByLocation_management("current_CBD_temp_diss_lyr",
    "CONTAINS","CBD_Points_lyr",selection_type="NEW_SELECTION")
arcpy.Select_analysis("current_CBD_temp_diss_lyr","current_CBD")

#Add in a name field to the file
arcpy.AddField_management("current_CBD.shp","Name","TEXT",
    "", "", 128)

#Make current CBD file a layer
arcpy.MakeFeatureLayer_management("current_CBD.shp",
    "current_CBD_lyr")

#Create a dummy cursor and use it to loop through CBDs in current
# file - Will actually only loop once to set current CBD name
dummy_cursor = arcpy.UpdateCursor("current_CBD_lyr")
for dummy_value in dummy_cursor:
    dummy_value.setValue("Name",current_EUC_name)
    dummy_cursor.updateRow(dummy_value)

#####
##Append CBD polygon to the CBD polygons shapefile##
#####

#Create file if first iteration, otherwise, append to file
if loop_count == 1:
    arcpy.Select_analysis("current_CBD_lyr","CBD_polygons")
else:
    arcpy.Append_management("current_CBD_lyr","CBD_polygons.shp",
        "NO_TEST")

#Increase loop counter for next iteration
loop_count = loop_count + 1

#####
##Cleanup all temporary shapefiles##
#####

#Delete EUC layer, blk layer, and CBD points layer for cleanup purposes
if arcpy.Exists("EUC_lyr") == 1:
    arcpy.Delete_management("EUC_lyr","LAYER")
if arcpy.Exists("blk_lyr") == 1:
    arcpy.Delete_management("blk_lyr","LAYER")
if arcpy.Exists("CBD_Points_lyr" == 1):
    arcpy.Delete_management("CBD_Points_lyr","LAYER")

```

```

#Delete temporary EUC block layer for cleanup purposes
if arcpy.Exists("EUC_blks_temp.shp") == 1:
    arcpy.Delete_management("EUC_blks_temp.shp", "SHAPEFILE")

# Delete Spatial Weights Matrix for cleanup purposes
if os.path.isfile(home_folder + "\current_EUC_weights.swm"):
    os.remove(home_folder + "\current_EUC_weights.swm")

#Delete Morans I output shapefile and layer for cleanup purposes
if arcpy.Exists("current_EUC_ClusterOutlier.shp") == 1:
    arcpy.Delete_management("current_EUC_ClusterOutlier.shp", "SHAPEFILE")
if arcpy.Exists("current_EUC_ClusterOutlier_lyr") == 1:
    arcpy.Delete_management("current_EUC_ClusterOutlier_lyr", "LAYER")

#Delete temporary current CBD shapefile and layer for cleanup purposes
if arcpy.Exists("current_CBD_temp.shp") == 1:
    arcpy.Delete_management("current_CBD_temp.shp", "SHAPEFILE")
if arcpy.Exists("current_CBD_temp_lyr") == 1:
    arcpy.Delete_management("current_CBD_temp_lyr", "LAYER")

#Delete donut blocks shapefile and layer for cleanup purposes
if arcpy.Exists("blk_donuts.shp") == 1:
    arcpy.Delete_management("blk_donuts.shp", "SHAPEFILE")
if arcpy.Exists("blk_donuts_lyr") == 1:
    arcpy.Delete_management("blk_donuts_lyr", "LAYER")

#Delete dissolved temp current CBD shapefile and layer for cleanup purposes
if arcpy.Exists("current_CBD_temp_diss.shp") == 1:
    arcpy.Delete_management("current_CBD_temp_diss.shp", "SHAPEFILE")
if arcpy.Exists("current_CBD_temp_diss_lyr") == 1:
    arcpy.Delete_management("current_CBD_temp_diss_lyr", "LAYER")

#Delete current CBD shapefile and layer for cleanup purposes
if arcpy.Exists("current_CBD.shp") == 1:
    arcpy.Delete_management("current_CBD.shp", "SHAPEFILE")
if arcpy.Exists("current_CBD_lyr") == 1:
    arcpy.Delete_management("current_CBD_lyr", "LAYER")

```

Appendix E: WalkScore_Locator.py – Full Python Script

```
#####WalkScore_Locator.py#####
##This program queries WalkScore.com to collect and report Walk Scores ##
##for lat/lon pairs. Currently the program works for as many pairs as ##
##the user would like. The program currently accepts a CSV file that has##
##only three columns, an identifier, a latitude, and a longitude. The ##
##file appends a column of Walk Scores to a new file. This program should##
##be executed and the Walk Score column should be added to the block file##
##before running EUC_Metrics.py. ##
#####

#Import Libraries
import urllib2
import csv
import time

#Initialize folder and file locations
blk_file_addr = 'H:/PHD/Metrics/block_latlon.csv'
write_file_addr = 'H:/PHD/Metrics/block_ws_all.csv'

#Initialize string where the Walk Score will be found in the source code
find_string = 'has a Walk Score of '
find_str_len = len(find_string)

#Open the block CSV file
with open(blk_file_addr) as readfile:

    #Create a blank list that will store Walk Scores
    new_rows_list = []

    #Loop though each row of the block CSV file to calculate Walk Score
    # for each block lat/lon
    readdata = csv.reader(readfile)
    for readrow in readdata:
        #Try to get the numeric value of the first entry in the row
        try:
            float(readrow[0])
        #If this fails, this should be the header row, store header labels
        # and skip to next loop iteration
        except ValueError:
            col1_name = readrow[0]
            col2_name = readrow[1]
            col3_name = readrow[2]
            continue
        #As long as the row only has three columns, store these as the
        # geographic id, latitude, and longitude
        if len(readrow) == 3:
            gid = readrow[0]
            lat = readrow[1]
            lon = readrow[2]

            #Preprocess latitude and longitude values for proper formatting
            # as expected by WalkScore.com by first removing + signs and
            # then removing leading 0s. Walk Score site has a problem with
            # leading 0s giving different scores even though otherwise
            # identical numbers are used
            if lat[0] == '+':
                lat = lat[1:]
            if lon[0] == '+':
                lon = lon[1:]
            if lon[0] == '-' and lon[1] == '0':
                lon = lon[0] + lon[2:]
```

```

#Create the web address to be called
url_str = ('https://www.walkscore.com/score/loc/lat=' +
          lat + '/lng=' + lon)

#Try to call the website and receive a response. If it fails,
# pause for 10 seconds and keep trying until it succeeds.
# Site tends to fail every 1000 calls or so.
loaded = 0
while loaded == 0:
    try:
        req = urllib2.Request(url_str)
        response = urllib2.urlopen(req)
        loaded = 1
    except:
        time.sleep(10)

#Store the source code for the page
page_data_full = response.read()

#Find the string that preempts the Walk Score value in the
# source code
ws_position_start = page_data_full.find(find_string)

#If find command succeeds, store the value
if ws_position_start <> -1:

    #Locate Walk Score position and store 3 possible values for
    # Walk Score (in case it is a three digit value, e.g., 100)
    ws_position = ws_position_start + find_str_len
    ws_string_1 = page_data_full[ws_position]
    ws_string_2 = page_data_full[ws_position+1]
    ws_string_3 = page_data_full[ws_position+2]

    #Store the Walk Score
    # If value is single digit, store just 1 character
    # If two digits, store 2 characters
    # If three digits, store 3 characters
    if ws_string_2 == ' ':
        ws_string = ws_string_1
    elif ws_string_3 == ' ':
        ws_string = ws_string_1 + ws_string_2
    else:
        ws_string = ws_string_1 + ws_string_2 + ws_string_3

    #Convert Walk Score to a number
    walk_score = int(ws_string)

#If find command failed, assume Walk Score is 0
else:
    walk_score = 0

#Create a new row to be appended to the list and append the row
new_row = [gid, lat, lon, walk_score]
new_rows_list.append(new_row)

#Close the file after all blocks are complete
readfile.close()

```



```

#Open the new file for writing
with open(write_file_addr,'ab') as writefile:
    writedata = csv.writer(writefile)

    #Write the column headers
    writedata.writerow([coll_name,col2_name,col3_name,'WalkScore'])

    #Loop through each row in the list of Walk Scores and write the row to
    # the new CSV file
    for row in new_rows_list:
        writedata.writerow([row[0], row[1], row[2], row[3]])

#Close the file after all blocks have been written to the file
writefile.close()

```

Appendix F: EUC_Sprawl_Score.py – Full Python Script

```
#####EUC_Sprawl_Score.py#####
##This program uses Principal Components Analysis (PCA) to calculate four##
##factor category scores: development density, land use mix, activity    ##
##centering, and street accessibility. These scores are then weighted  ##
##and an overall Sprawl Score is calculated for each of the 42 pilot EUCS. ##
##This program should be executed only after using EUC_metrics.py to    ##
##solve for the 21 metric values. No steps need to be completed prior to##
##executing other than that. The only input file is EUC_All.shp. The   ##
##five values are appended to this file and it is the only output. The  ##
##approach used to complete the PCA and the Sprawl Score calculation is  ##
##outlined in the dissertation Chapter 7, but there should be enough    ##
##comments here to follow along.                                       ##
#####

#Note: Uses city population - must be pre-joined as of the current
#version - should be updated so that this data comes from 2010 places
#with demographics joined

#Import Libraries and checkout extensions
import numpy
import arcpy

#####
##Initializations##
#####

#Initialize folder and file locations
arcpy.env.workspace = "H:\PHD\Metrics" #Set location of working folder
EUC_file = "EUC_All.shp"

#Set groups of cities to be used for PCA
#This version of the program calculates PCA once for each group
#Each row is a separate group
#The pilot group includes four groups, totaling 42 cities
Name_Group = ["Traverse City, MI", "Adrian, MI", "Mount Pleasant, MI",
              "Marquette, MI"],
              ["Owosso, MI", "Escanaba, MI", "Alma--St. Louis, MI",
              "Coldwater, MI", "Houghton, MI", "Ionia, MI", "Alpena, MI",
              "Big Rapids, MI", "Lapeer, MI", "Sault Ste. Marie, MI",
              "Sturgis, MI", "Cadillac, MI", "Hillsdale, MI",
              "Ludington, MI"],
              ["Greenville, MI", "Manistee, MI", "Houghton Lake, MI",
              "Gaylord, MI", "Petoskey, MI", "Hastings, MI", "Caro, MI"],
              ["Cheboygan, MI", "Fremont, MI", "East Tawas, MI",
              "Charlevoix, MI", "Grayling, MI", "Harrison, MI", "Bad Axe, MI",
              "Manistique, MI", "Newberry, MI", "Iron River, MI",
              "Munising, MI", "Gladwin, MI", "Sandusky, MI", "Kalkaska, MI",
              "Rogers City, MI", "Hart, MI", "St. Ignace, MI"]]

#Number of principal components to use from PCA
num_components = 1

#Initialize number of variables per category
num_vars_d = 8 #_d for development [d]ensity (used throughout program)
num_vars_m = 3 #_m for land use [m]ix (used throughout program)
num_vars_c = 5 #_c for activity [c]entering (used throughout program)
num_vars_s = 5 #_s for [s]treet accessibility (used throughout program)

#####
##Create New Fields##
#####

#F1 - Development Density (uses metrics D1-D8)
arcpy.AddField_management(EUC_file, "EW_F1_dens", "DOUBLE")
```

```

#F2 - Mixed Use (uses metrics M1-M3)
arcpy.AddField_management(EUC_file,"EW_F2_mixu","DOUBLE")

#F3 - Centering (uses metrics C1-C5)
arcpy.AddField_management(EUC_file,"EW_F3_cntr","DOUBLE")

#F4 - Street (uses metrics S1-S5)
arcpy.AddField_management(EUC_file,"EW_F4_strt","DOUBLE")

#Final Sprawl Score
arcpy.AddField_management(EUC_file,"SprwlScore","DOUBLE")

#####
##Make Layers for Processing##
#####

#Make EUC a layer
if arcpy.Exists("EUC_lyr") == 1:
    arcpy.Delete_management("EUC_lyr","LAYER")
arcpy.MakeFeatureLayer_management(EUC_file,"EUC_lyr")

#####
##Main Loop - for each group in the pilot group##
#####

for i in range(0,len(Name_Group)):

    #####
    ##Initializations for current iteration of loop##
    #####

    #Initialize arrays for each of the 21 metrics
    EWpopden = numpy.array([]) #D1
    EWempden = numpy.array([]) #D2
    EWlt1500 = numpy.array([]) #D3
    EWgt12500 = numpy.array([]) #D4
    EWurbden = numpy.array([]) #D5
    EWdgcen = numpy.array([]) #D6
    EWpopdcen = numpy.array([]) #D7
    EWempdcen = numpy.array([]) #D8
    EWjobpop = numpy.array([]) #M1
    EWjobmix = numpy.array([]) #M2
    EWwalkscor = numpy.array([]) #M3
    EWvarpop = numpy.array([]) #C1
    EWvaremp = numpy.array([]) #C2
    EWdgrad = numpy.array([]) #C3
    EWpopcen = numpy.array([]) #C4
    EWempcen = numpy.array([]) #C5
    EWsmlblk = numpy.array([]) #M1
    EWavgbklsz = numpy.array([]) #M2
    EWavgbklkn = numpy.array([]) #M3
    EWintden = numpy.array([]) #M4
    EW4way = numpy.array([]) #M5

    #Initiaailize an array for population of the EUC
    EUCpop = numpy.array([])

    #####
    ##Loop 2 - for each EUC##
    #####

    #Loop 2 and 3 will go through each EUC in the name group
    # and add all metrics to the metric arrays

    #Create a cursor to go through each EUC
    EUC_cursor = arcpy.SearchCursor("EUC_lyr")
    for current_EUC in EUC_cursor:

```

```

#Store the name of the current EUC
current_EUC_name = current_EUC.getValue("EUC_Name")

#####
##Loop 3 - for each name in the current name group##
#####

for j in range(0,len(Name_Group[i])):

    #if current EUC row matches the name of the EUC in the current
    # name group, then append all metric values onto arrays
    if current_EUC_name == Name_Group[i][j]:
        EWpopden = numpy.append(EWpopden,
                                current_EUC.getValue("EWpopden"))
        EWempden = numpy.append(EWempden,
                                current_EUC.getValue("EWempden"))
        EWlt1500 = numpy.append(EWlt1500,
                                current_EUC.getValue("EWlt1500"))
        EWgt12500 = numpy.append(EWgt12500,
                                current_EUC.getValue("EWgt12500"))
        EWurbden = numpy.append(EWurbden,
                                current_EUC.getValue("EWurbden"))
        EWdgcen = numpy.append(EWdgcen,
                                current_EUC.getValue("EWdgcen"))
        EWpopdcen = numpy.append(EWpopdcen,
                                current_EUC.getValue("EWpopdcen"))
        EWempdcen = numpy.append(EWempdcen,
                                current_EUC.getValue("EWempdcen"))
        EWjobpop = numpy.append(EWjobpop,
                                current_EUC.getValue("EWjobpop"))
        EWjobmix = numpy.append(EWjobmix,
                                current_EUC.getValue("EWjobmix"))
        EWwalkscor = numpy.append(EWwalkscor,
                                current_EUC.getValue("EWwalkscor"))
        EWvarpop = numpy.append(EWvarpop,
                                current_EUC.getValue("EWvarpop"))
        EWvaremp = numpy.append(EWvaremp,
                                current_EUC.getValue("EWvaremp"))
        EWdgrad = numpy.append(EWdgrad,
                                current_EUC.getValue("EWdgrad"))
        EWpopcen = numpy.append(EWpopcen,
                                current_EUC.getValue("EWpopcen"))
        EWempcen = numpy.append(EWempcen,
                                current_EUC.getValue("EWempcen"))
        EWsmlblk = numpy.append(EWsmlblk,
                                current_EUC.getValue("EWsmlblk"))
        EWavgbklsz = numpy.append(EWavgbklsz,
                                current_EUC.getValue("EWavgbklsz"))
        EWavgbklkn = numpy.append(EWavgbklkn,
                                current_EUC.getValue("EWavgbklkn"))
        EWintden = numpy.append(EWintden,
                                current_EUC.getValue("EWintden"))
        EW4way = numpy.append(EW4way,
                                current_EUC.getValue("EW4way"))

    #Note - Largest group uses average of four populations
    # Smallest group uses city population
    # Middle groups use EUC population
    # This should be modified for future implementations
    if i == 0:
        pop_use = 56947
    elif i == 3:
        pop_use = current_EUC.getValue("pop_city")
    else:
        pop_use = current_EUC.getValue("pop")
    EUCpop = numpy.append(EUCpop, pop_use)

```

```

#####
##End Loop 3 and Loop 2##
#####

#Concatenate metric scores into category arrays
EWdensity = numpy.array([EWpopden,EWempden,EWlt1500,EWgt12500,EWurbden,
    EWdgcen,EWpopdcen,EWempdcen])
EWmixuse = numpy.array([EWjobpop,EWjobmix,EWwalkscor])
EWcentering = numpy.array([EWvarpop,EWvaremp,EWdgrad,EWpopcen,
    EWempcen])
EWstreet = numpy.array([EWSmlblk,EWavgbklsz,EWavgbklkn,EWintden,
    EW4way])

#####
##Principal Components Analysis##
#####

##PCA - Step 1 - Correlation matrices##
#Determine correlation matrices for each category
corrmat_d = numpy.corrcoef(EWdensity)
corrmat_m = numpy.corrcoef(EWmixuse)
corrmat_c = numpy.corrcoef(EWcentering)
corrmat_s = numpy.corrcoef(EWstreet)

##PCA - Step 2 - Eigenvalues/eigenvectors##
#Determine Eigenvalue, proportions, and eigenvectors for each category
eig_val_d, eig_vec_d = numpy.linalg.eig(corrmat_d)
eig_prop_d = eig_val_d / eig_val_d.sum()

eig_val_m, eig_vec_m = numpy.linalg.eig(corrmat_m)
eig_prop_m = eig_val_m / eig_val_m.sum()

eig_val_c, eig_vec_c = numpy.linalg.eig(corrmat_c)
eig_prop_c = eig_val_c / eig_val_c.sum()

eig_val_s, eig_vec_s = numpy.linalg.eig(corrmat_s)
eig_prop_s = eig_val_s / eig_val_s.sum()

#Group and sort the eigenvalues, propotions, and eigenvectors
eig_group_d = [(eig_val_d[k],eig_prop_d[k],eig_vec_d[:,k])
    for k in range(len(eig_val_d))]
eig_group_d.sort()
eig_group_d.reverse()

eig_group_m = [(eig_val_m[k],eig_prop_m[k],eig_vec_m[:,k])
    for k in range(len(eig_val_m))]
eig_group_m.sort()
eig_group_m.reverse()

eig_group_c = [(eig_val_c[k],eig_prop_c[k],eig_vec_c[:,k])
    for k in range(len(eig_val_c))]
eig_group_c.sort()
eig_group_c.reverse()

eig_group_s = [(eig_val_s[k],eig_prop_s[k],eig_vec_s[:,k])
    for k in range(len(eig_val_s))]
eig_group_s.sort()
eig_group_s.reverse()

##PCA - Step 3 - Weights vector##
#Verify that eigenvalues and eigenvectors are real
pc_weights_d = eig_group_d[0][2].real
pc_weights_m = eig_group_m[0][2].real
pc_weights_c = eig_group_c[0][2].real
pc_weights_s = eig_group_s[0][2].real

```

```

#Correct for the problem where more eigenvalues are negative
# than positive by negating weights
if (pc_weights_d < 0).sum() > .5*pc_weights_d.shape[0]:
    pc_weights_d_correct = -pc_weights_d
else:
    pc_weights_d_correct = pc_weights_d
if (pc_weights_m < 0).sum() > .5*pc_weights_m.shape[0]:
    pc_weights_m_correct = -pc_weights_m
else:
    pc_weights_m_correct = pc_weights_m

if (pc_weights_c < 0).sum() > .5*pc_weights_c.shape[0]:
    pc_weights_c_correct = -pc_weights_c
else:
    pc_weights_c_correct = pc_weights_c

if (pc_weights_s < 0).sum() > .5*pc_weights_s.shape[0]:
    pc_weights_s_correct = -pc_weights_s
else:
    pc_weights_s_correct = pc_weights_s

#Fix problem with weights - should probably fix this for V2
fix_d = [1,1,-1,1,1,1,1,1]
fix_m = [1,1,1]
fix_c = [1,1,-1,1,1]
fix_s = [1,-1,-1,1,1]
pc_weights_d_correct = abs(pc_weights_d_correct) * fix_d
pc_weights_m_correct = abs(pc_weights_m_correct) * fix_m
pc_weights_c_correct = abs(pc_weights_c_correct) * fix_c
pc_weights_s_correct = abs(pc_weights_s_correct) * fix_s

#Compile weights matrices for each category and store
# first eigenvector weights only
#Also rechecks to make sure less values are negative than positive
W_d = pc_weights_d_correct.reshape(num_vars_d,1)
if (W_d < 0).sum() > .5*W_d.shape[0]:
    W_d = -W_d

W_m = pc_weights_m_correct.reshape(num_vars_m,1)
if (W_m < 0).sum() > .5*W_m.shape[0]:
    W_m = -W_m

W_c = pc_weights_c_correct.reshape(num_vars_c,1)
if (W_c < 0).sum() > .5*W_c.shape[0]:
    W_c = -W_c

W_s = pc_weights_s_correct.reshape(num_vars_s,1)
if (W_s < 0).sum() > .5*W_s.shape[0]:
    W_s = -W_s

##PCA - Step 4 - Means and standard deviations##
#Calculate statistics for each of the 21 metrics by category

#Means
means_d = EWdensity.mean(axis = 1)
num_vals_d = EWdensity.shape[1]
means_m = EWMixuse.mean(axis = 1)
num_vals_m = EWMixuse.shape[1]
means_c = EWcentering.mean(axis = 1)
num_vals_c = EWcentering.shape[1]
means_s = EWstreet.mean(axis = 1)
num_vals_s = EWstreet.shape[1]

#Standard Deviations
#multiplier converts population SD to sample SD
stdevs_d = (numpy.sqrt(EWdensity.var(axis = 1) *
    num_vals_d / (num_vals_d - 1)))

```

```

stdevs_m = (numpy.sqrt(EWmixuse.var(axis = 1) *
    num_vals_m / (num_vals_m - 1)))
stdevs_c = (numpy.sqrt(EWcentering.var(axis = 1) *
    num_vals_c / (num_vals_c - 1)))
stdevs_s = (numpy.sqrt(EWstreet.var(axis = 1) *
    num_vals_s / (num_vals_s - 1)))

##PCA - Step 5 - Z-values##
#Intialize standardized value arrays for each of the 21 metrics
# by category
EWdensity_standardized = numpy.array([],order=2)
EWmixuse_standardized = numpy.array([],order=2)
EWcentering_standardized = numpy.array([],order=2)
EWstreet_standardized = numpy.array([],order=2)

#Compile standardized values (z-values) for each of the 21 metrics
# by category
for var in range(0,num_vars_d):
    new_array = (EWdensity[var] - means_d[var])/stdevs_d[var]
    EWdensity_standardized = numpy.append(
        EWdensity_standardized,new_array)
EWdensity_standardized = EWdensity_standardized.reshape(
    (num_vars_d,num_vals_d))

for var in range(0,num_vars_m):
    new_array = (EWmixuse[var] - means_m[var])/stdevs_m[var]
    EWMixuse_standardized = numpy.append(
        EWMixuse_standardized,new_array)
EWMixuse_standardized = EWMixuse_standardized.reshape(
    (num_vars_m,num_vals_m))

for var in range(0,num_vars_c):
    new_array = (EWcentering[var] - means_c[var])/stdevs_c[var]
    EWcentering_standardized = numpy.append(
        EWcentering_standardized,new_array)
EWcentering_standardized = EWcentering_standardized.reshape(
    (num_vars_c,num_vals_c))

for var in range(0,num_vars_s):
    new_array = (EWstreet[var] - means_s[var])/stdevs_s[var]
    EWstreet_standardized = numpy.append(
        EWstreet_standardized,new_array)
EWstreet_standardized = EWstreet_standardized.reshape(
    (num_vars_s,num_vals_s))

##PCA - Step 6 - Principal Component Extraction##
#Calculate score for each of the four categories
score = 0 #principal component
for var in range(0,num_vars_d):
    score = score + W_d[var]*EWdensity_standardized[var]
EWdensity_score = 100 + 25*score #normalized score

score = 0 #principal component
for var in range(0,num_vars_m):
    score = score + W_m[var]*EWMixuse_standardized[var]
EWMixuse_score = 100 + 25*score #normalized score

score = 0 #principal component
for var in range(0,num_vars_c):
    score = score + W_c[var]*EWcentering_standardized[var]
EWcentering_score = 100 + 25*score #normalized score

score = 0 #principal component
for var in range(0,num_vars_s):
    score = score + W_s[var]*EWstreet_standardized[var]
EWstreet_score = 100 + 25*score #normalized score

```

```

#####
##Sprawl Score Calculator##
#####

##SS - Step 1 - Perform linear regression of scores on ln of pop##

#Calculate total of raw normalized scores
raw_score_total = (EWDensity_score + EWMixuse_score + EWcentering_score +
    EWstreet_score)

#Take natural log of population vector
EUCpop_ln = numpy.log(EUCpop)

#Prepare population vector as a matrix
EUCpop_ln_matrix = numpy.vstack(
    [EUCpop_ln, numpy.ones(len(EUCpop_ln))]).T

#Perform the linear regression storing slope and intercept
slope, intercept = numpy.linalg.lstsq(
    EUCpop_ln_matrix, raw_score_total)[0]

##SS - Step 2 - Calculate standardized residuals##

#Predict the sprawl score total from regression values
predicted_score_total = slope * EUCpop_ln + intercept

#Residuals are raw score minus the predicted score
residuals = raw_score_total - predicted_score_total

#Standardize the residuals
sd_residuals = numpy.sqrt(
    numpy.var(residuals) * numpy.shape(residuals)[0] /
    (numpy.shape(residuals)[0] - 1))

#Calculate standard residuals - this is the raw sprawl score
standard_residuals = residuals / sd_residuals

##SS - Step 3 - Calculate normalized Sprawl Score##
sprawl_score = 100.0 + standard_residuals * 25.0

#####
##Store final scores in EUC file##
#####

#Initialize counter for number of scores entered
score_count = 0

#####
##Loop 4 - for each EUC##
#####

#Create a cursor to go through each EUC
EUC_cursor = arcpy.UpdateCursor("EUC_lyr")
for current_EUC in EUC_cursor:

    #Store the name of the current EUC
    current_EUC_name = current_EUC.getValue("EUC_Name")

    #####
    ##Loop 5 - for each name in the current name group##
    #####

    #Loop through each name in the current name group
    for j in range(0,len(Name_Group[i])):

```



```

#if current EUC row matches the name of the EUC in the current
# name group, then store the five values in the EUC file
if current_EUC_name == Name_Group[i][j]:

    current_EUC.setValue("EW_F1_dens",
        EWdensity_score[score_count]) #F1 - Density
    current_EUC.setValue("EW_F2_mixu",
        EWmixuse_score[score_count]) #F2 - Mixed Use
    current_EUC.setValue("EW_F3_cntr",
        EWcentering_score[score_count]) #F3 - Centering
    current_EUC.setValue("EW_F4_strt",
        EWstreet_score[score_count]) #F4 - Streets
    current_EUC.setValue("SprwlScore",
        sprawl_score[score_count]) #Final Sprawl Score

    #increment score counter
    score_count += 1

#Update the EUC row
EUC_cursor.updateRow(current_EUC)

```

Appendix G: Sprawl Scorecards for All 42 Pilot Cities

This section contains a Sprawl Scorecard for each of the 42 pilot cities in the study. They are in alphabetical order. For an overview of the scorecards, see Chapter 7. High Resolution scorecards are included in Appendix H, the enclosed DVD that is included in the media pocket on the inside back cover of this dissertation.

The street map used in Figures G.1-G.42 is sourced from ESRI's World Street Map dataset which is available for academic use [112]. Population numbers are from the 2010 Census [7]. The rest of the data in the figures were compiled from the datasets used to derive the Sprawl scores [79,101,102,105,106].

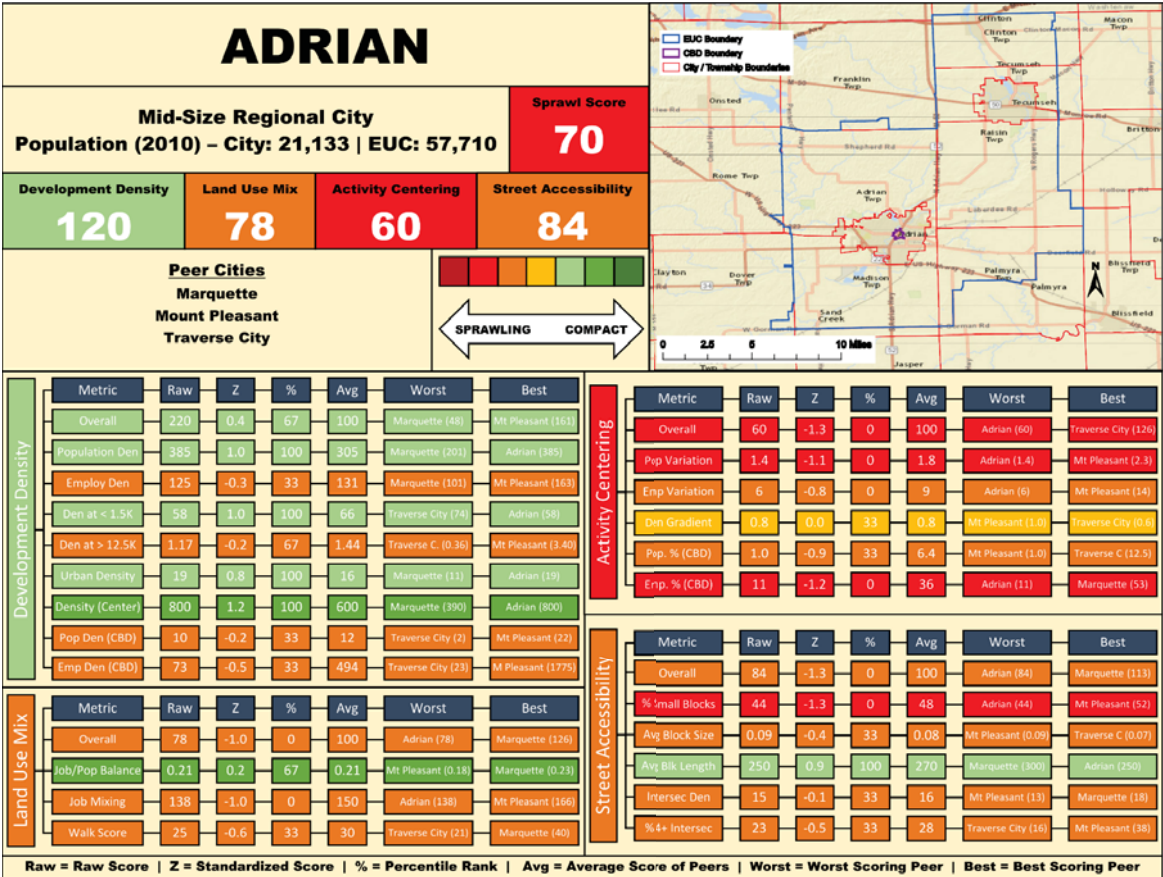


Figure G.2: Alma Sprawl Scorecard

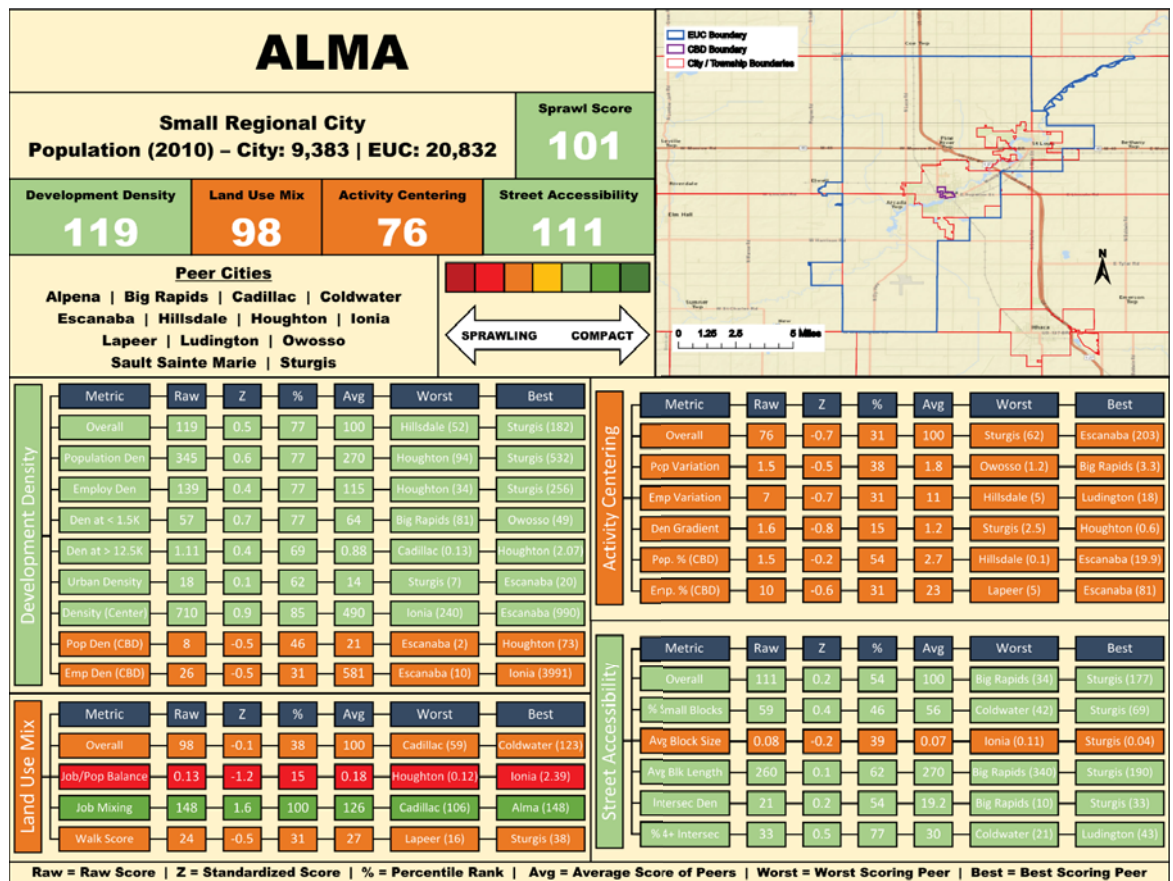


Figure G.3: Alpena Sprawl Scorecard

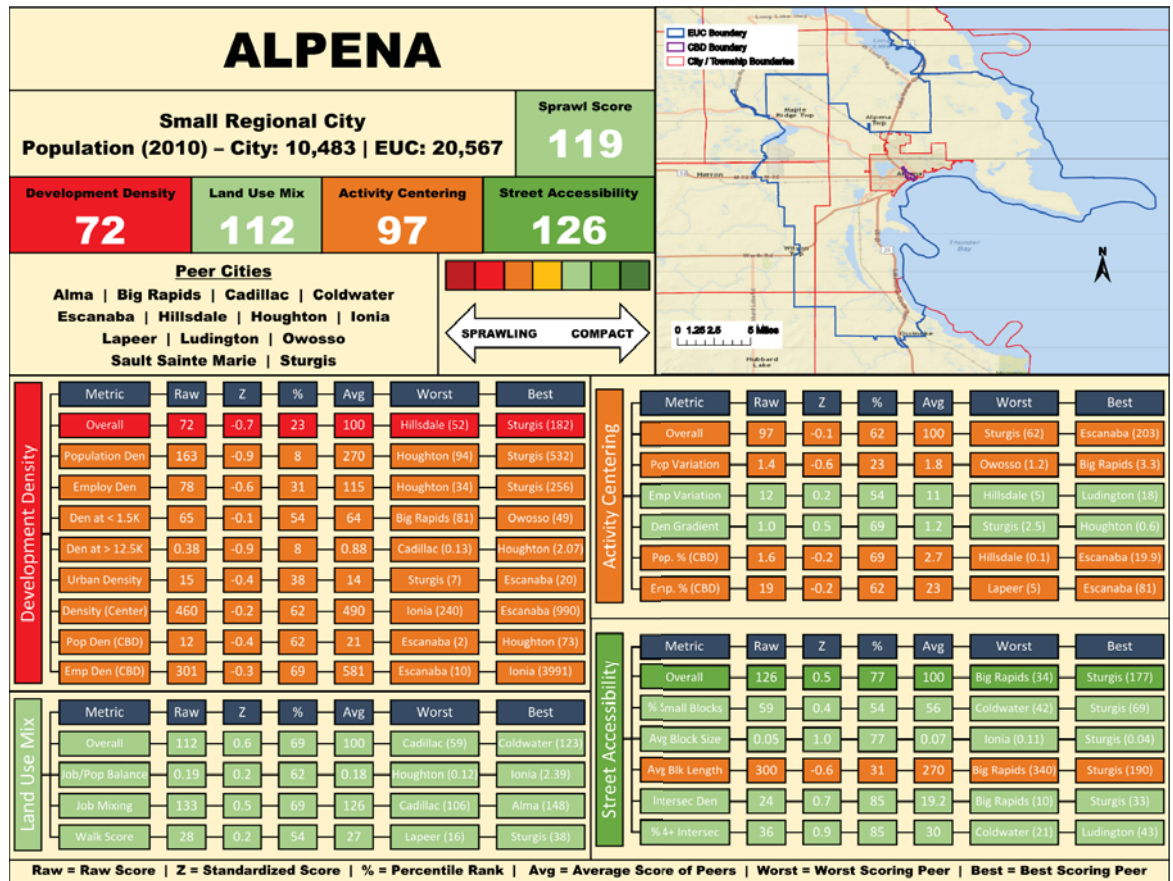


Figure G.4: Bad Axe Sprawl Scorecard

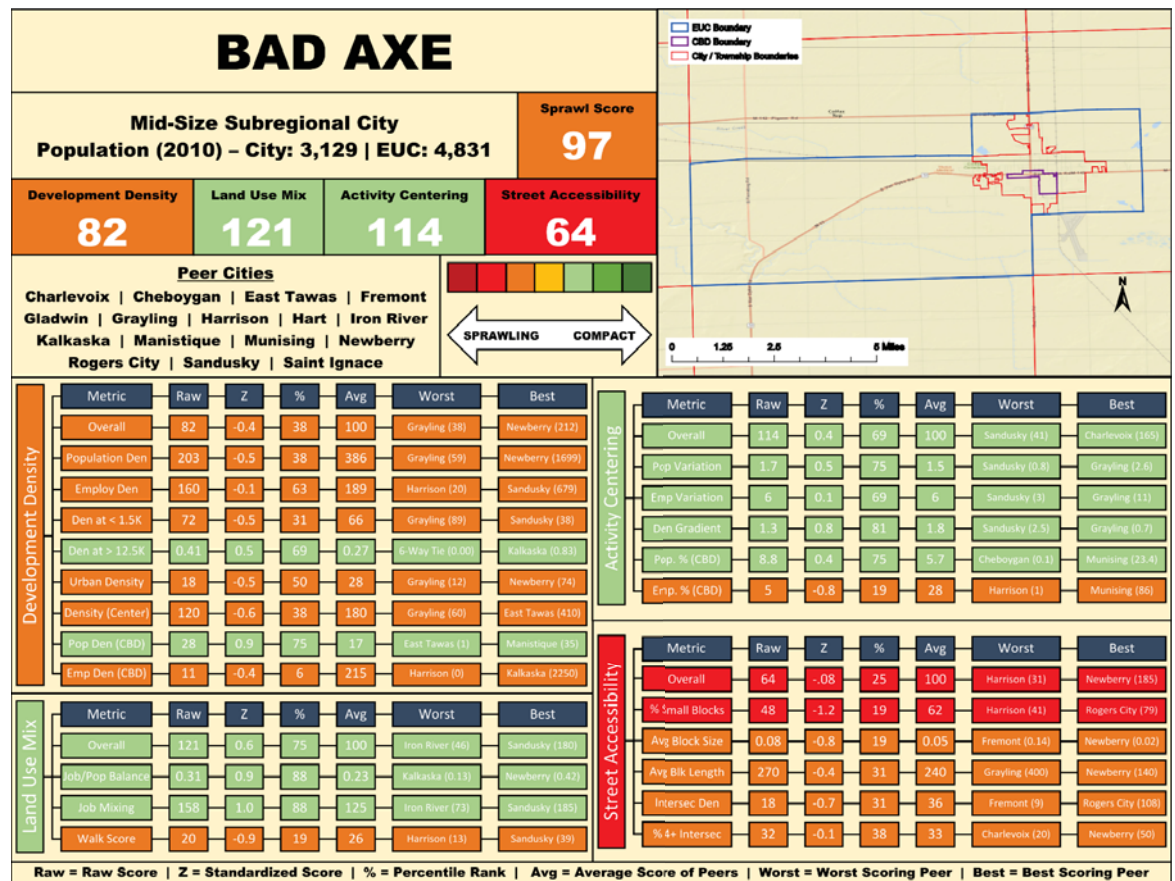


Figure G.5: Big Rapids Sprawl Scorecard

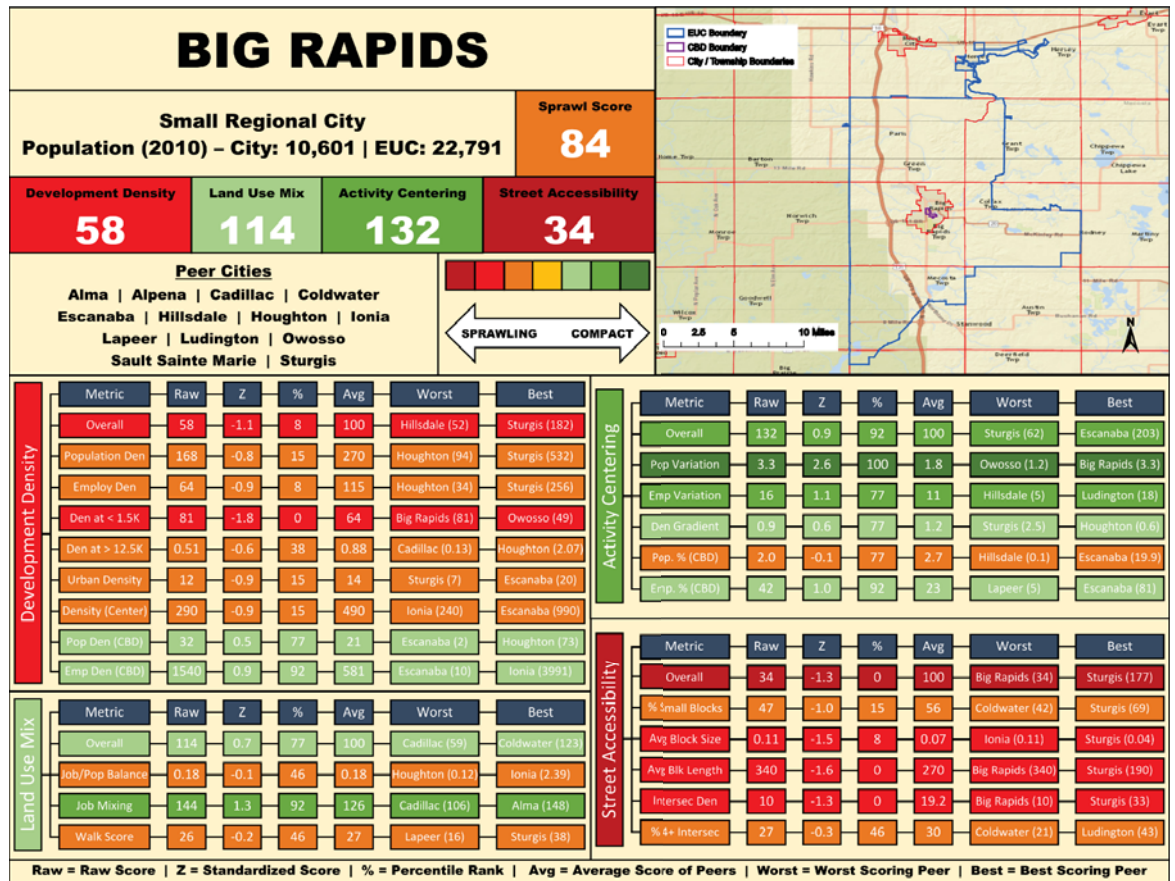


Figure G-6: Cadillac Sprawl Scorecard

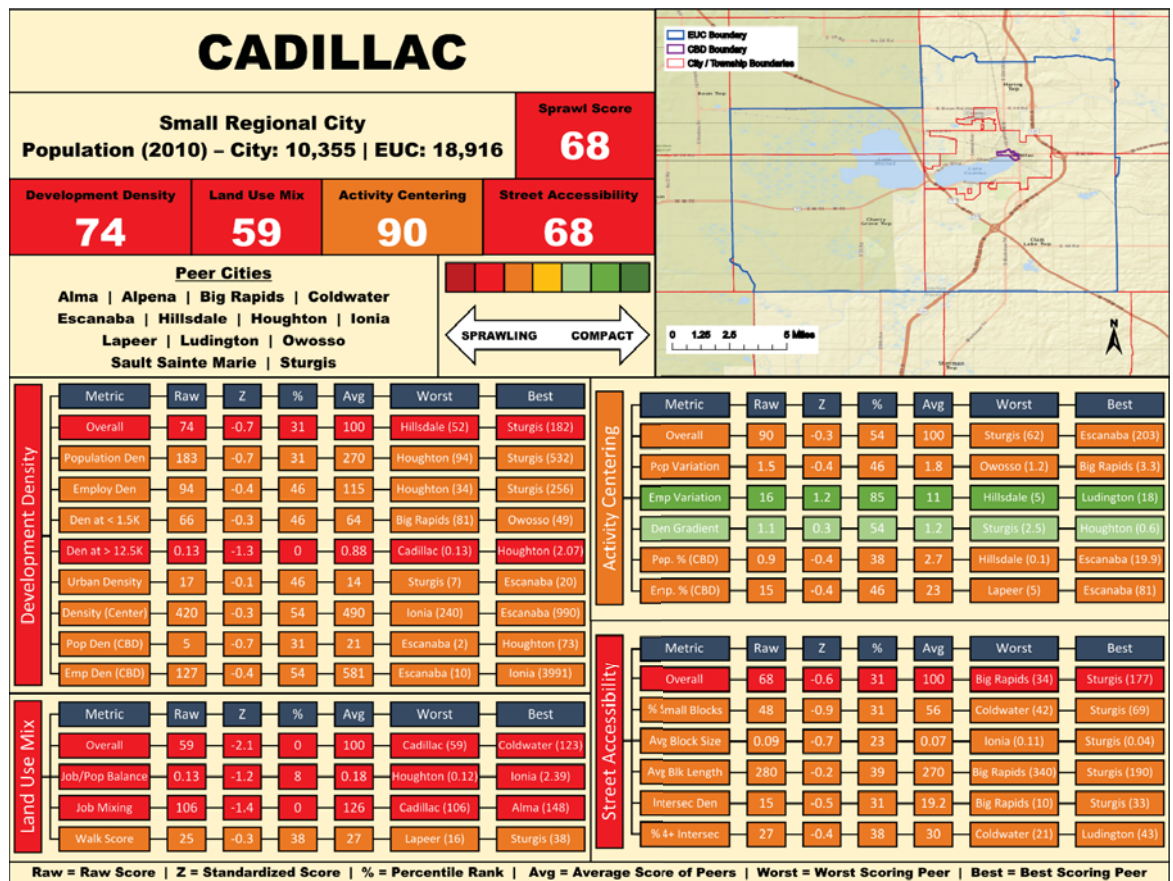
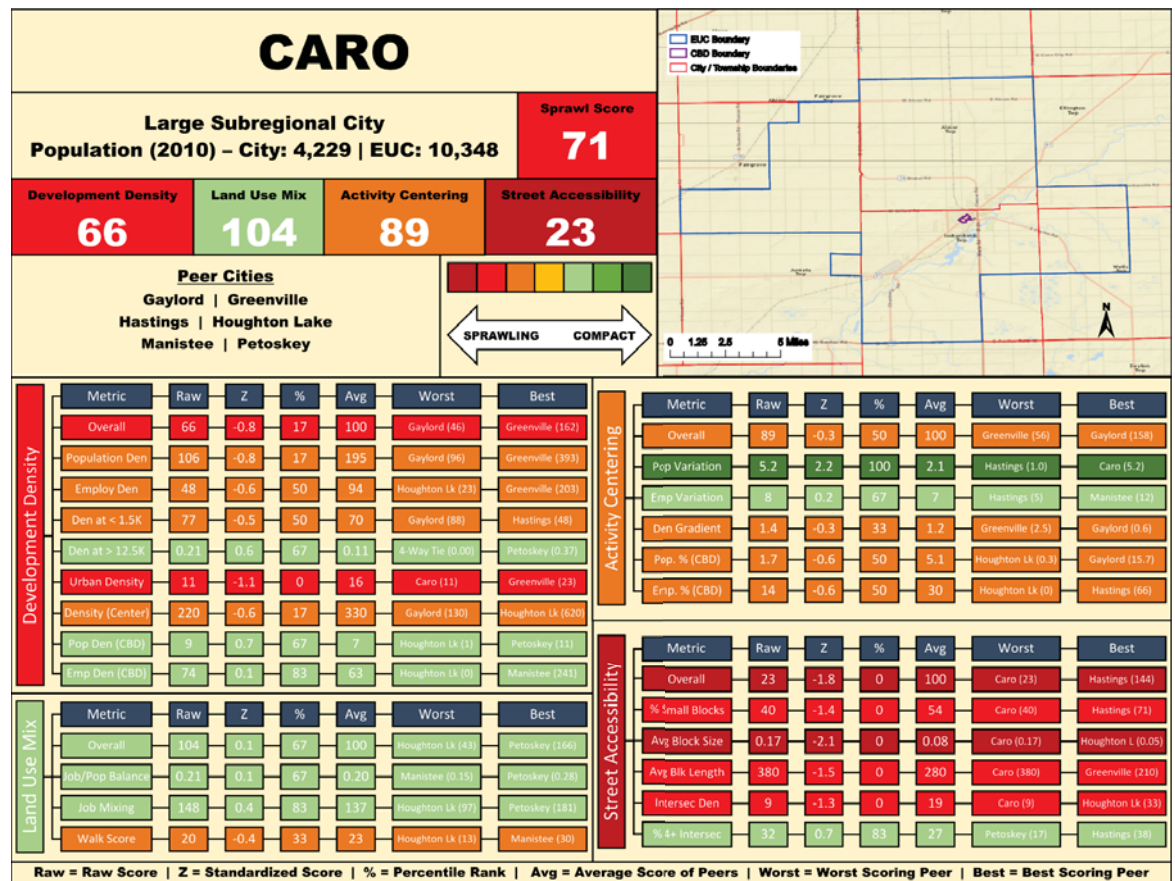


Figure G.7: Caro Sprawl Scorecard



Raw = Raw Score | Z = Standardized Score | % = Percentile Rank | Avg = Average Score of Peers | Worst = Worst Scoring Peer | Best = Best Scoring Peer

Figure G.8: Charlevoix Sprawl Scorecard

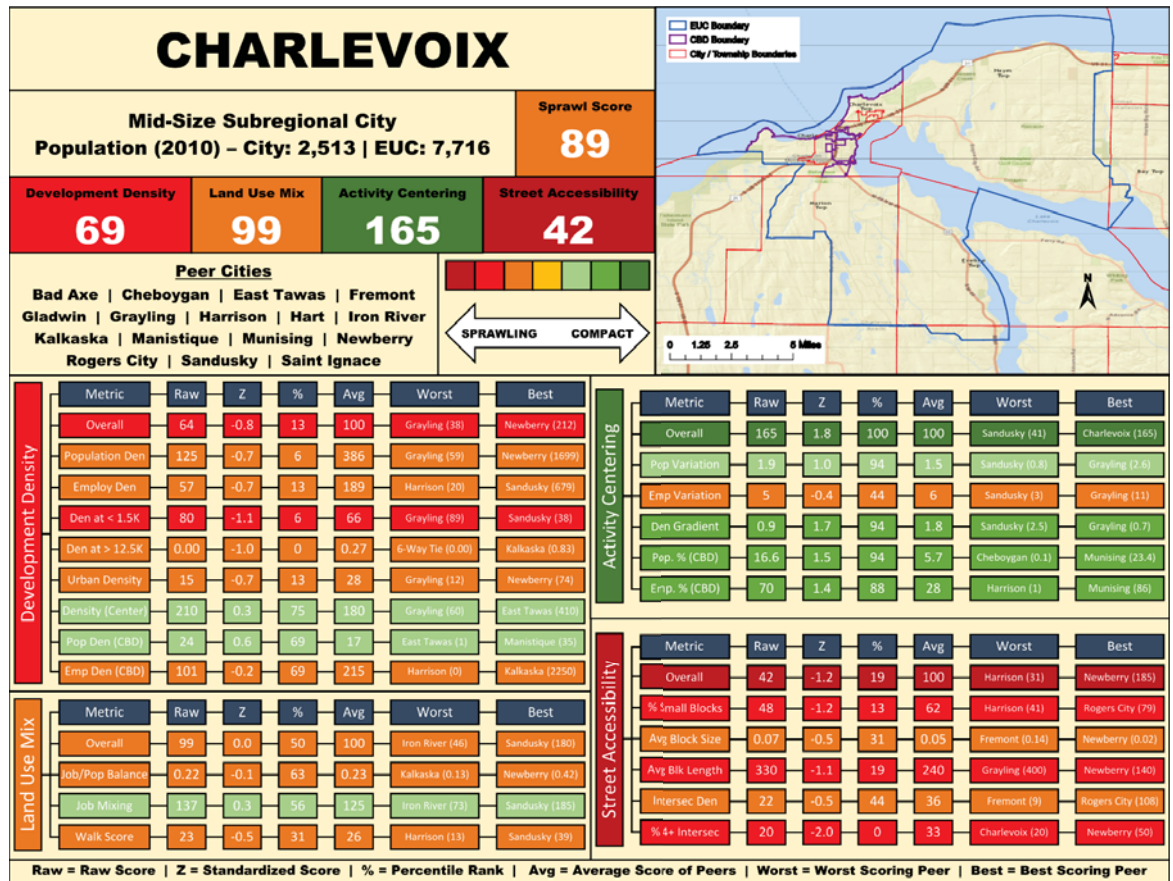


Figure G.9: Cheboygan Sprawl Scorecard

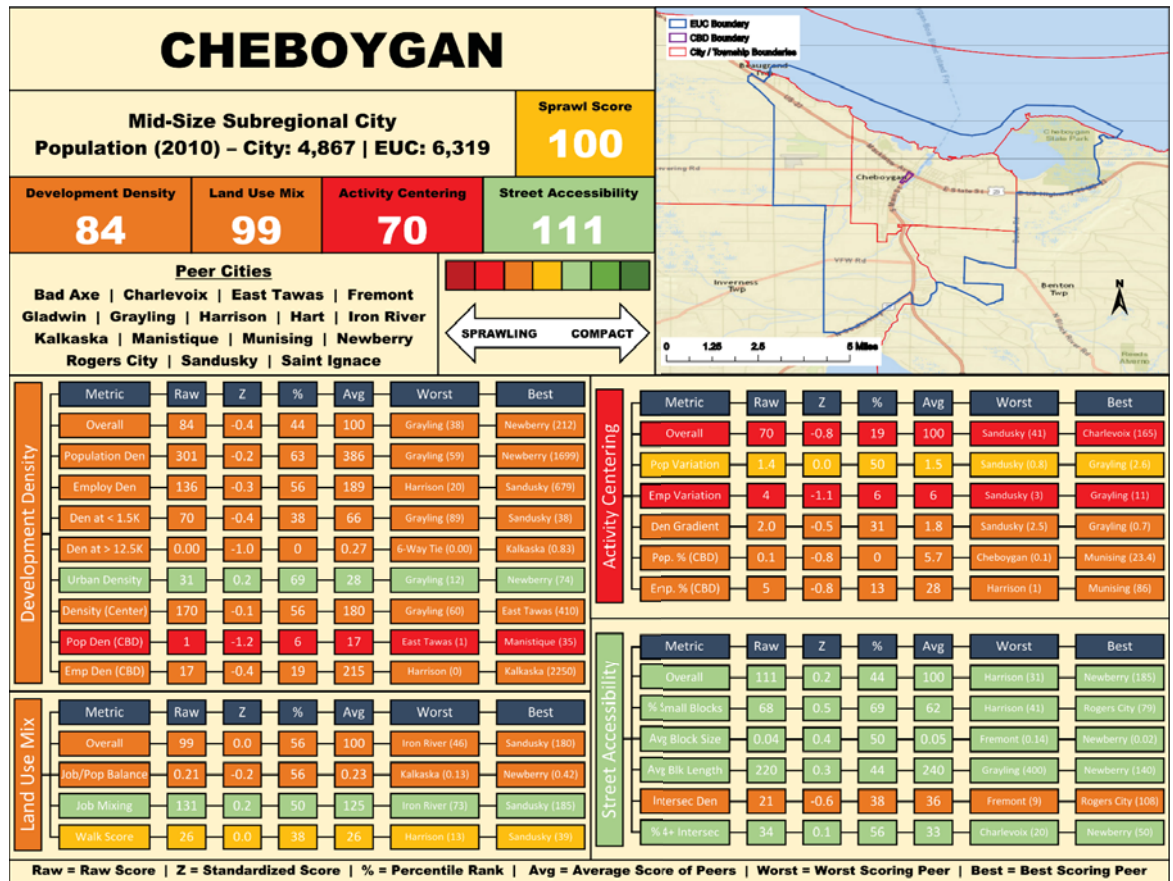


Figure G.10: Coldwater Sprawl Scorecard

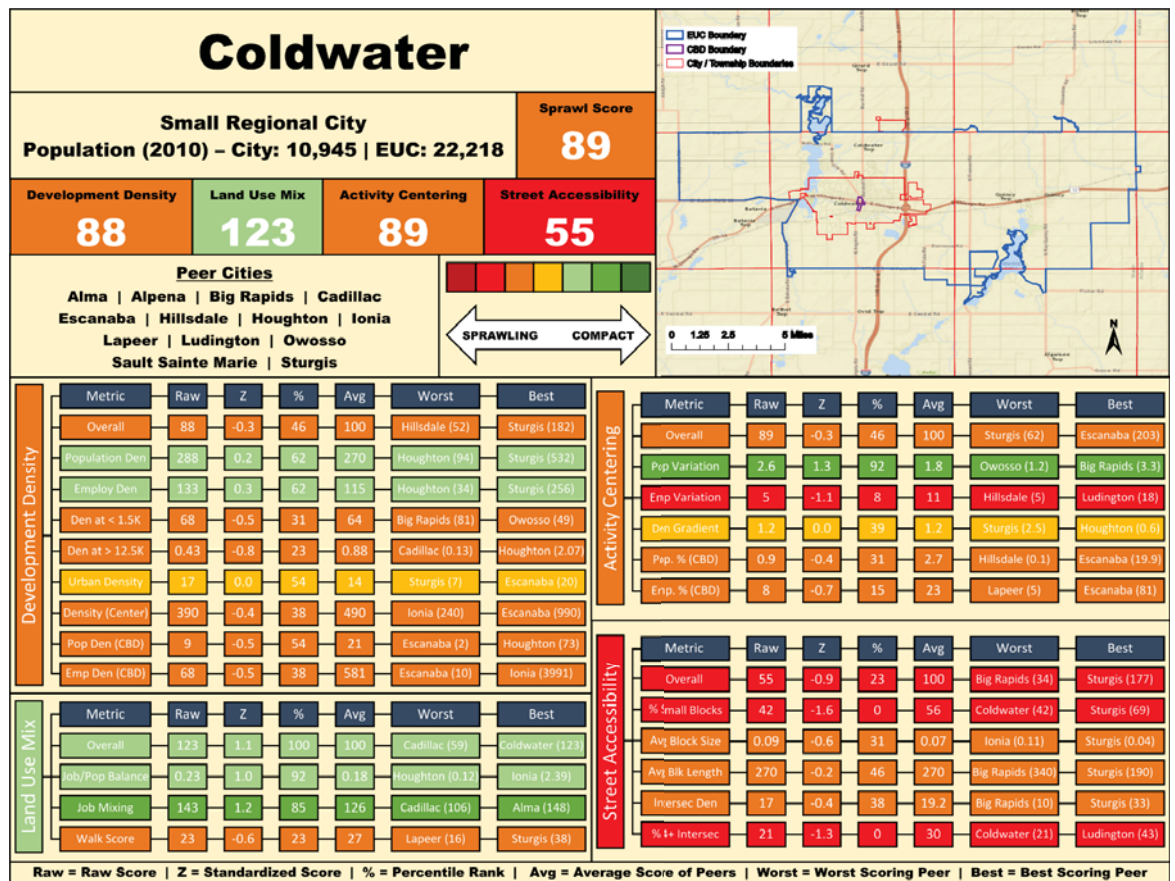


Figure G.11: East Tawas Sprawl Scorecard

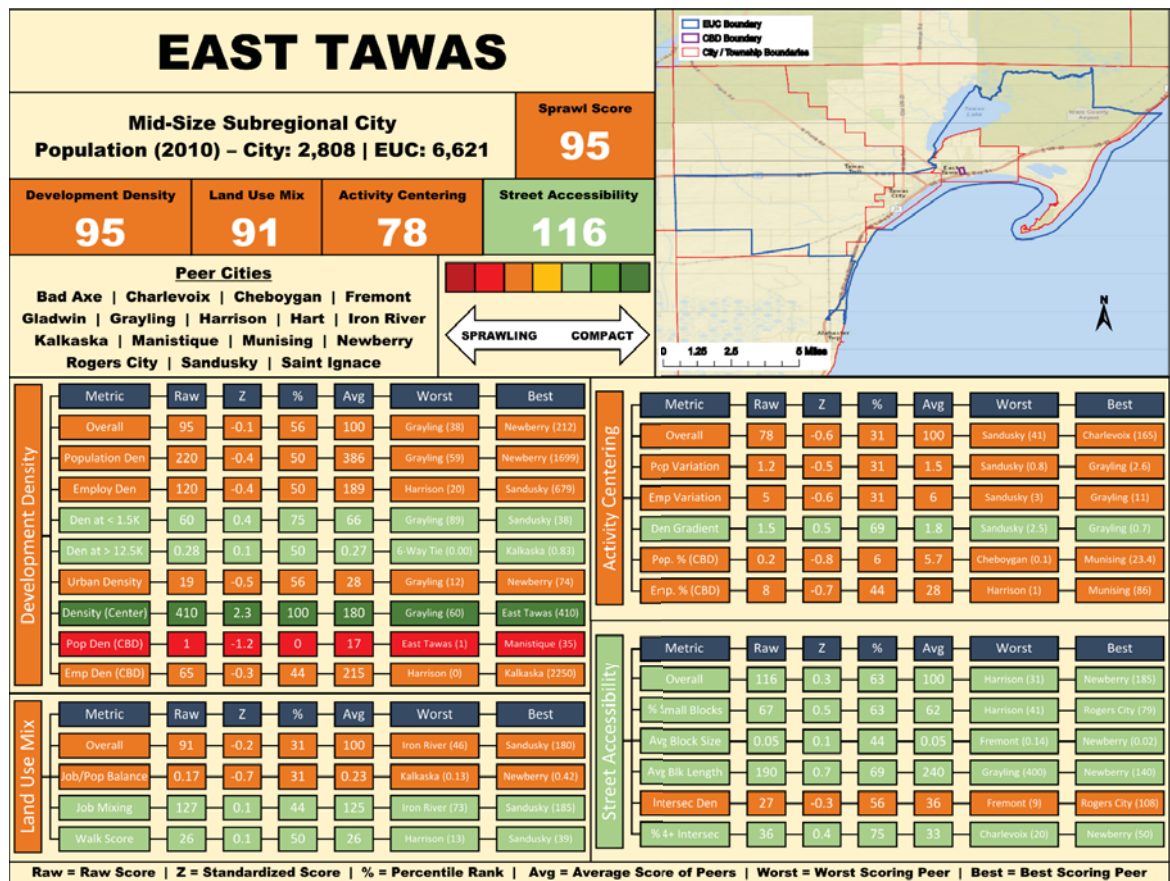


Figure G.12: Escanaba Sprawl Scorecard

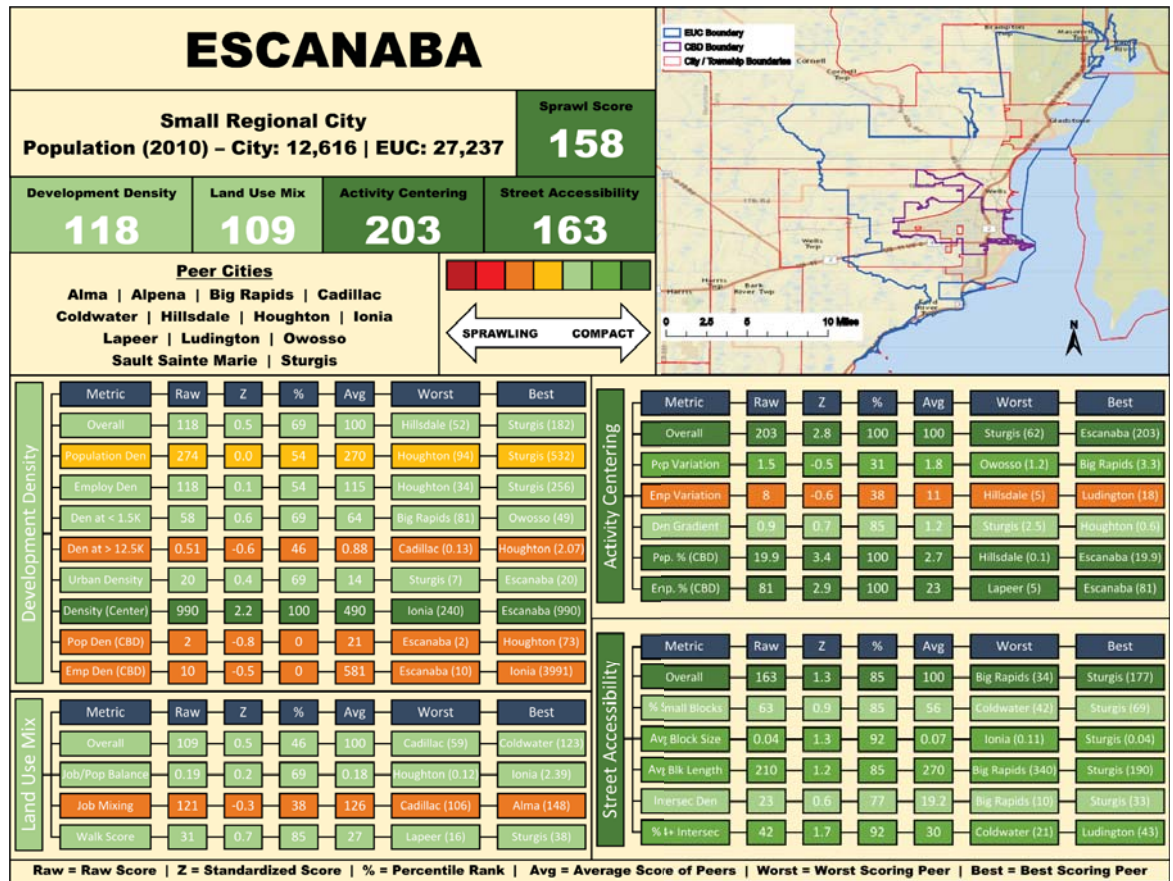


Figure G.13: Fremont Sprawl Scorecard

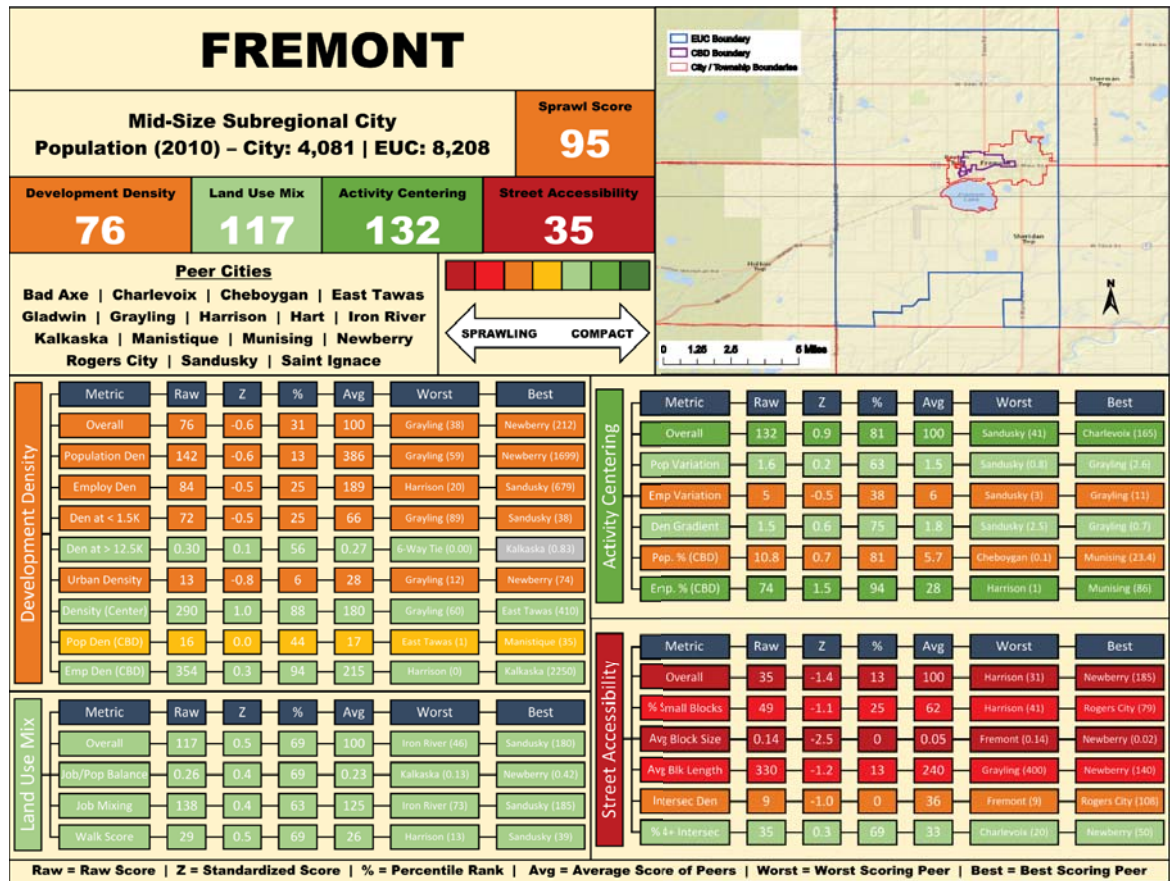


Figure G.14: Gaylord Sprawl Scorecard

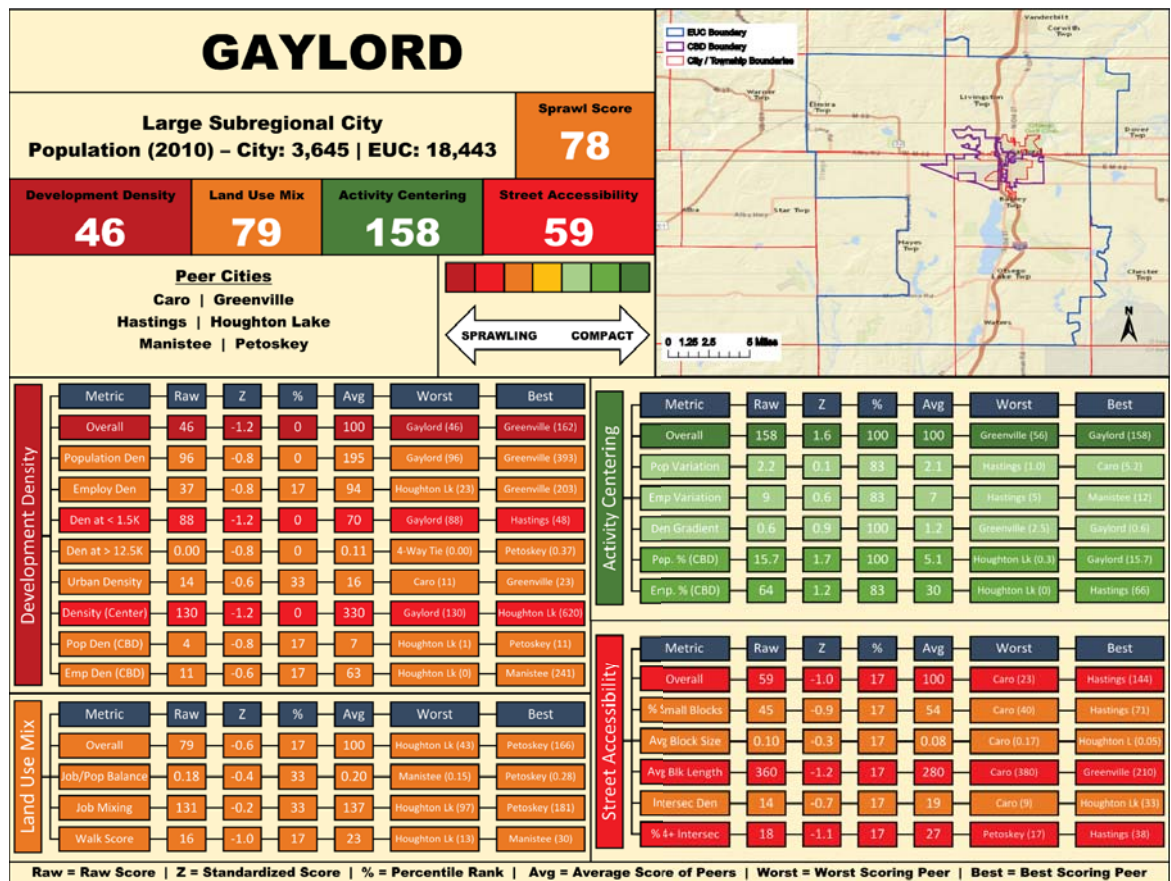


Figure G.15: Gladwin Sprawl Scorecard

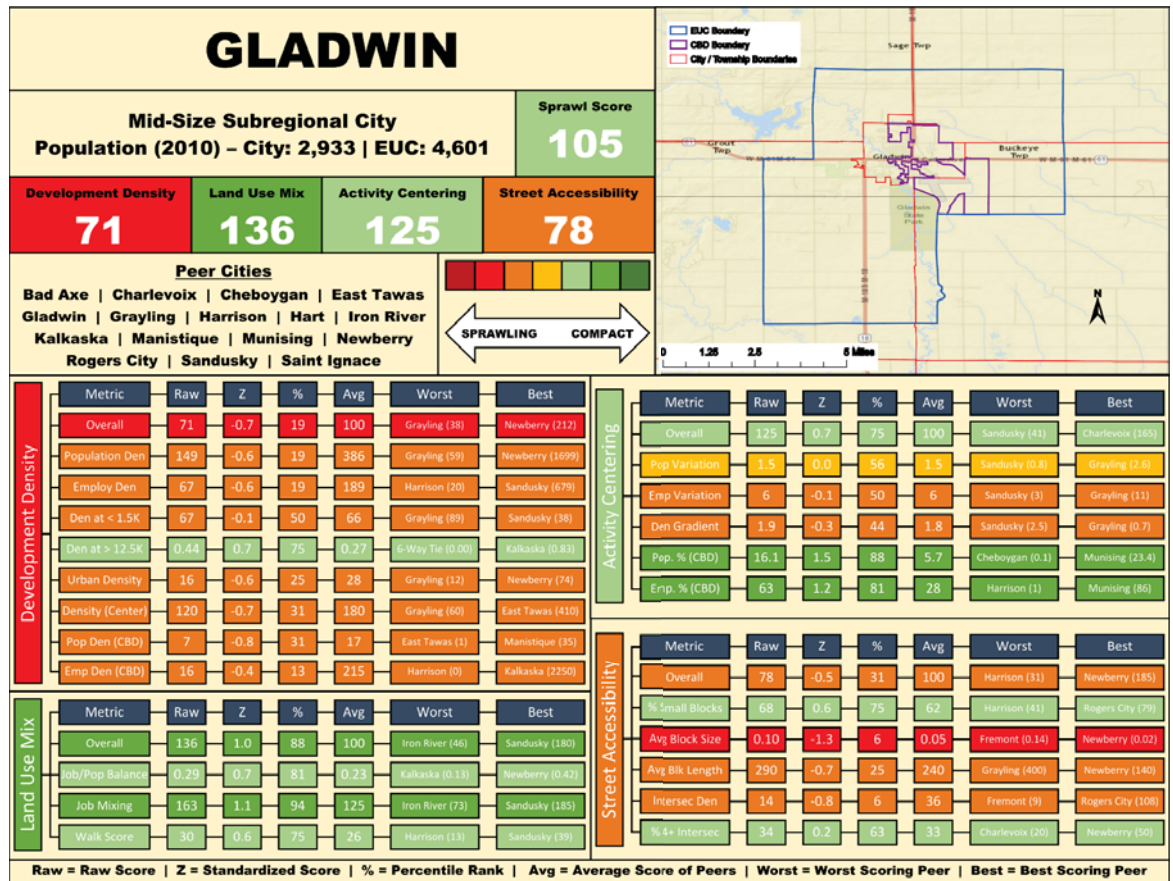


Figure G.16: Grayling Sprawl Scorecard

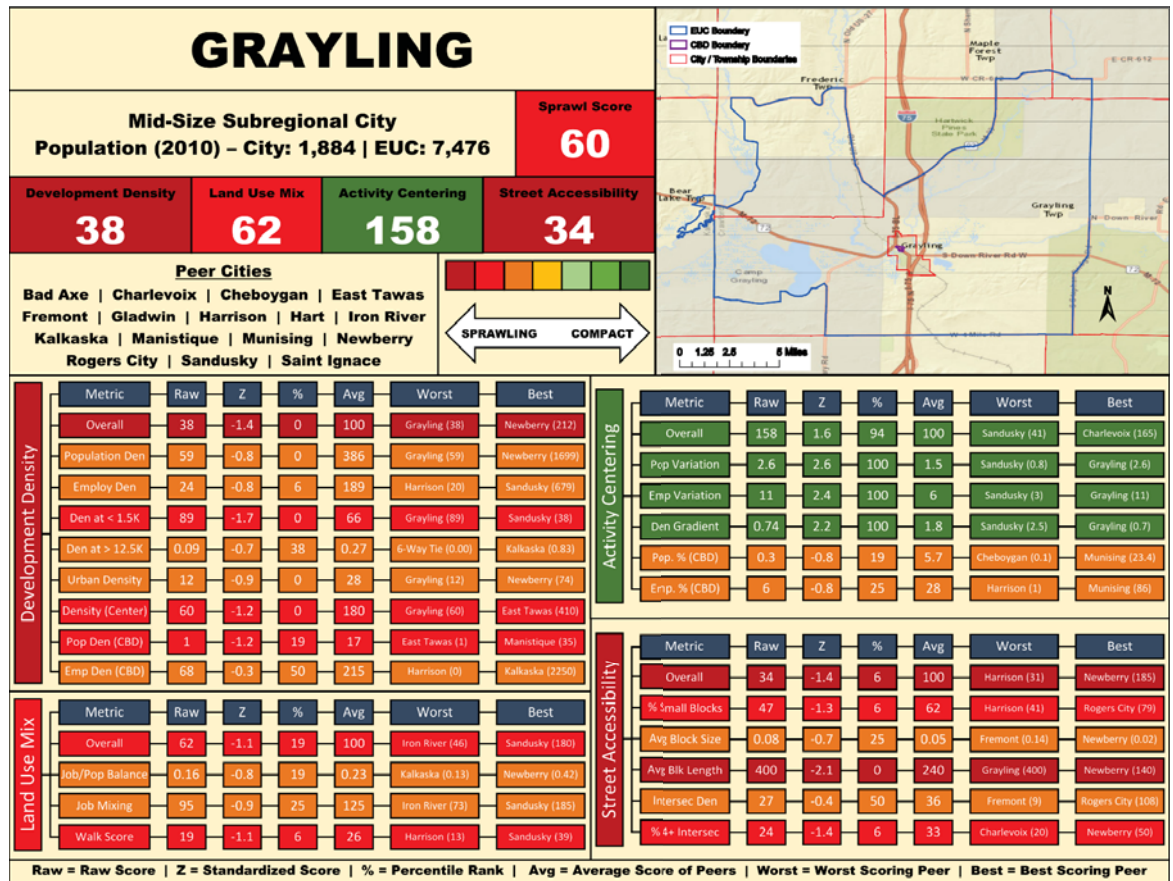


Figure G.17: Greenville Sprawl Scorecard

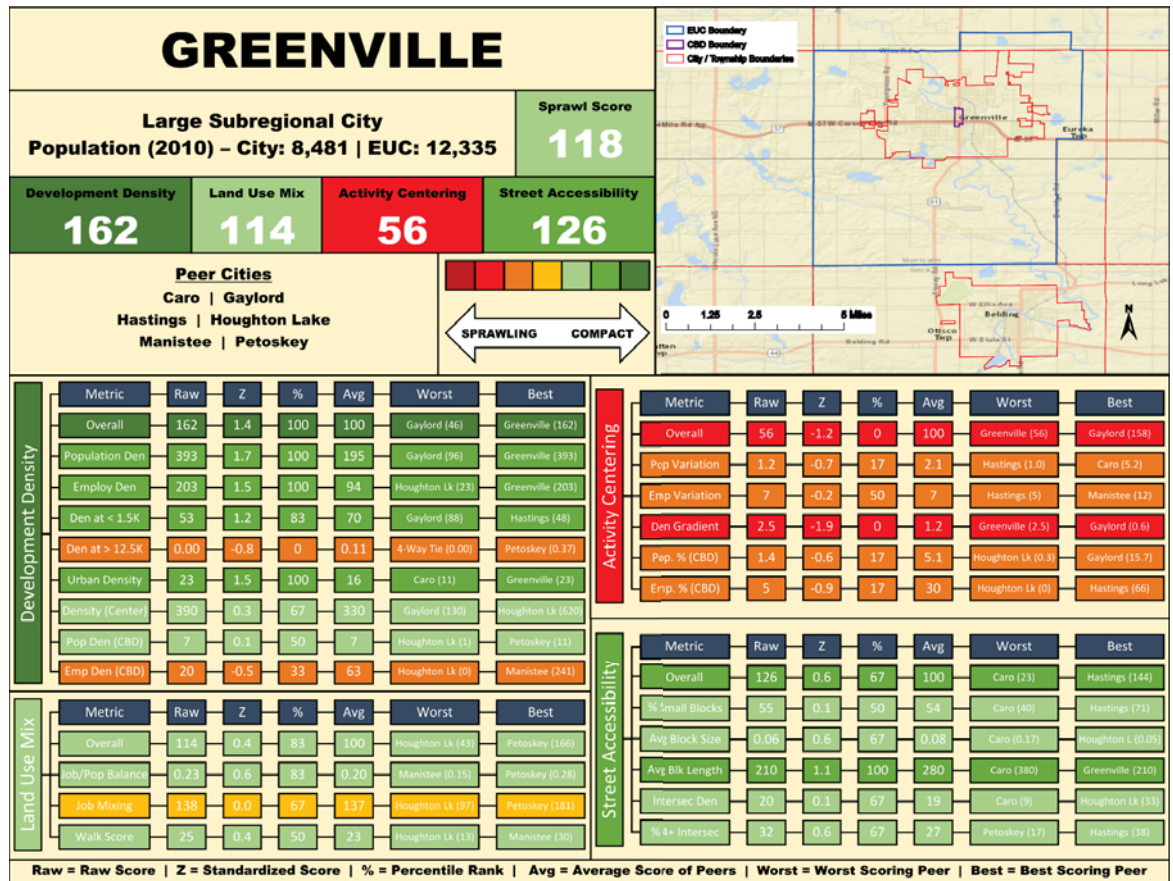


Figure G.18: Harrison Sprawl Scorecard

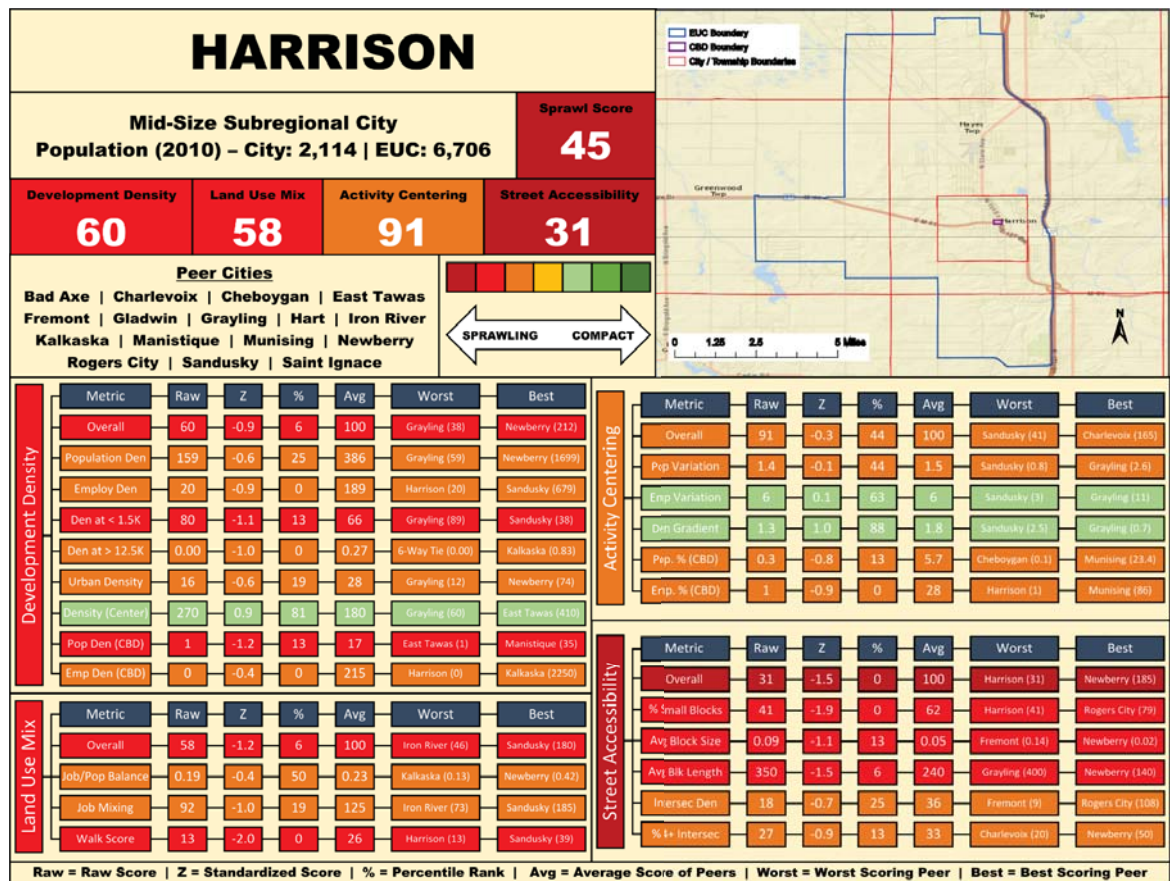


Figure G.19: Hart Sprawl Scorecard

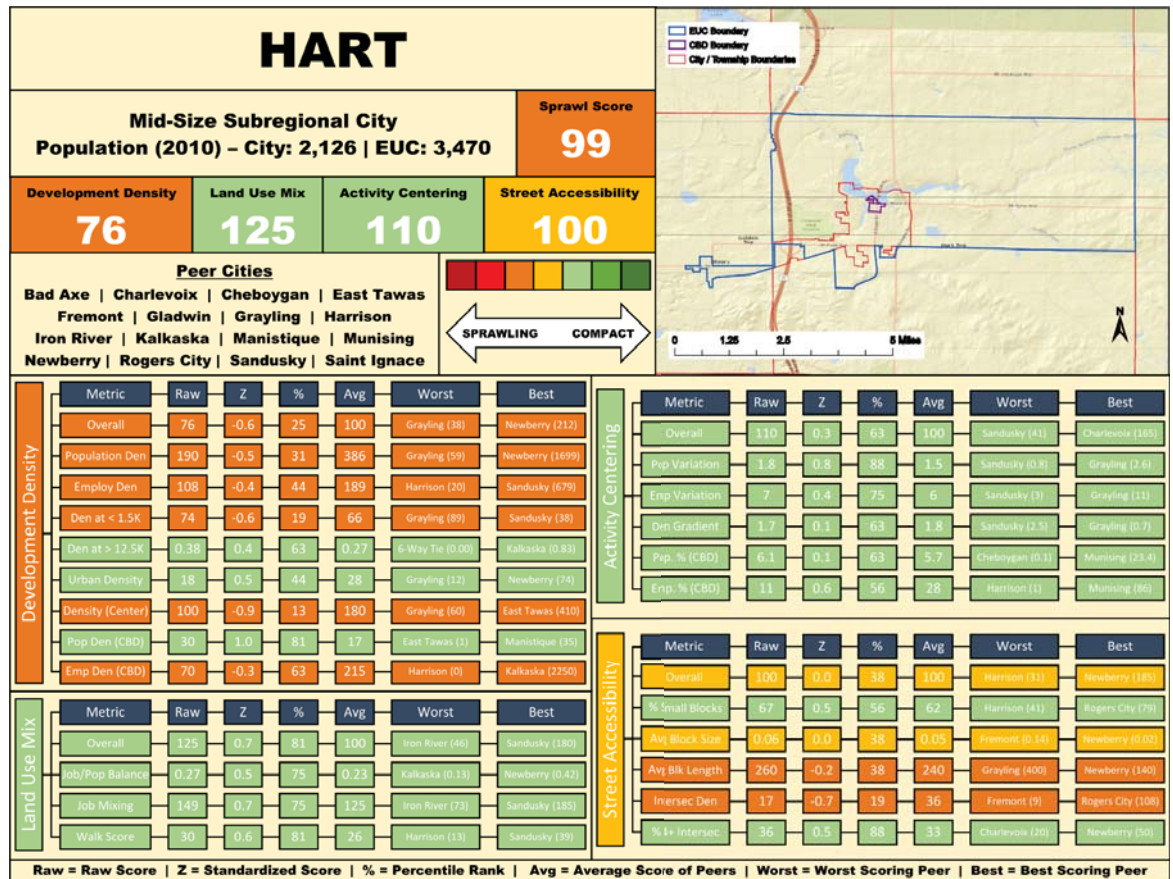


Figure G.20: Hastings Sprawl Scorecard

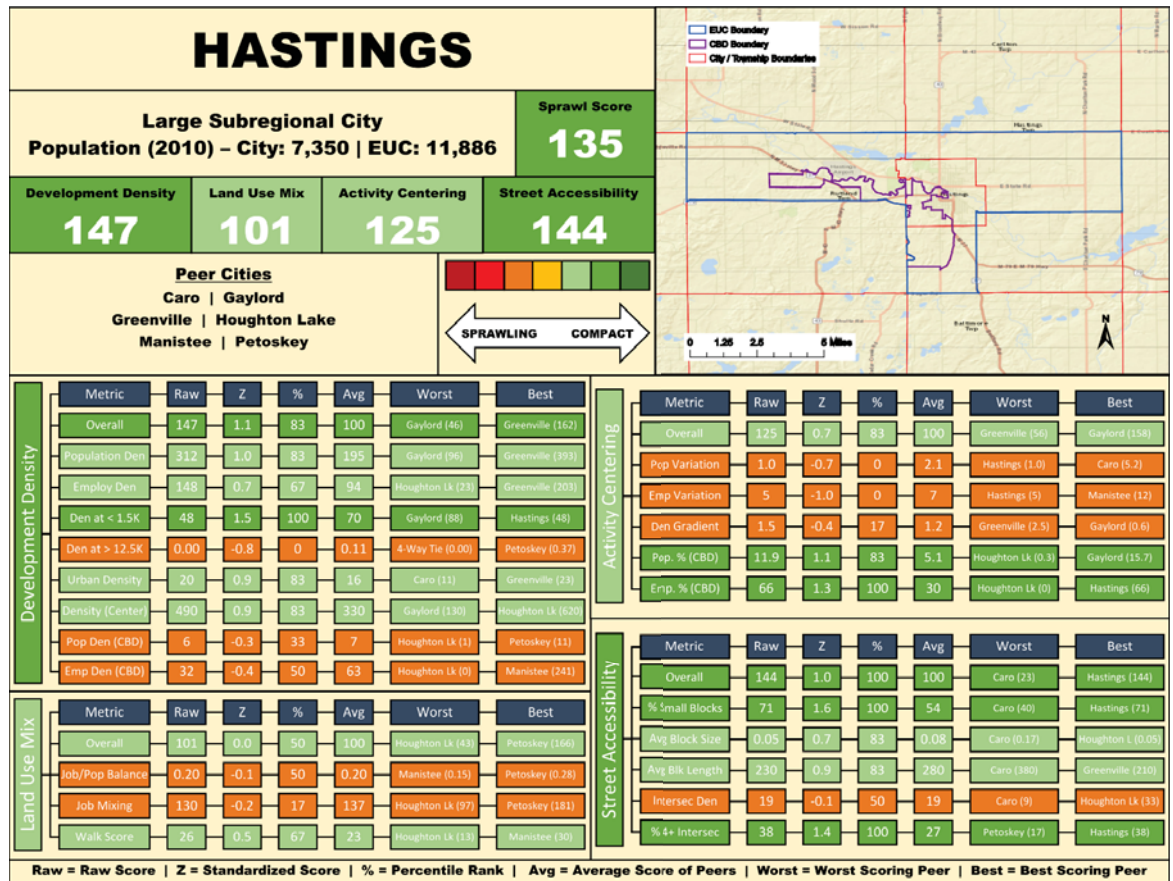


Figure G.21: Hillsdale Sprawl Scorecard

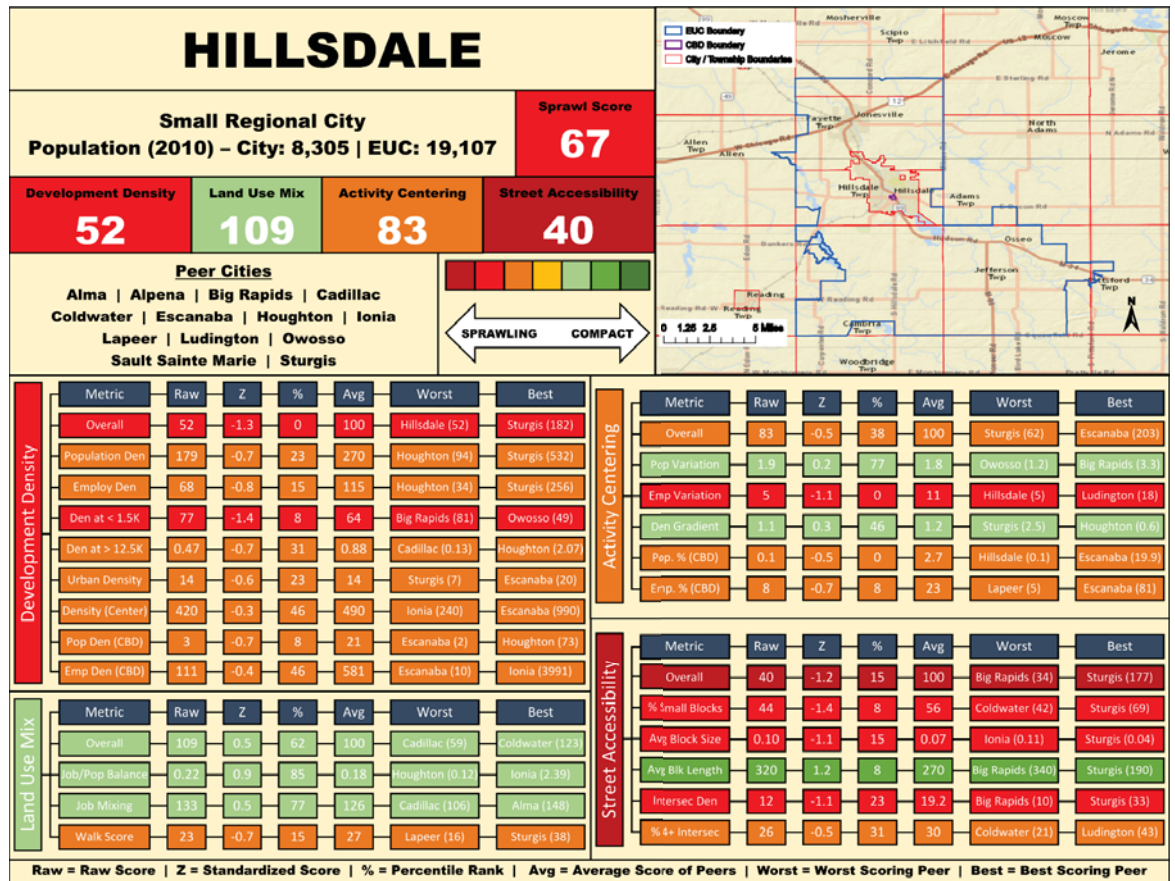


Figure G.22: Houghton Sprawl Scorecard

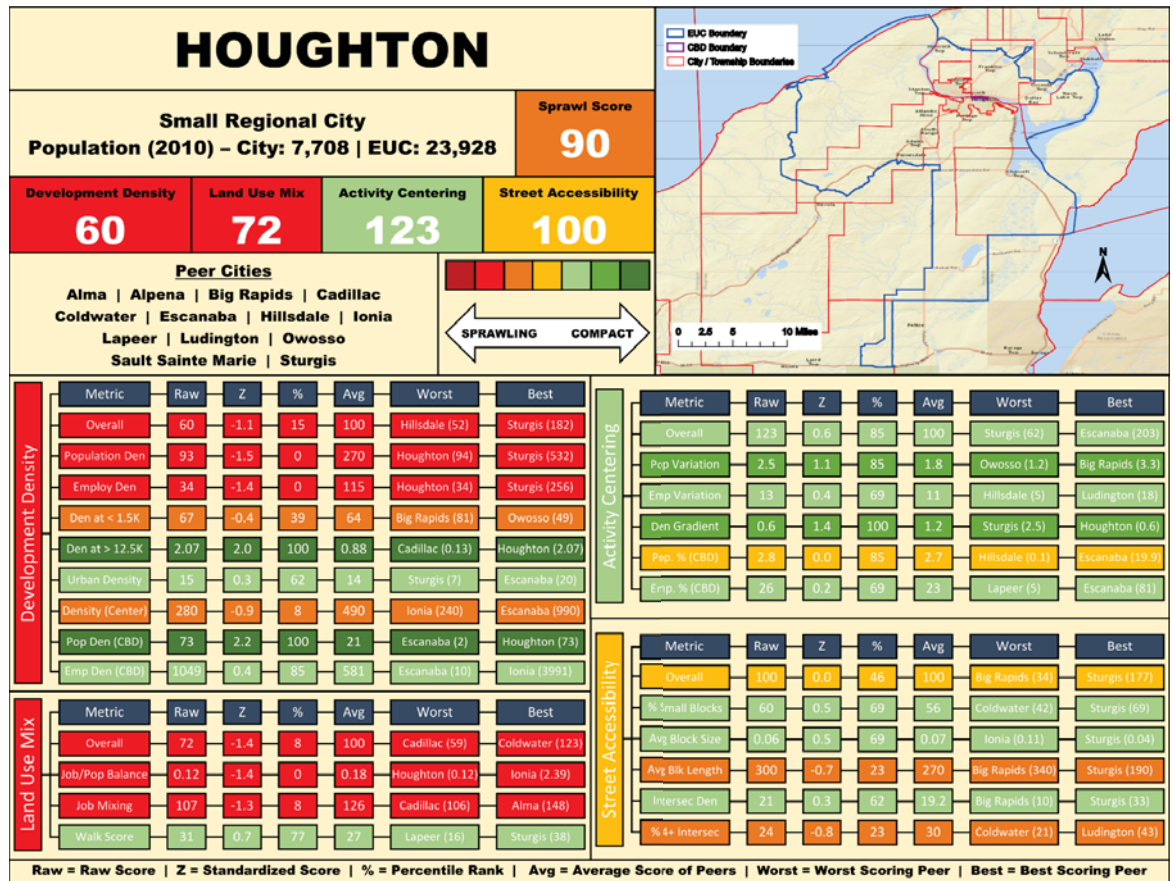


Figure G.23: Houghton Lake Sprawl Scorecard

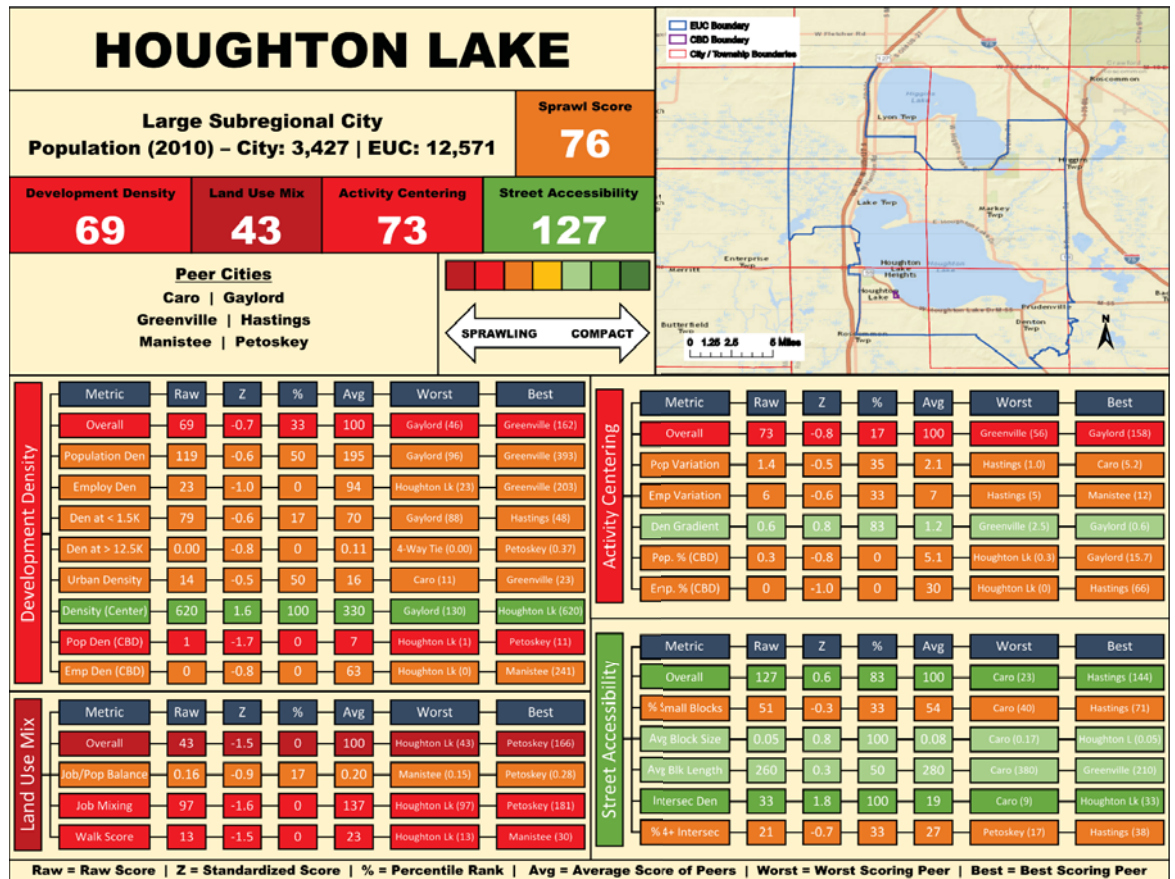


Figure G.24: Ionia Sprawl Scorecard

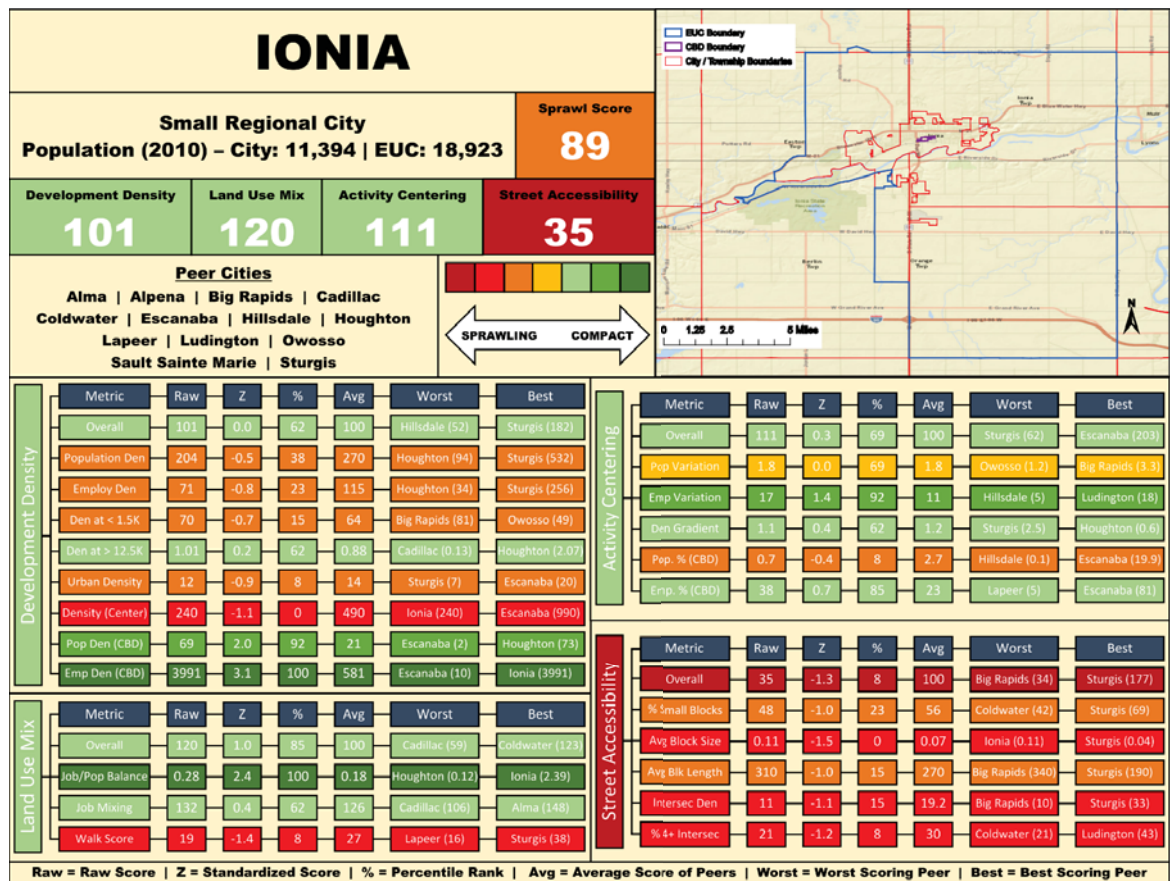


Figure G.25: Iron River Sprawl Scorecard

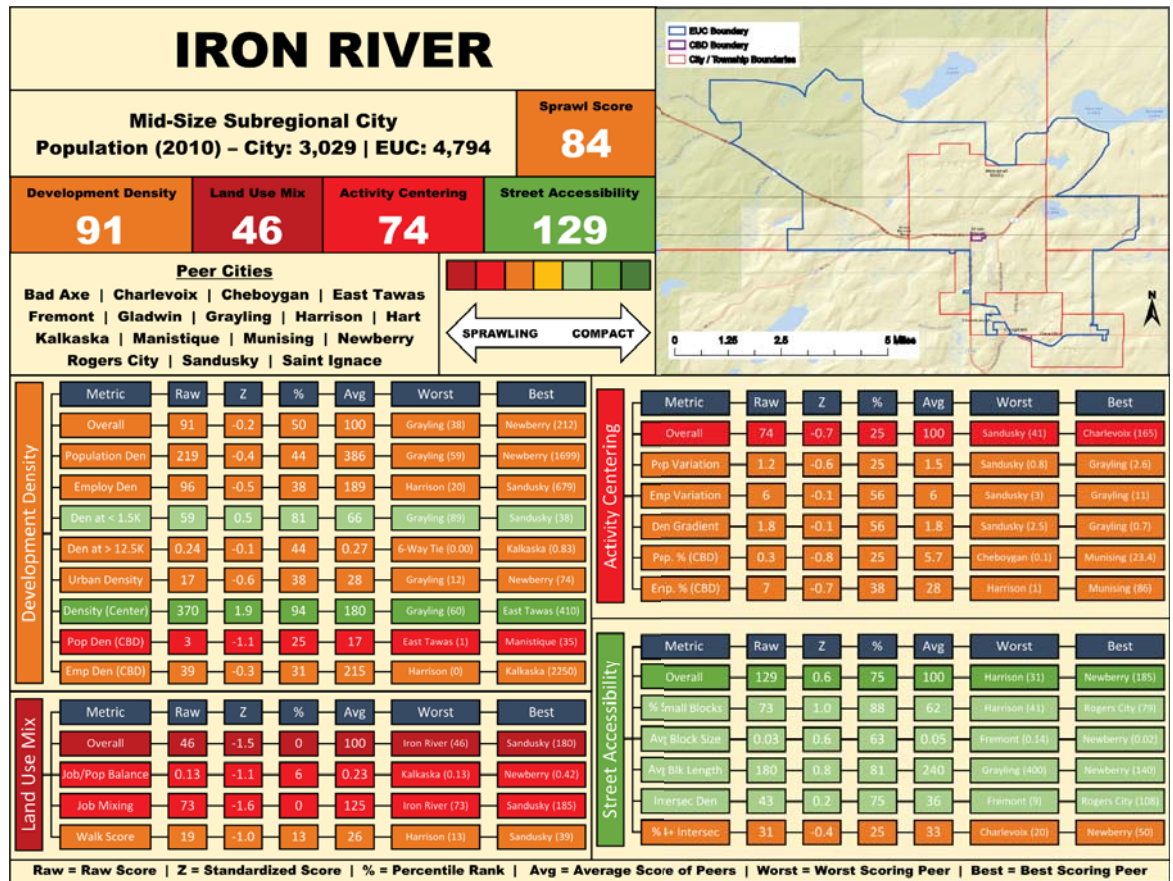


Figure G.26: Kalkaska Sprawl Scorecard

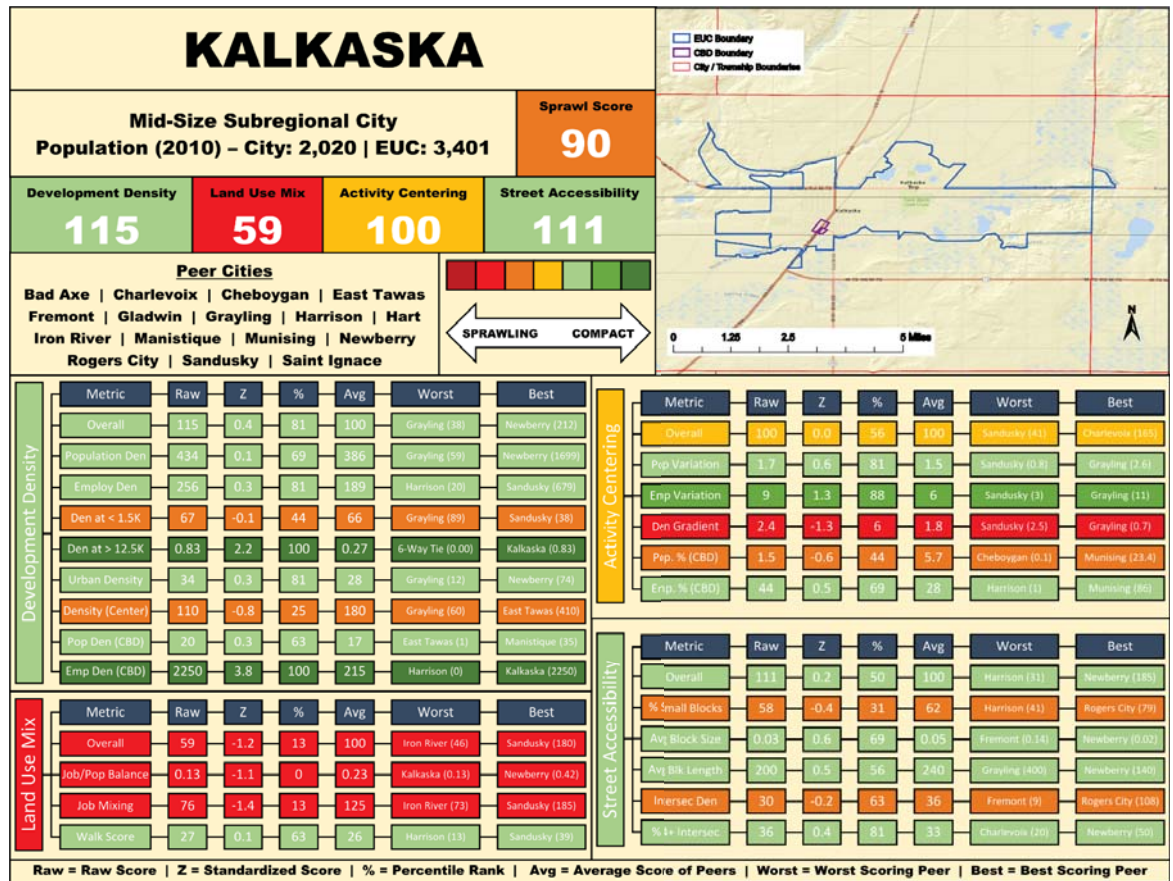


Figure G.27: Lapeer Sprawl Scorecard

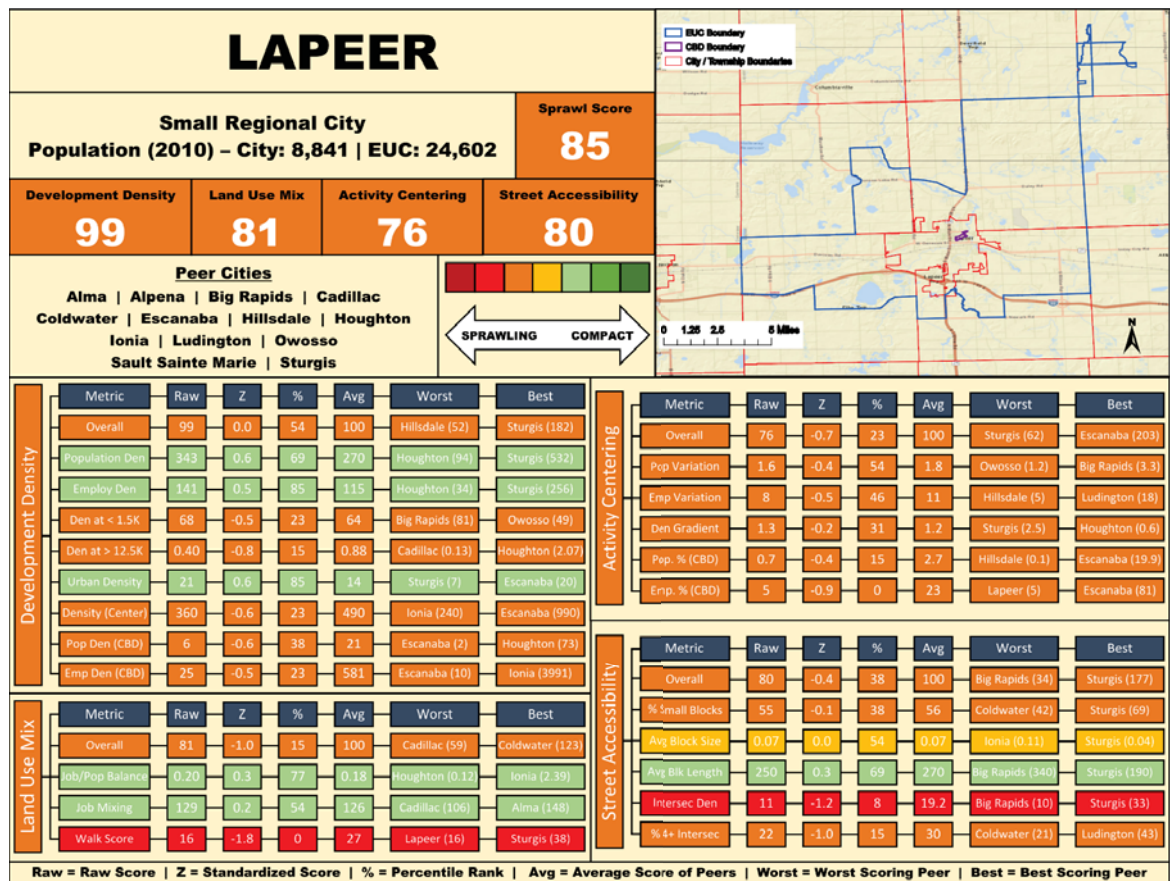


Figure G.28: Ludington Sprawl Scorecard

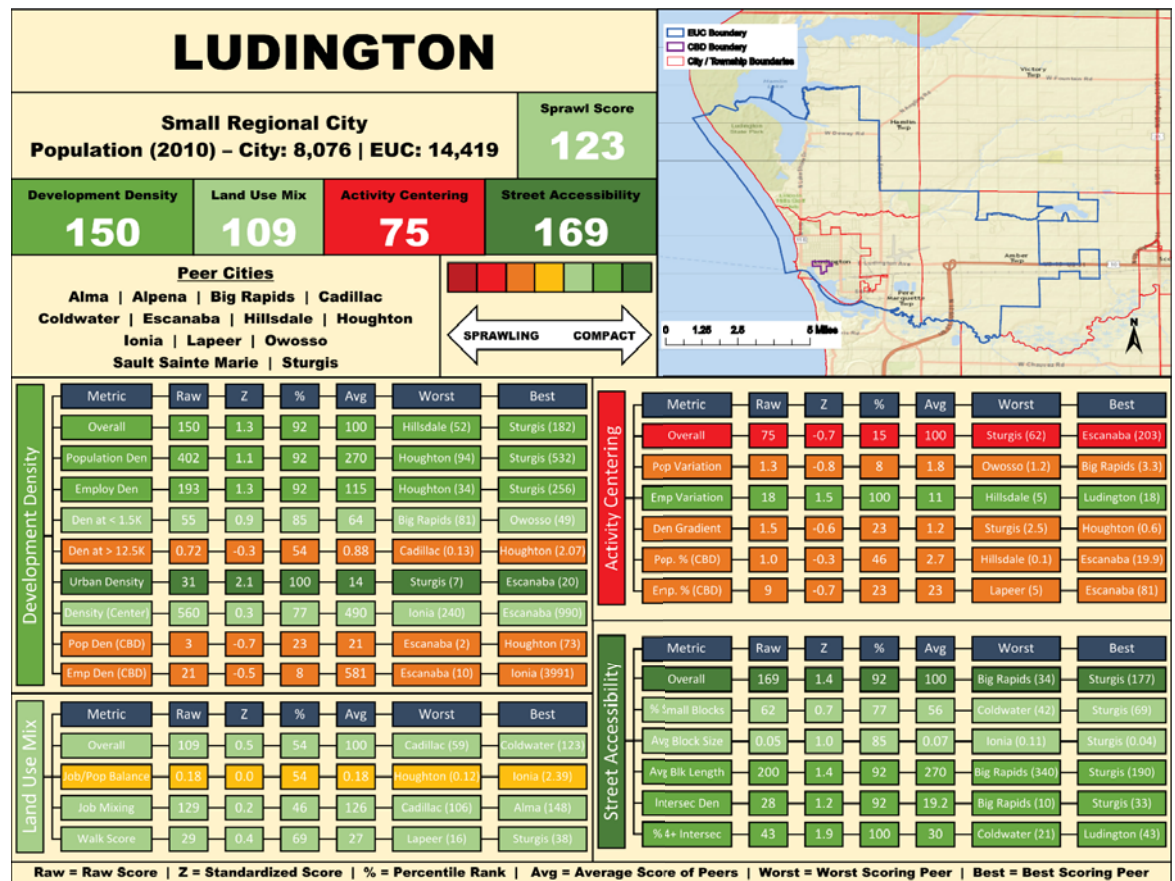


Figure G.29: Manistee Sprawl Scorecard

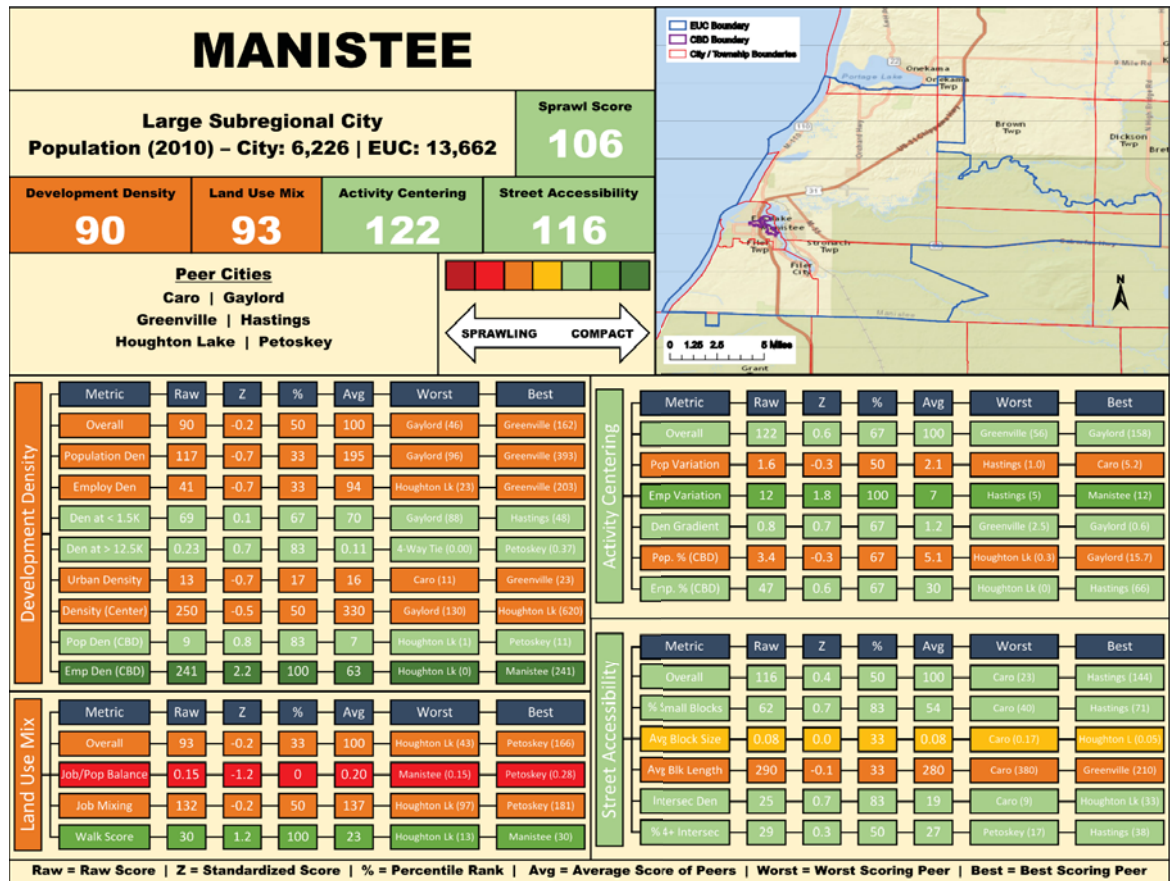


Figure G.30: Manistique Sprawl Scorecard

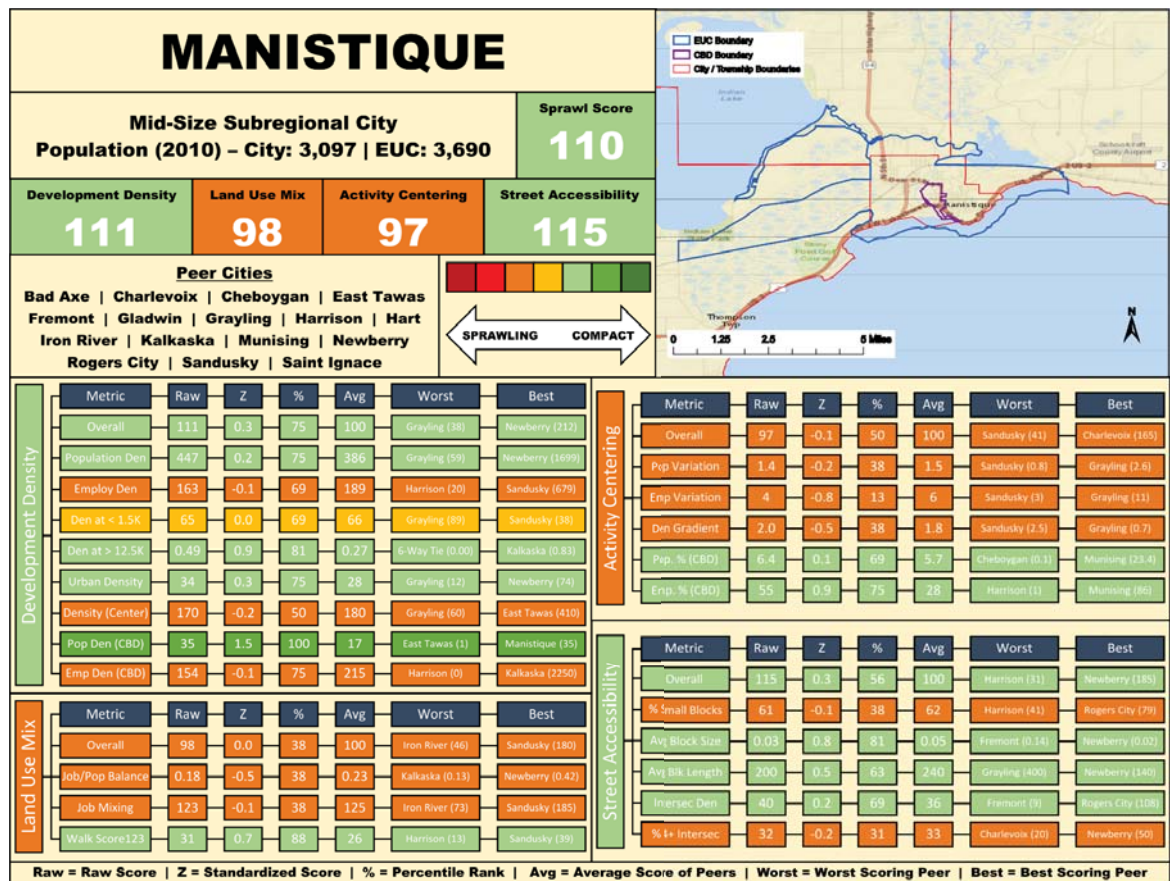


Figure G.3.1: Marquette Sprawl Scorecard

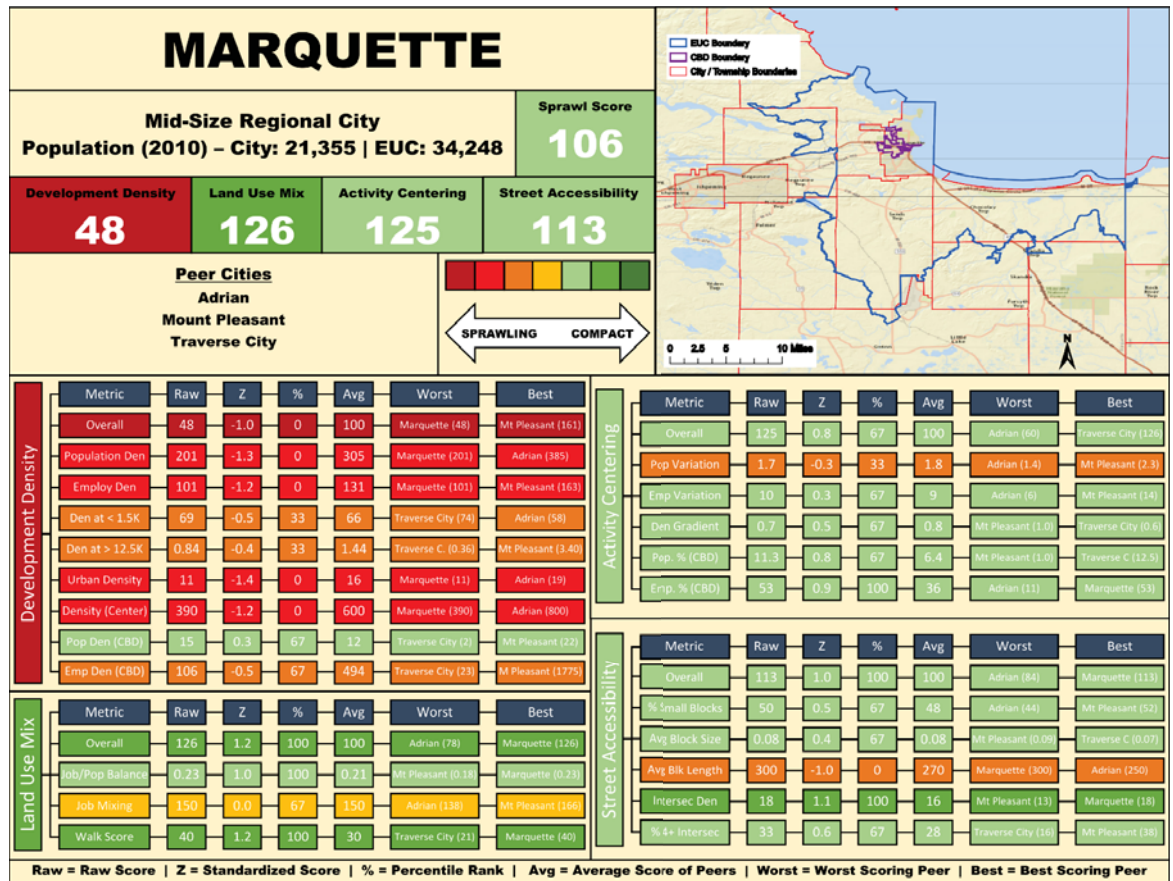


Figure G.32: Mount Pleasant Sprawl Scorecard

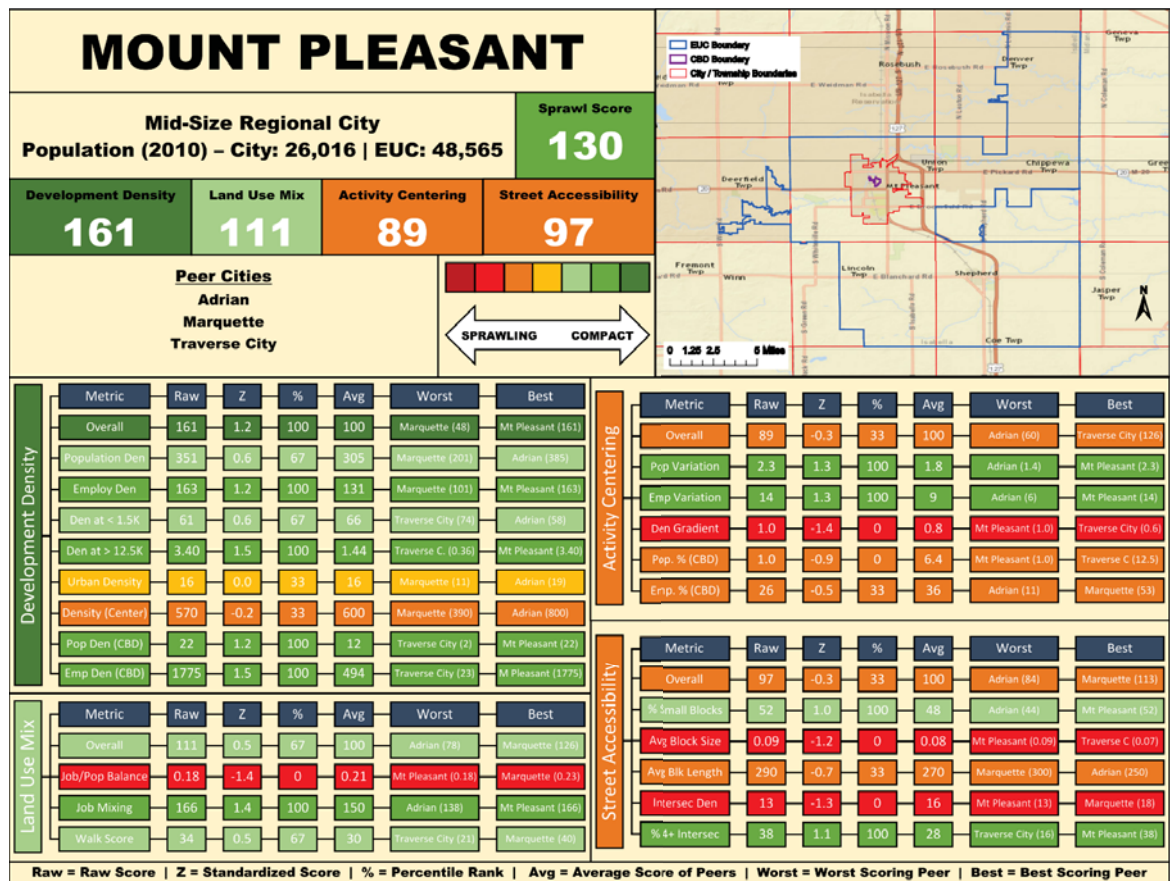


Figure G.33: Munising Sprawl Scorecard

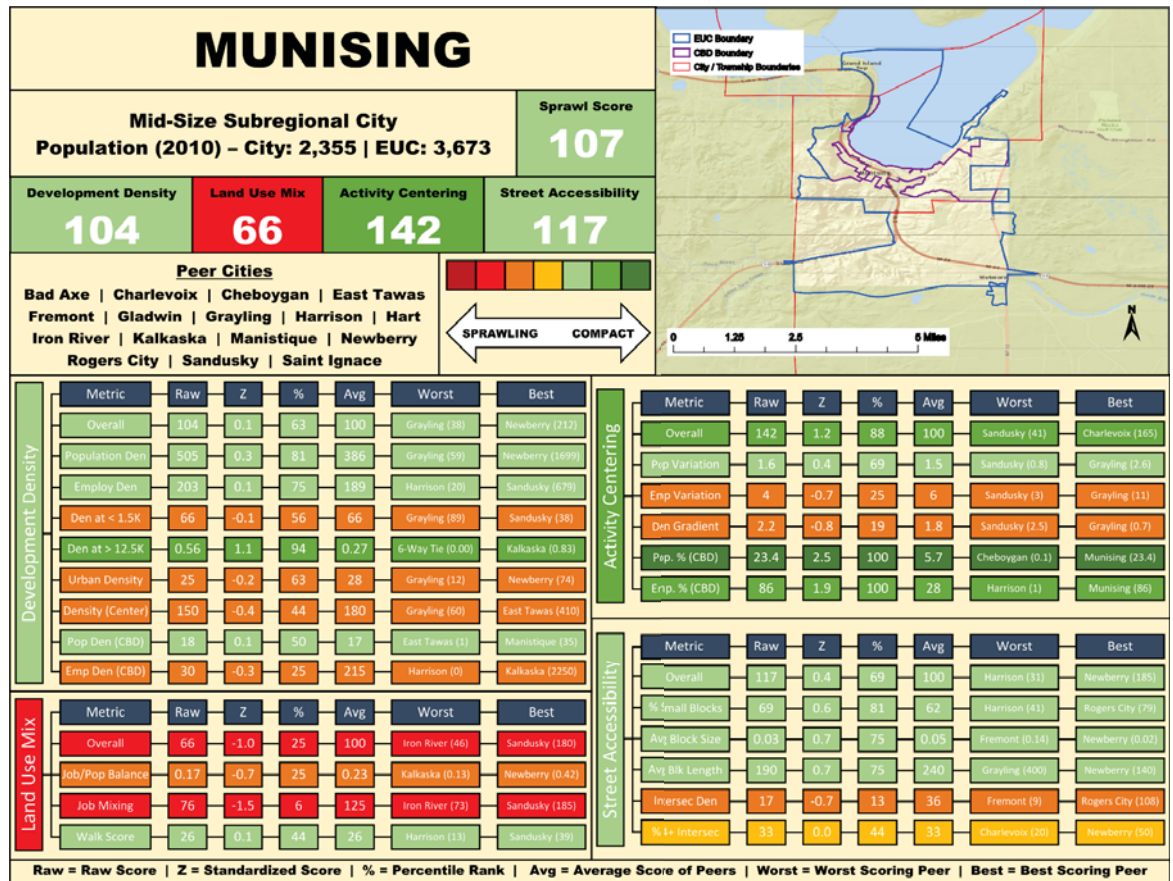
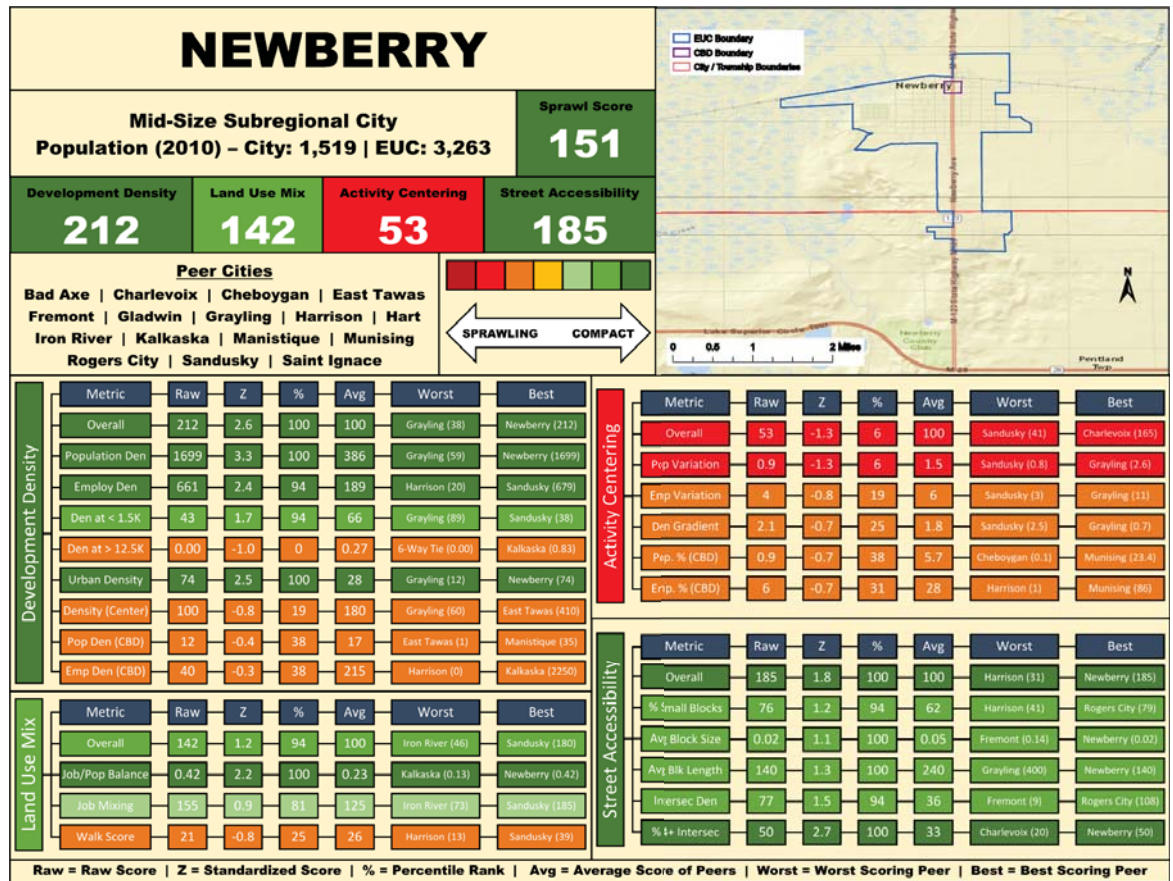


Figure G.34: Newberry Sprawl Scorecard



Raw = Raw Score | Z = Standardized Score | % = Percentile Rank | Avg = Average Score of Peers | Worst = Worst Scoring Peer | Best = Best Scoring Peer

Figure G.35: Owosso Sprawl Scorecard

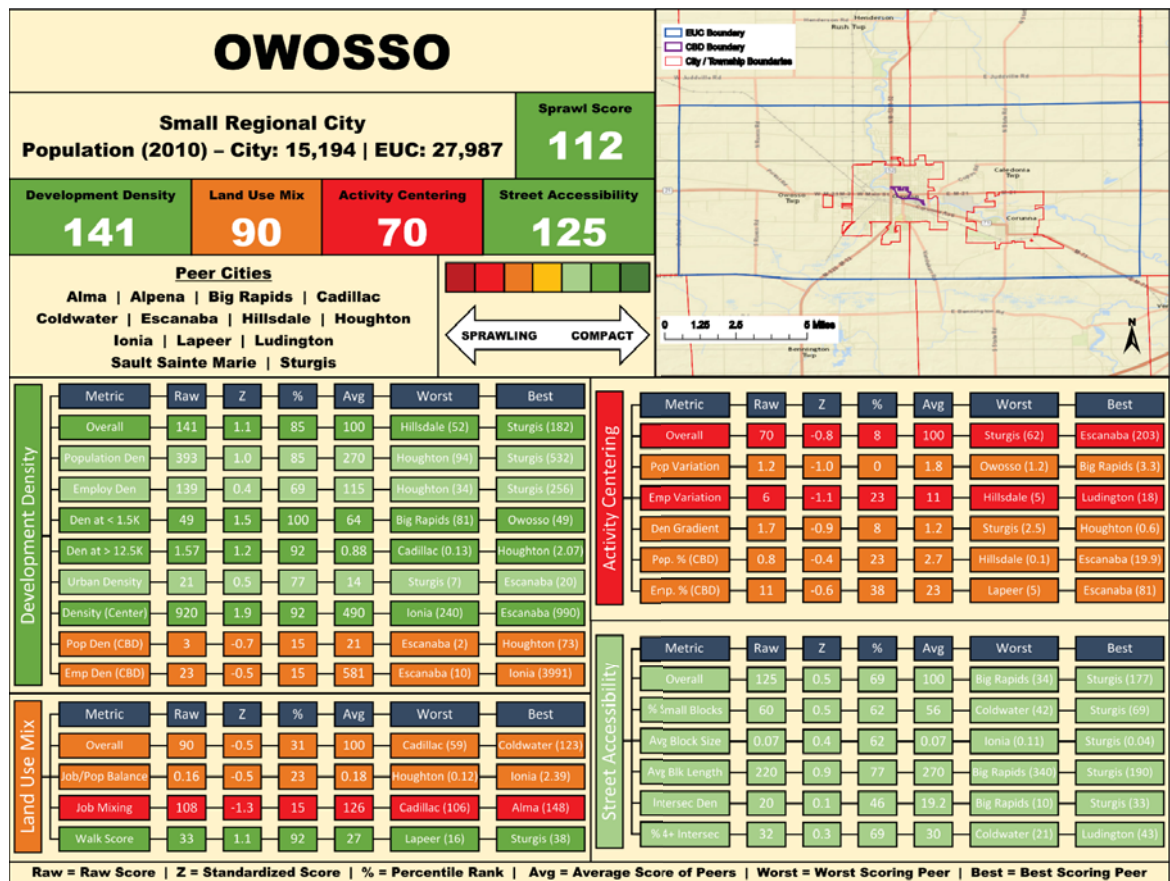


Figure G.36: Petoskey Sprawl Scorecard

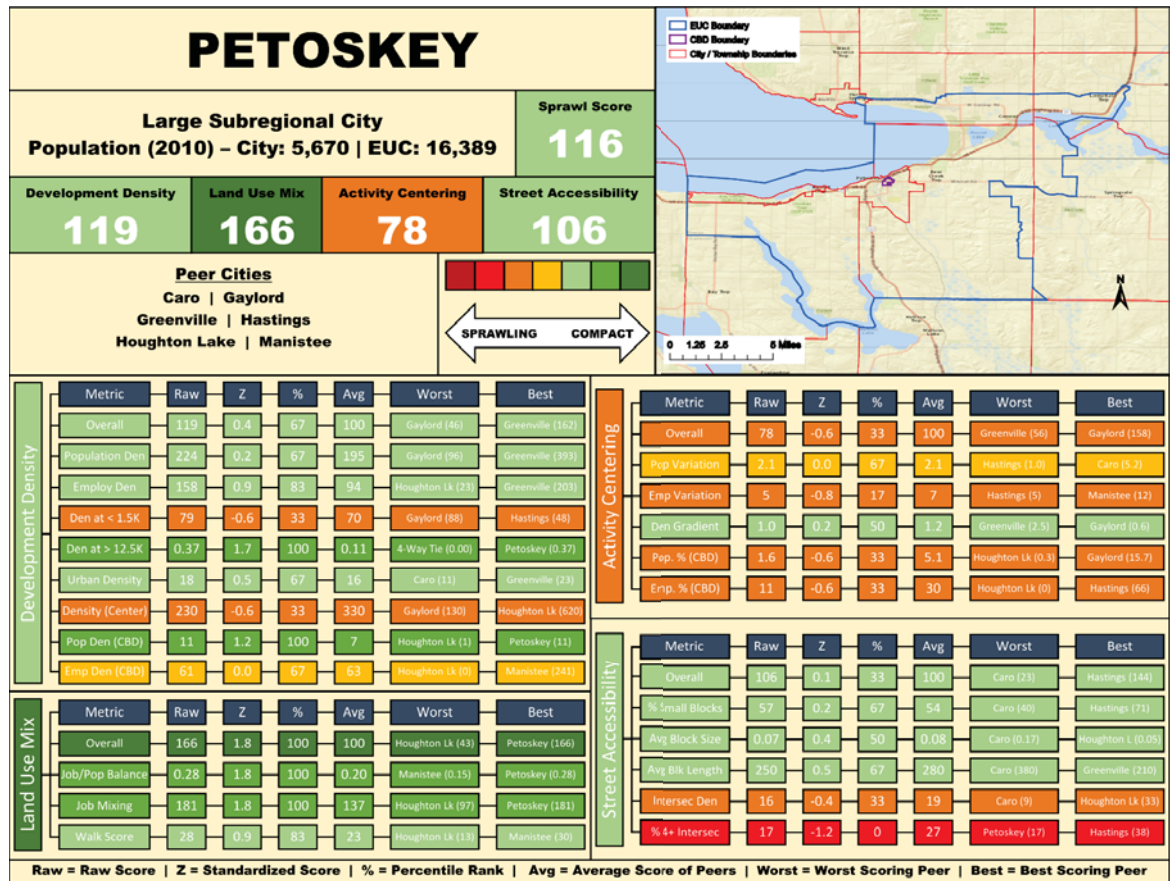


Figure G.37: Rogers City Sprawl Scorecard

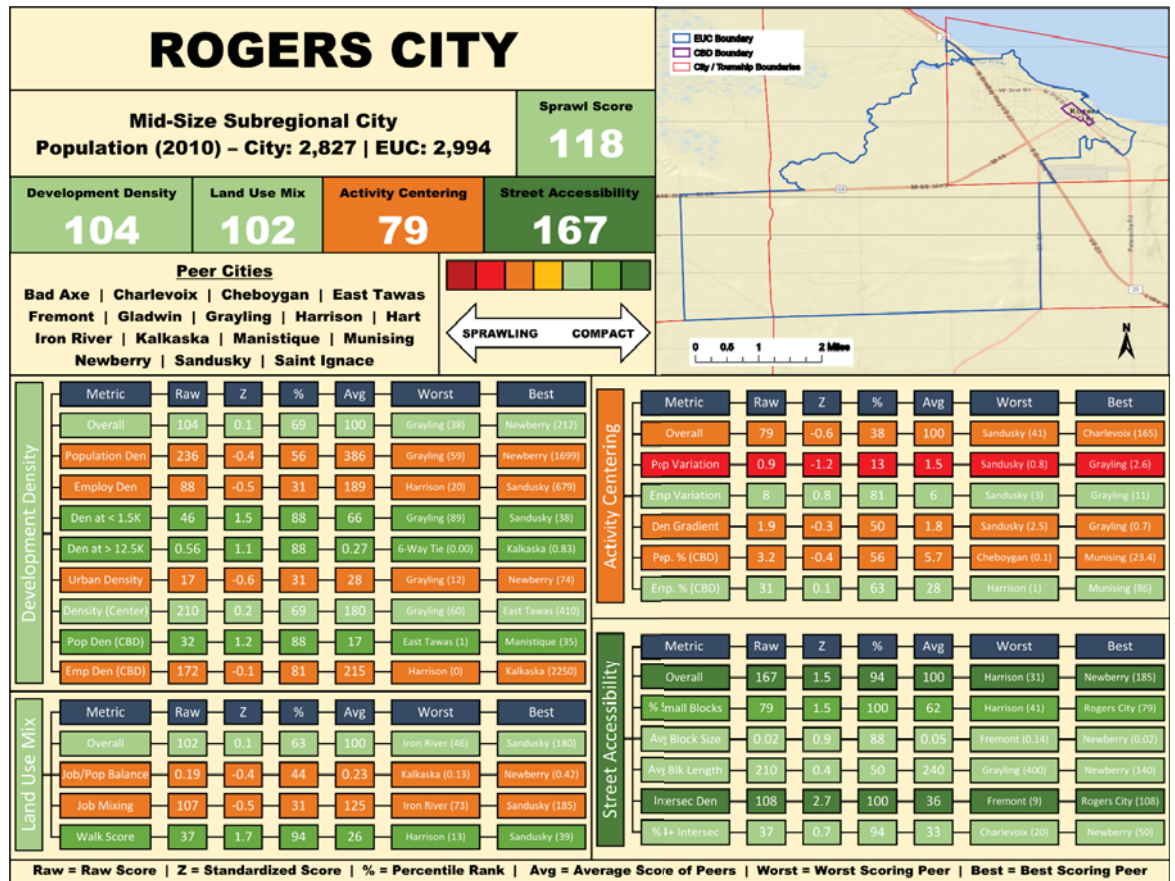


Figure G.38: Saint Ignace Sprawl Scorecard

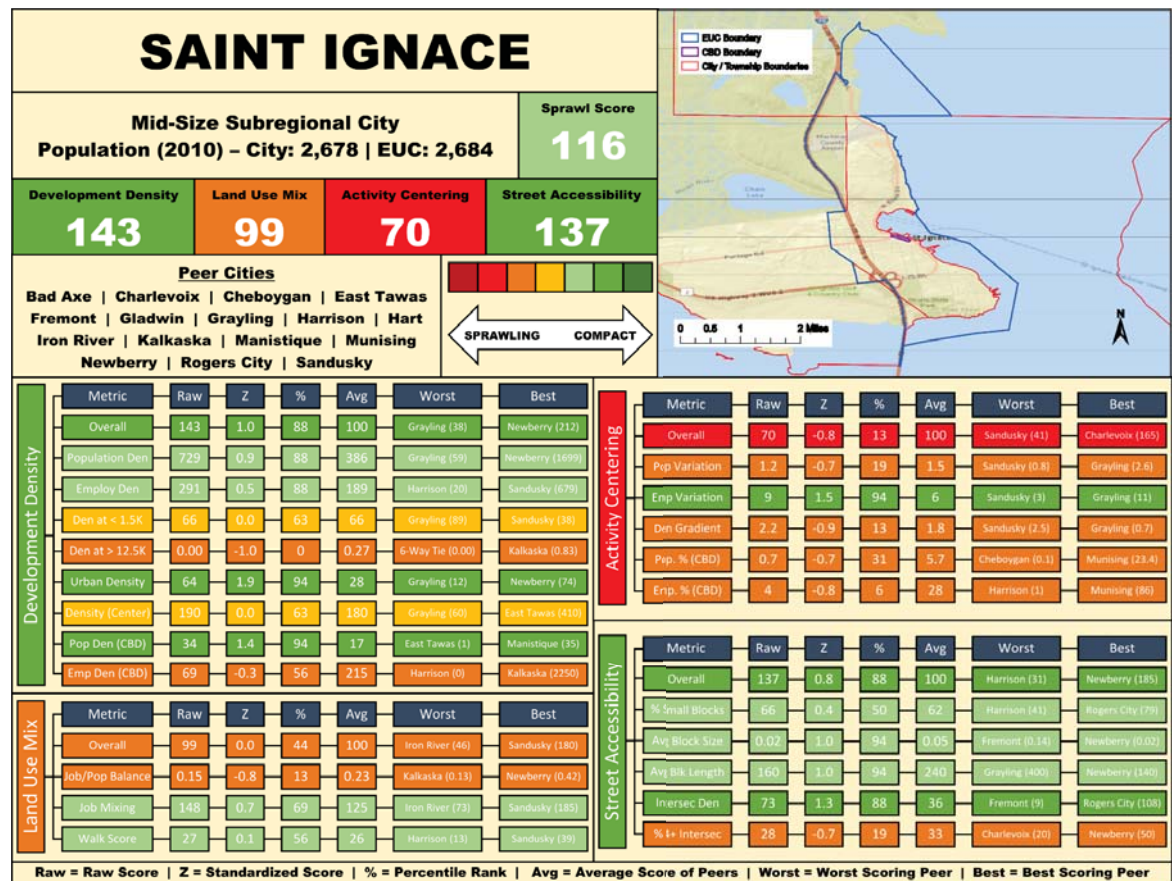
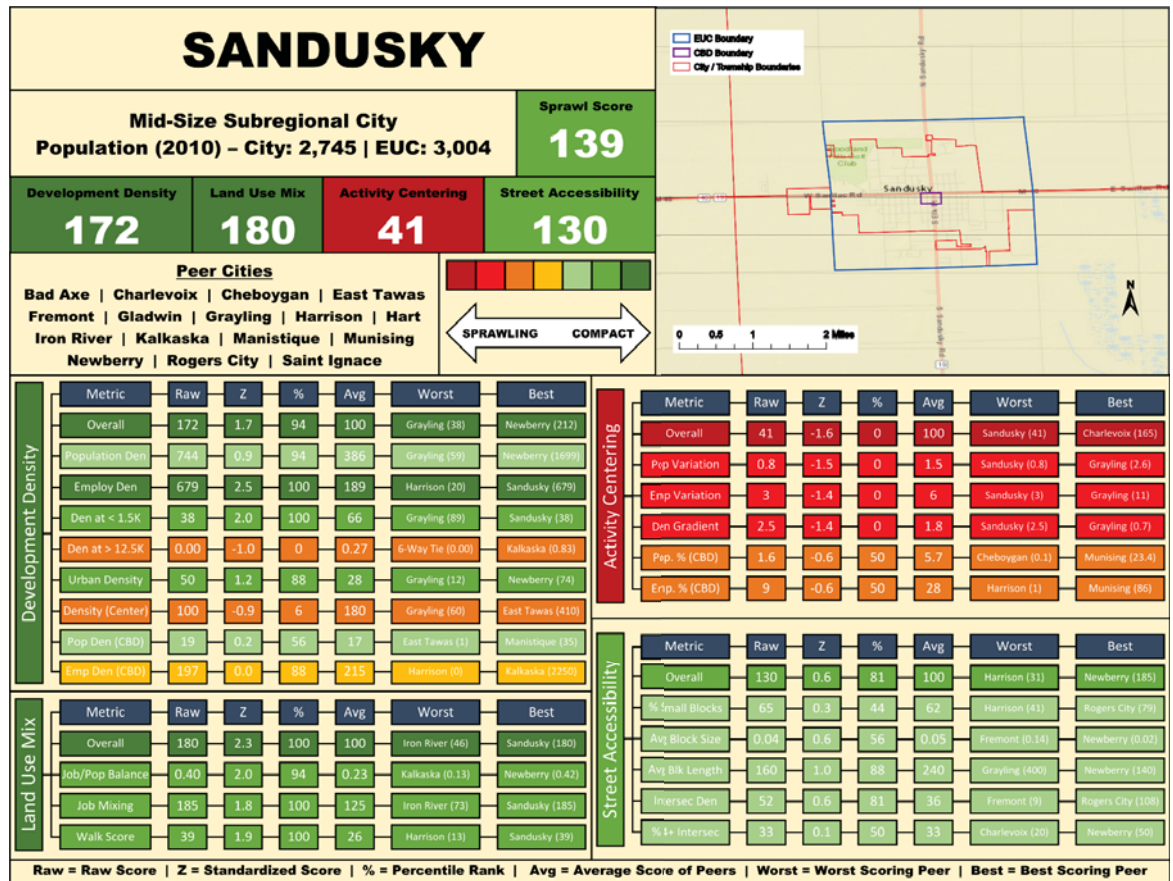


Figure G.39: Sandusky Sprawl Scorecard



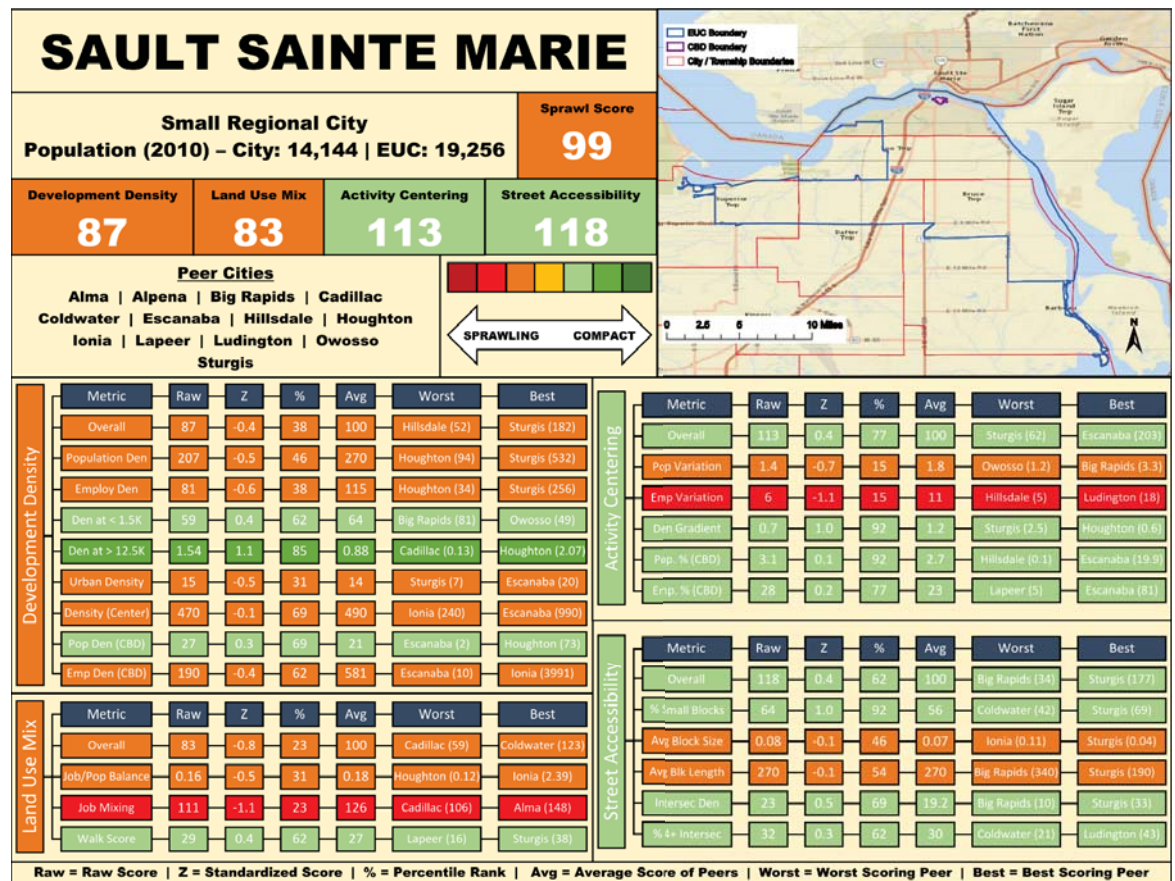


Figure G.40: Sault Sainte Marie Sprawl Scorecard

Figure G-41: Sturgis Sprawl Scorecard

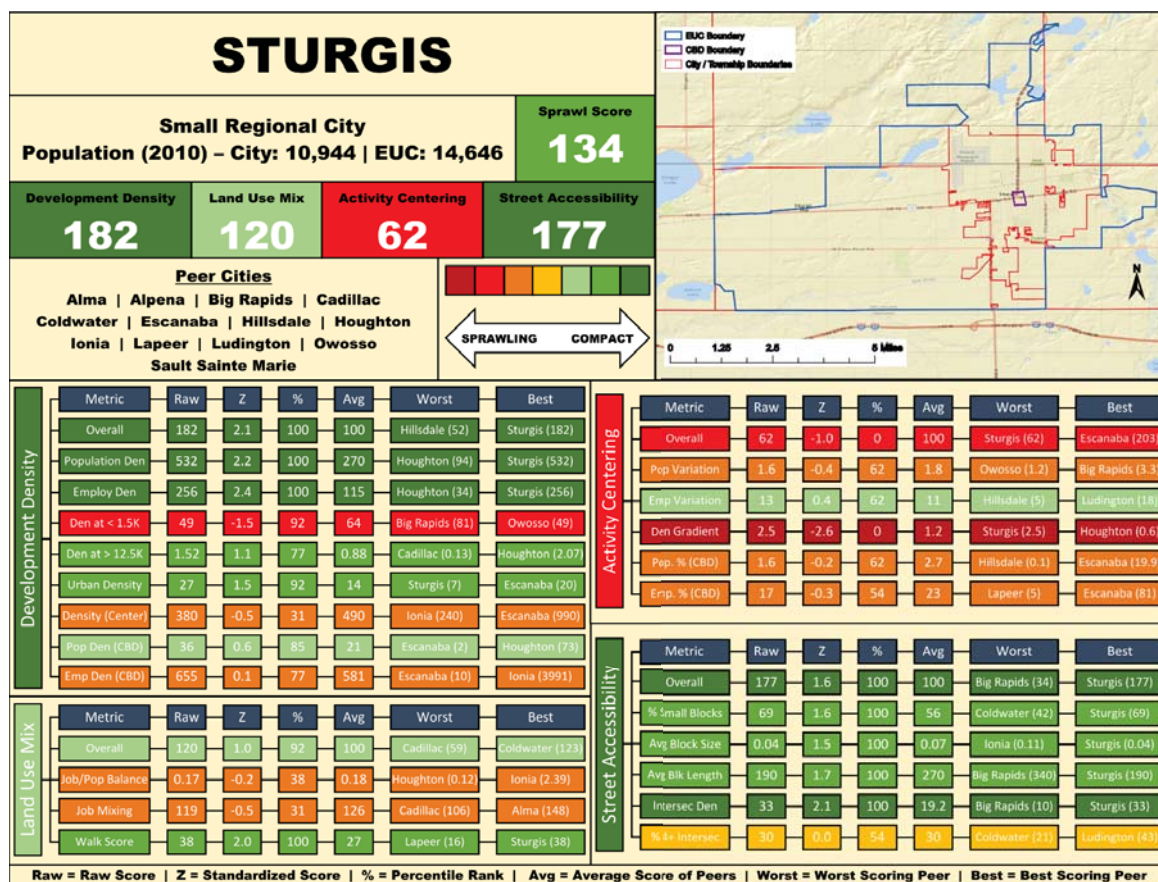
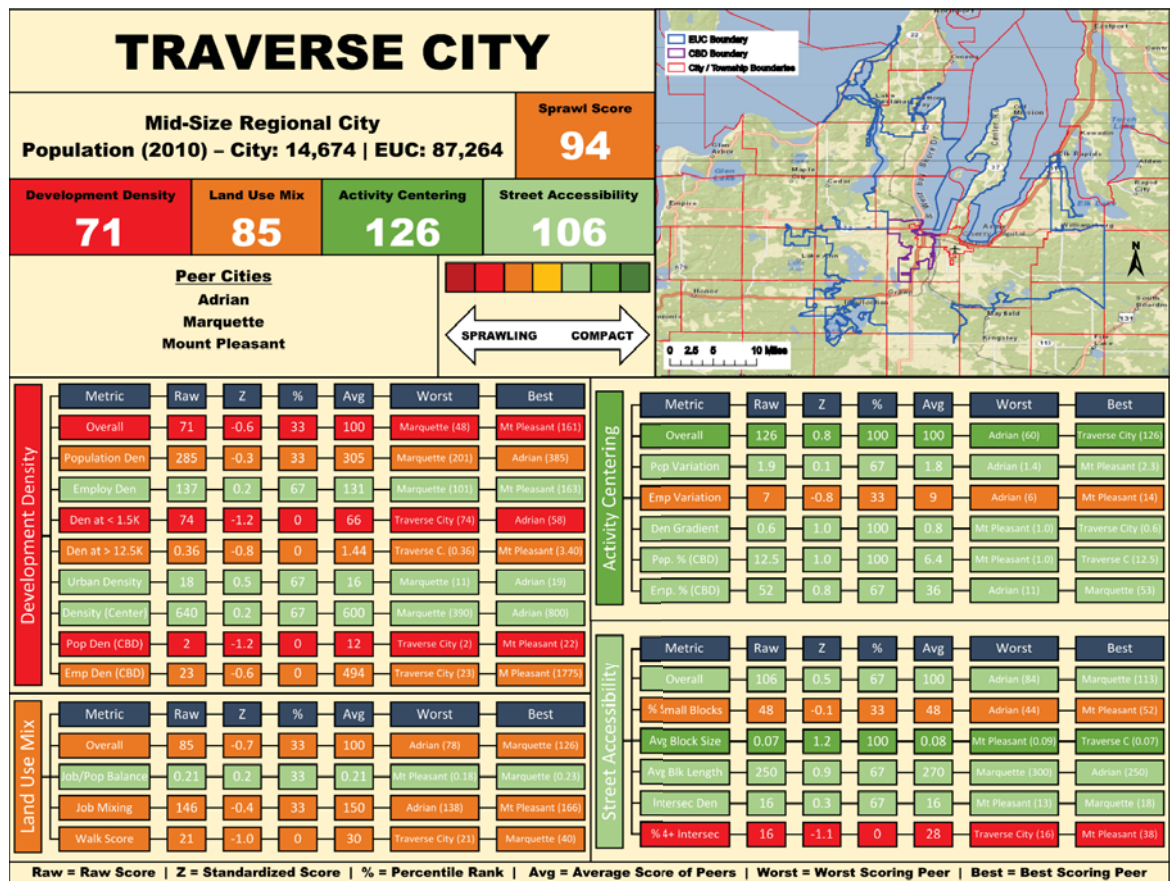


Figure G.42: Traverse City Sprawl Scorecard



Appendix H: Support Files on Supplemental DVD

The following folders and files are included on the DVD in the media pocket on the inside back cover of this dissertation. Each Python script is in a separate folder which includes all files necessary to run the script.

- CBD_Creator
 - CBD_Creator.py
 - CBD_Points_UA_Named_City_Only.shp*
 - EUC_All.shp*
 - tl_2010_26_tabblock10.shp*
- EUC_Creator
 - EUC_All.shp*
 - EUC_Creator.py
 - EUC_Pilot.shp*
 - MI_UAC_2010.shp*
 - ruca2010Michigan.csv
 - tl_2010_26_bg10.shp*
 - tl_2010_26_tabblock10.shp*
- EUC_Metrics
 - EUC_Metrics.py
 - allroads_miv14a.shp*
 - CBD_Points_UA_Named_City_Only.shp*
 - CBD_polygons_all.shp*
 - EUC_All.shp*
 - intersections_MI_EUC.shp*
 - mi_wac_S000_JT01_2010.csv
 - NLCD_table.dbf^
 - tl_2010_26_tabblock10.shp*
- EUC_Sprawl_Score
 - EUC_All.shp
 - EUC_Sprawl_Score.py
- Final_EUC_Shapefile
 - EUC_All.shp
- Sprawl_Scorecards
 - CityX_Scorecard.pdf#
- WS_Locator
 - block_latlon.csv
 - WalkScore_Locator.py

*Shapefiles are actually a collection of between 4 and 7 files

#42 files where CityX is the city name

^NLCD raster file to produce table not included as it is 16GB. The file can be obtained at <http://www.mrlc.gov/>