

# Projekt 1: Wskazanie optymalnej lokalizacji farmy fotowoltaicznej – analizy wielokryterialne (MCE)

Adrian Fabisiewicz (328935)

14 stycznia 2025

## Spis treści

<b>1 Wybór lokalizacji farmy fotowoltaicznej</b>	<b>2</b>
<b>2 Cel i analizowany obszar</b>	<b>2</b>
<b>3 Analizowane kryteria</b>	<b>2</b>
<b>4 Realizacja</b>	<b>3</b>
4.1 Ustalenie środowiska pracy i ścieżek do danych . . . . .	3
4.2 Kryterium 1: odległość od rzek i zbiorników wodnych . . . . .	5
4.2.1 Opis działania . . . . .	5
4.2.2 Kod . . . . .	5
4.2.3 Wynik . . . . .	6
4.3 Kryterium 2: odległość od budynków mieszkalnych . . . . .	7
4.3.1 Opis działania . . . . .	7
4.3.2 Kod . . . . .	7
4.3.3 Wynik . . . . .	8
4.4 Kryterium 3: pokrycie terenu . . . . .	9
4.4.1 Opis działania . . . . .	9
4.4.2 Kod . . . . .	9
4.4.3 Wynik . . . . .	10
4.5 Kryterium 4: dostęp do dróg utwardzonych . . . . .	11
4.5.1 Opis działania . . . . .	11
4.5.2 Kod . . . . .	11
4.5.3 Wynik . . . . .	12
4.6 Kryterium 5: nachylenie stoków . . . . .	13
4.6.1 Opis działania . . . . .	13
4.6.2 Kod . . . . .	13
4.6.3 Wynik . . . . .	13
4.7 Kryterium 6: dostęp do światła słonecznego . . . . .	15
4.7.1 Opis działania . . . . .	15
4.7.2 Kod . . . . .	15
4.7.3 Wynik . . . . .	16
4.8 Kryterium 7: dojazd do istotnych drogowych węzłów komunikacyjnych . . . . .	17
4.8.1 Opis działania . . . . .	17
4.8.2 Kod . . . . .	17
4.8.3 Wynik . . . . .	17
4.9 Ocena przydatności terenu . . . . .	18
4.9.1 Opis działania . . . . .	18
4.9.2 Kod . . . . .	18
4.9.3 Wynik . . . . .	19
4.10 Wybór przydatnych działek . . . . .	20
4.10.1 Opis działania . . . . .	20
4.10.2 Kod . . . . .	21

4.11 Koszt przyłącza do sieci SN . . . . .	22
4.11.1 Opis działania . . . . .	22
4.11.2 Kod . . . . .	23
4.11.3 Wynik . . . . .	25
4.11.4 Ostateczny wybór działki . . . . .	26
<b>5 Test modelu na danych z innego obszaru</b>	<b>26</b>
5.1 Opis obszaru . . . . .	26
5.2 Kryterium 1: odległość od rzek i zbiorników wodnych . . . . .	27
5.3 Kryterium 2: odległość od budynków mieszkalnych . . . . .	28
5.4 Kryterium 3: pokrycie terenu . . . . .	29
5.5 Kryterium 4: dostęp do dróg utwardzonych . . . . .	30
5.6 Kryterium 5: nachylenie stoków . . . . .	31
5.7 Kryterium 6: dostęp do światła słonecznego . . . . .	32
5.8 Kryterium 7: dojazd do istotnych drogowych węzłów komunikacyjnych . . . . .	33
5.9 Ocena przydatności terenu . . . . .	34
5.10 Wybór przydatnych działek . . . . .	36
5.11 Przyłącze do sieci SN . . . . .	37

## 1 Wybór lokalizacji farmy fotowoltaicznej

co należy wziąć pod uwagę, wybierając lokalizację farmy fotowoltaicznej (rozważania teoretyczne, akty prawne wraz z cytowaniami źródeł / bibliografią)

## 2 Cel i analizowany obszar

Celem projektu było wskazanie optymalnej lokalizacji nowej farmy fotowoltaicznej dla obszaru gminy Świeradów-Zdrój (powiat lubański, województwo dolnośląskie).

## 3 Analizowane kryteria

Lp	Kryterium	Parametry	Źródło danych do kryterium
1	odległość od rzek i zbiorników wodnych	jak najbliżej; nieprzekraczalna 100-metrowa strefa ochronna	BDOT10k(SWRS, PTWP)
2	odległość od budynków mieszkalnych	jak najdalej, powyżej 150m	BDOT10k(BUBD)
3	pokrycie terenu	powyżej 15m od lasu, optymalnie powyżej 100m od lasu	BDOT10k(PTLZ)
4	dostęp do dróg utwardzonych	jak największe zagęszczenie	BDOT10k(SKDR)
5	nachylenie stoków	jak najbardziej płasko	NMT
6	dostęp światła słonecznego	optymalnie: stoki południowe (SW-SE)	NMT
7	dobry dojazd od istotnych drogowych węzłów komunikacyjnych	jak najkrótszy czas dojazdu	BDOT10k(SKDR)
Łączenie kryteriów			
8	ocena przydatności terenu (próg przydatności)	80% / 90% max. przydatności	
9	przydatne działki / grupy działek	min 60% działki na terenie przydatnym	EGIB
10	powierzchnia i min. szerokość obszaru	2ha / 50m	
11	koszt przyłącza do sieci SN (mapy kosztów)	jak najniższy	BDOT10k (wszystkie warstwy PT)

Tabela 1: Tabela z kryteriami lokalizacji

## 4 Realizacja

### 4.1 Ustalenie środowiska pracy i ścieżek do danych

Na początku pracy z pythonem należało zainportować odpowiednie moduły z biblioteki arcpy oraz zadbać o to, żeby jako interpreter języka Python została wybrana wersja Pythona instalowana razem z oprogramowaniem ArcGIS.

```

1 import arcpy.analysis
2 import arcpy.management
3 import arcpy.sa

```

Przed rozpoczęciem pracy z danymi należało ustalić odpowiednie parametry środowiska. Ustalono odpowiedni folder odczytu i zapisu danych na geobazę projektu, w której znajdowały się odpowiednie dane. Ustawiono układ współrzędnych na EPSG:2180 oraz rozmiar komórki na 5, zgodny z rozmiarem pobranego przez nas rastra NMT. Zakres oraz maskę projektu dostosowano do powierzchni warstwy, zawierającej ???-metrowy bufor wokół terenu gminy. Ustalono również nadpisywanie warstw w przypadku, jeżeli warstwa o tej samej nazwie już by istniała w folderze.

```

1 geobaza = r"C:\Users\adria\Desktop\STUDIA_FOLDERY\analizy\MyProject12\
2   MyProject12.gdb"
3 arcpy.env.workspace = "in_memory"
4 arcpy.env.outputCoordinateSystem = arcpy.SpatialReference(" 
5   ETRS_1989_Poland_CS92")
6 arcpy.env.extent = f"{geobaza}\gmina_buffer"
7 arcpy.env.mask = f"{geobaza}\gmina_buffer"
8 arcpy.env.cellSize = 5

```

```
7 arcpy.env.overwriteOutput = True
```

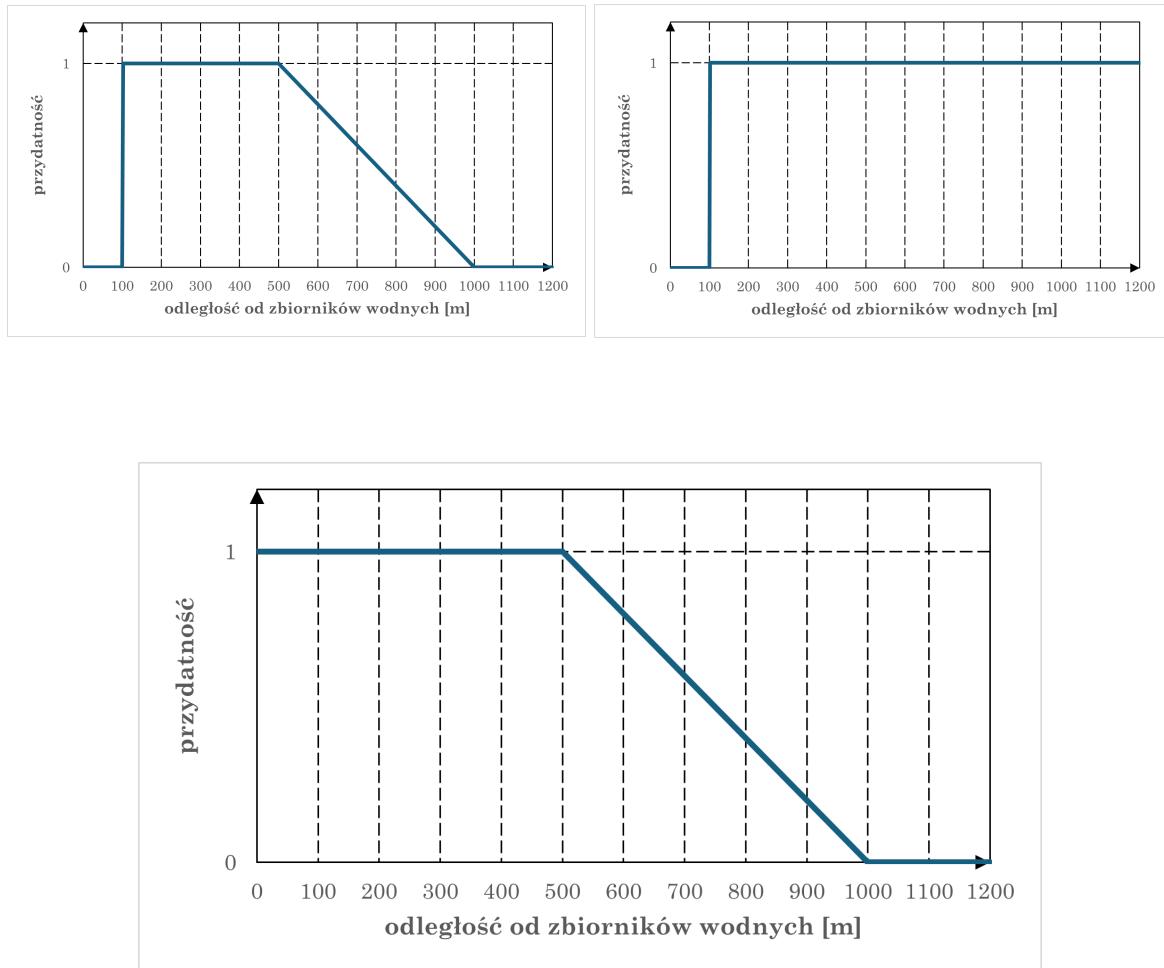
Następnie zapisano do zmiennych wszystkie niezbędne warstwy do przeprowadzenia analizy oraz, jeżeli było to konieczne, dokonano podstawowych operacji, w celułączenia danych dla dwóch powiatów.

```
1 swrs_0210_buffer = arcpy.analysis.Buffer(f'{geobaza}\\\SWRS_L_0210', f'{geobaza}\\\SWRS_L_0210_buffer', '1 Centimeter')
2 swrs_0212_buffer = arcpy.analysis.Buffer(f'{geobaza}\\\SWRS_L_0212', f'{geobaza}\\\SWRS_L_0212_buffer', '1 Centimeter')
3 water = arcpy.management.Merge([swrs_0210_buffer, swrs_0212_buffer, f'{geobaza}\\\PTWP_A_0210', f'{geobaza}\\\PTWP_A_0212'], 'water')
4 budynki = arcpy.management.Merge([f'{geobaza}\\\BUBD_A_0210', f'{geobaza}\\\BUBD_A_0212'], 'budynki')
5 ptlz = arcpy.management.Merge([f'{geobaza}\\\PTLZ_A_0210', f'{geobaza}\\\PTLZ_A_0212'], 'ptlz')
6 nmt = f'{geobaza}\\\nmt'
7 drogi = arcpy.management.Merge([f'{geobaza}\\\SKDR_L_0210', f'{geobaza}\\\SKDR_L_0212'], 'drogi')
8 wezly = f'{geobaza}\\\wezly_raster'
9 dzialki = f'{geobaza}\\\dzialki'
10 pt_merged = f'{geobaza}\\\PT_merged'
11 linie_elektroenergetyczne = arcpy.management.Merge([f'{geobaza}\\\SULN_L_0210', f'{geobaza}\\\SULN_L_0212'], 'linie_elektroenergetyczne')
```

## 4.2 Kryterium 1: odległość od rzek i zbiorników wodnych

### 4.2.1 Opis działania

\* dodaj tu coś \*



Reklasyfikacja dla kryterium 1.

### 4.2.2 Kod

```
1 out_distance_accumulation_raster = arcpy.sa.DistanceAccumulation(in_source_data  
=water)  
2 woda_rosnaca = arcpy.sa.FuzzyMembership(out_distance_accumulation_raster,  
fuzzy_function="LINEAR 100 102")  
3 woda_malejaca = arcpy.sa.FuzzyMembership(out_distance_accumulation_raster,  
fuzzy_function="LINEAR 1000 500")  
4 woda_mapa = arcpy.sa.FuzzyOverlay([woda_rosnaca, woda_malejaca], 'AND')  
5 woda_mapa.save(f'{geobaza}\kryterium_1')
```

#### 4.2.3 Wynik



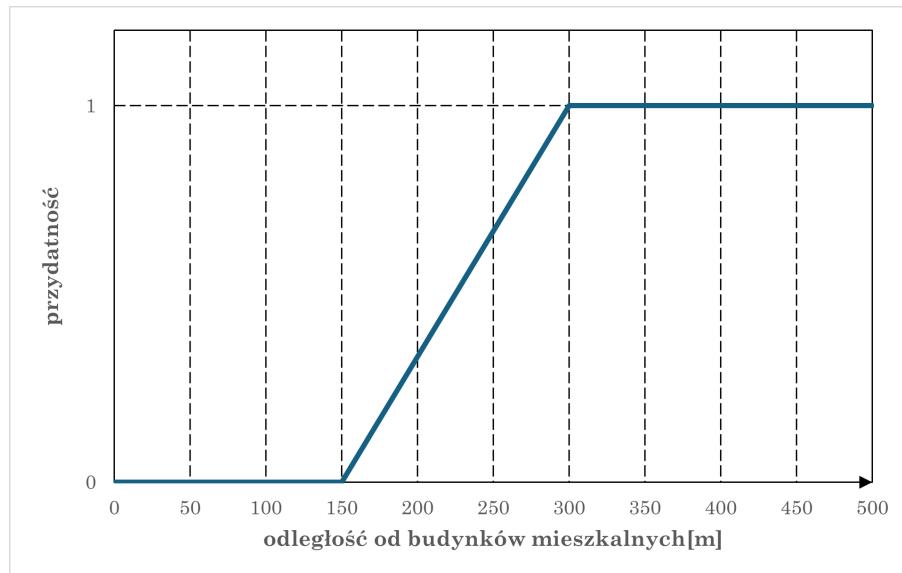
Mapa przydatności dla kryterium 1.



Mapa przydatności dla kryterium 1. zawierająca rzeki oraz zbiorniki wodne

## 4.3 Kryterium 2: odległość od budynków mieszkalnych

### 4.3.1 Opis działania

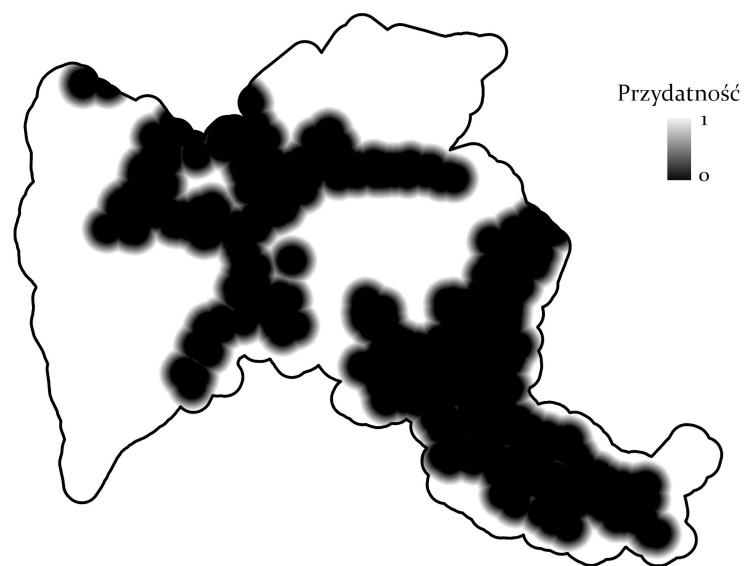


Reklasyfikacja dla kryterium 2.

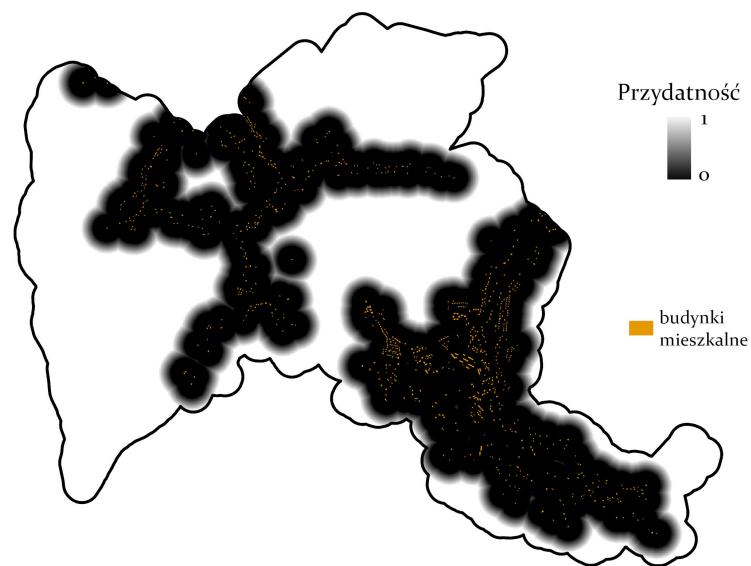
### 4.3.2 Kod

```
1 query = "FOBUD = 'budynki mieszkalne'"
2 arcpy.analysis.Select(budynki, 'budynki_mieszkalne', query)
3 out_distance_accumulation_buildings = arcpy.sa.DistanceAccumulation(
4     in_source_data='budynki_mieszkalne')
5 budynki_mieszkalne = arcpy.sa.FuzzyMembership(
    out_distance_accumulation_buildings, fuzzy_function="LINEAR 150 300")
6 budynki_mieszkalne.save(f'{geobaza}\kryterium_2')
```

#### 4.3.3 Wynik



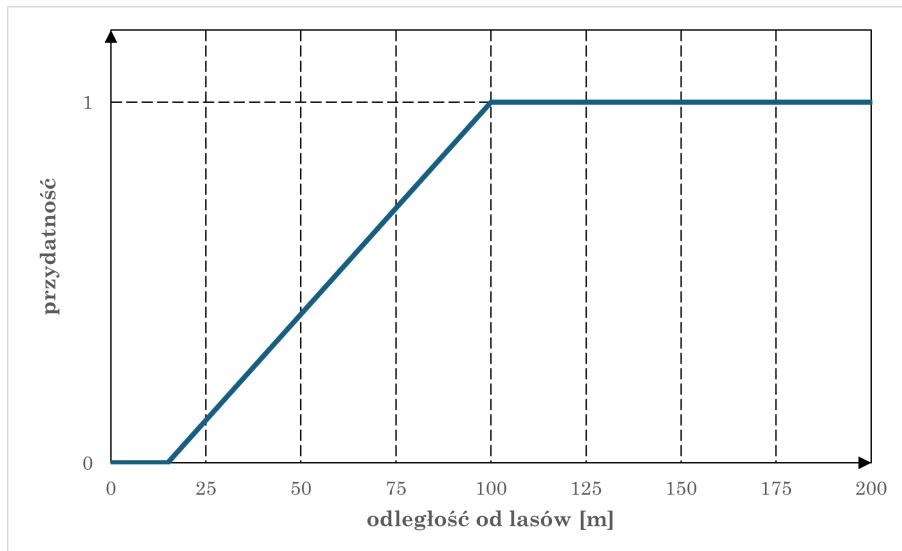
Mapa przydatności dla kryterium 2.



Mapa przydatności dla kryterium 2. zawierająca budynki mieszkalne

## 4.4 Kryterium 3: pokrycie terenu

### 4.4.1 Opis działania

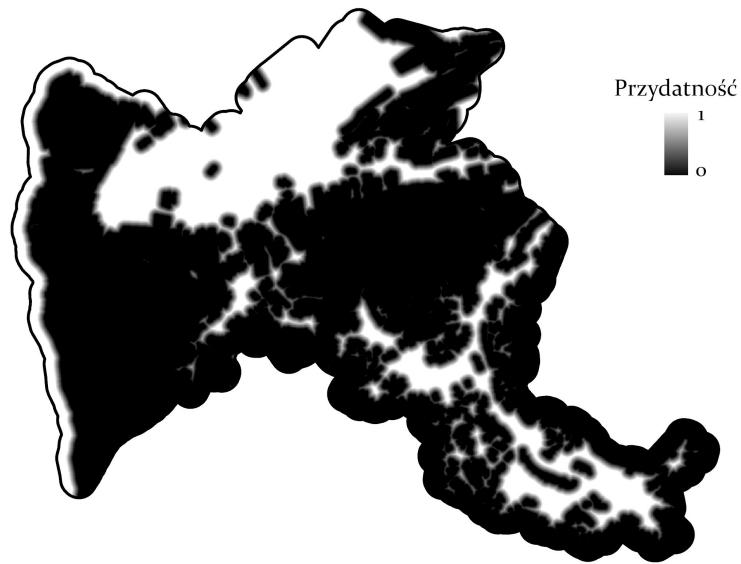


Reklasyfikacja dla kryterium 3.

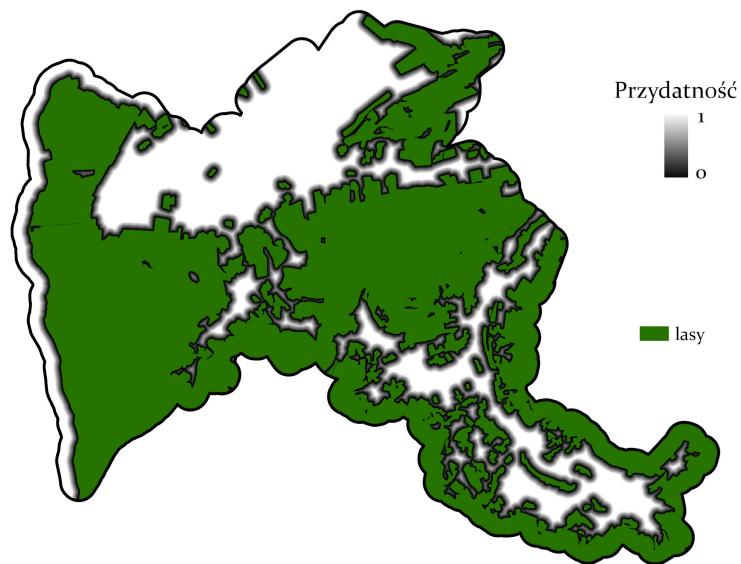
### 4.4.2 Kod

```
1 query = "RODZAJ = 'las'"
2 arcpy.analysis.Select(ptlz, 'ptlz_las', query)
3 out_distance_accumulation_ptlz = arcpy.sa.DistanceAccumulation(in_source_data="ptlz_las")
4 lasy_fuzzy = arcpy.sa.FuzzyMembership(out_distance_accumulation_ptlz,
5     fuzzy_function="LINEAR 15 100")
lasy_fuzzy.save(f'{geobaza}\kryterium_3')
```

#### 4.4.3 Wynik



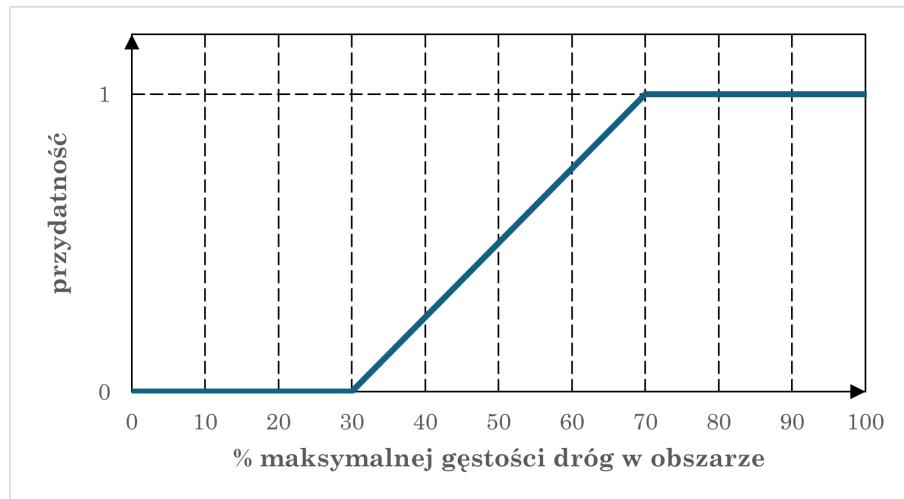
Mapa przydatności dla kryterium 3.



Mapa przydatności dla kryterium 3. zawierająca lasy

## 4.5 Kryterium 4: dostęp do dróg utwardzonych

### 4.5.1 Opis działania

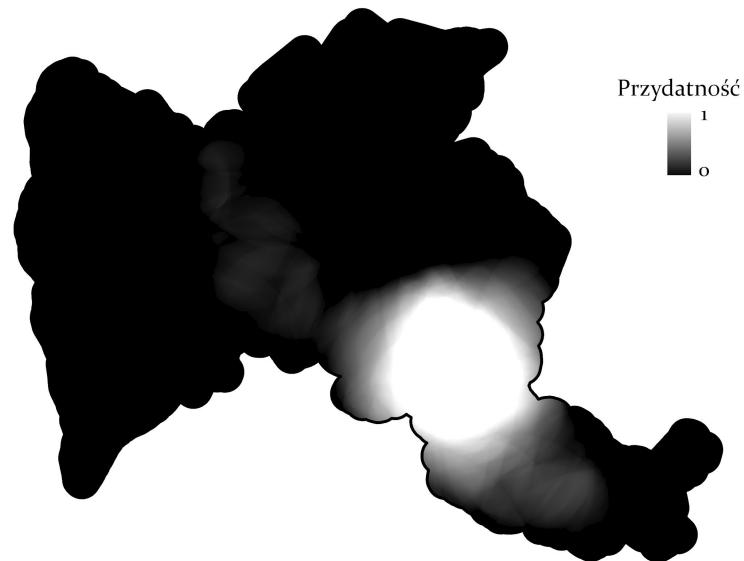


Reklasyfikacja dla kryterium 4.

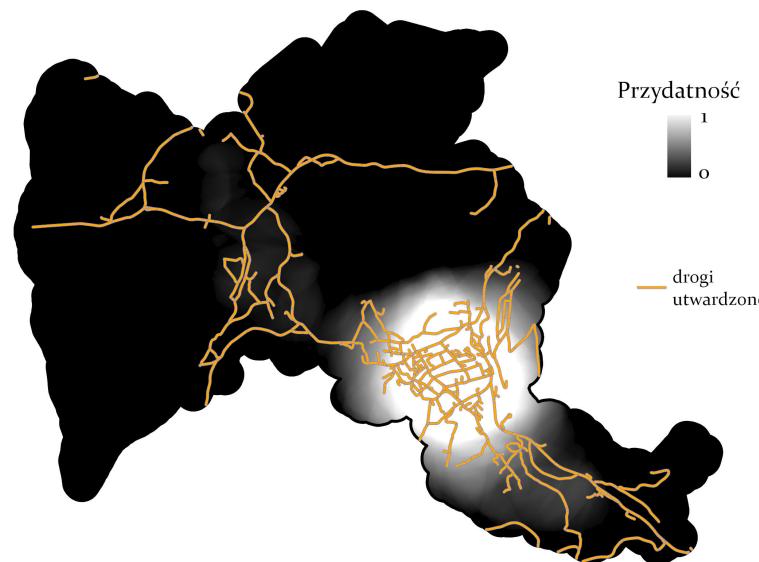
### 4.5.2 Kod

```
1 query = "MATE\NAWIE IN ('beton', 'bruk', 'kostka kamienna', 'kostka
2     prefabrykowana', 'masa bitumiczna', 'plyty betonowe')"
drogi_utwardzone = arcpy.analysis.Select(drogi, 'drogi_utwardzone', query)
3
4 density = arcpy.sa.LineDensity(
5     in_polyline_features=drogi_utwardzone,
6     population_field=None,
7     cell_size=5,
8     search_radius=1000,
9     area_unit_scale_factor="SQUARE_KILOMETERS",
10 )
11 density.save(f'{geobaza}\drogi_density')
12
13 arcpy.management.CalculateStatistics(density)
14 max_value = density.maximum
15
16 kryterium_4 = arcpy.sa.RescaleByFunction(
17     in_raster=density,
18     transformation_function=f"LINEAR {0.3 * max_value} {0.7 * max_value}",
19     from_scale=0,
20     to_scale=1
21 )
22 kryterium_4.save(f'{geobaza}\kryterium_4')
```

#### 4.5.3 Wynik



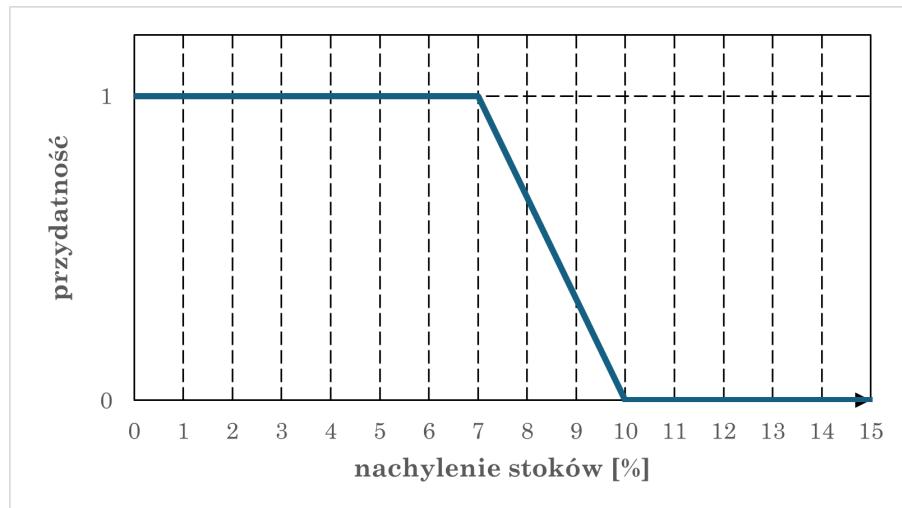
Mapa przydatności dla kryterium 4.



Mapa przydatności dla kryterium 4. zawierająca drogi utwardzone

## 4.6 Kryterium 5: nachylenie stoków

### 4.6.1 Opis działania



Reklasyfikacja dla kryterium 5.

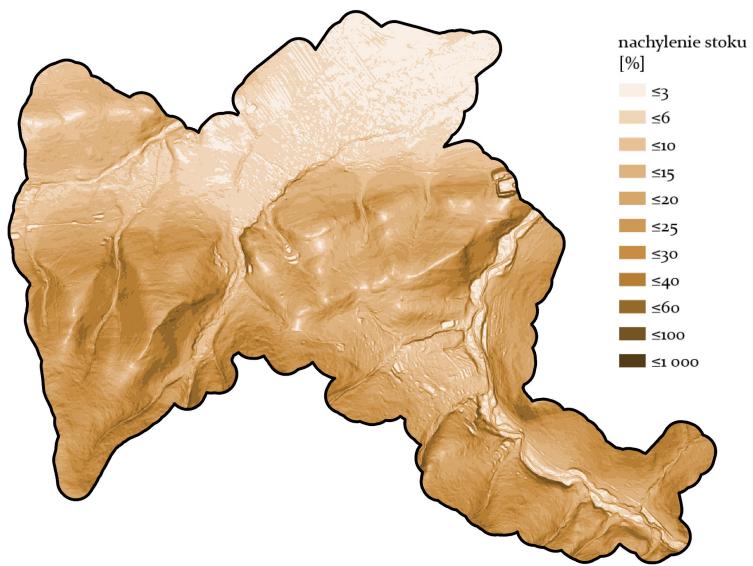
### 4.6.2 Kod

```
1 arcpy.ddd.Slope(nmt, "slope", "PERCENT_RISE", 1)
2 slope_fuzzy = arcpy.sa.FuzzyMembership("slope", fuzzy_function="LINEAR 10 7")
3 slope_fuzzy.save(f'{geobaza}\\kryterium_5')
```

### 4.6.3 Wynik



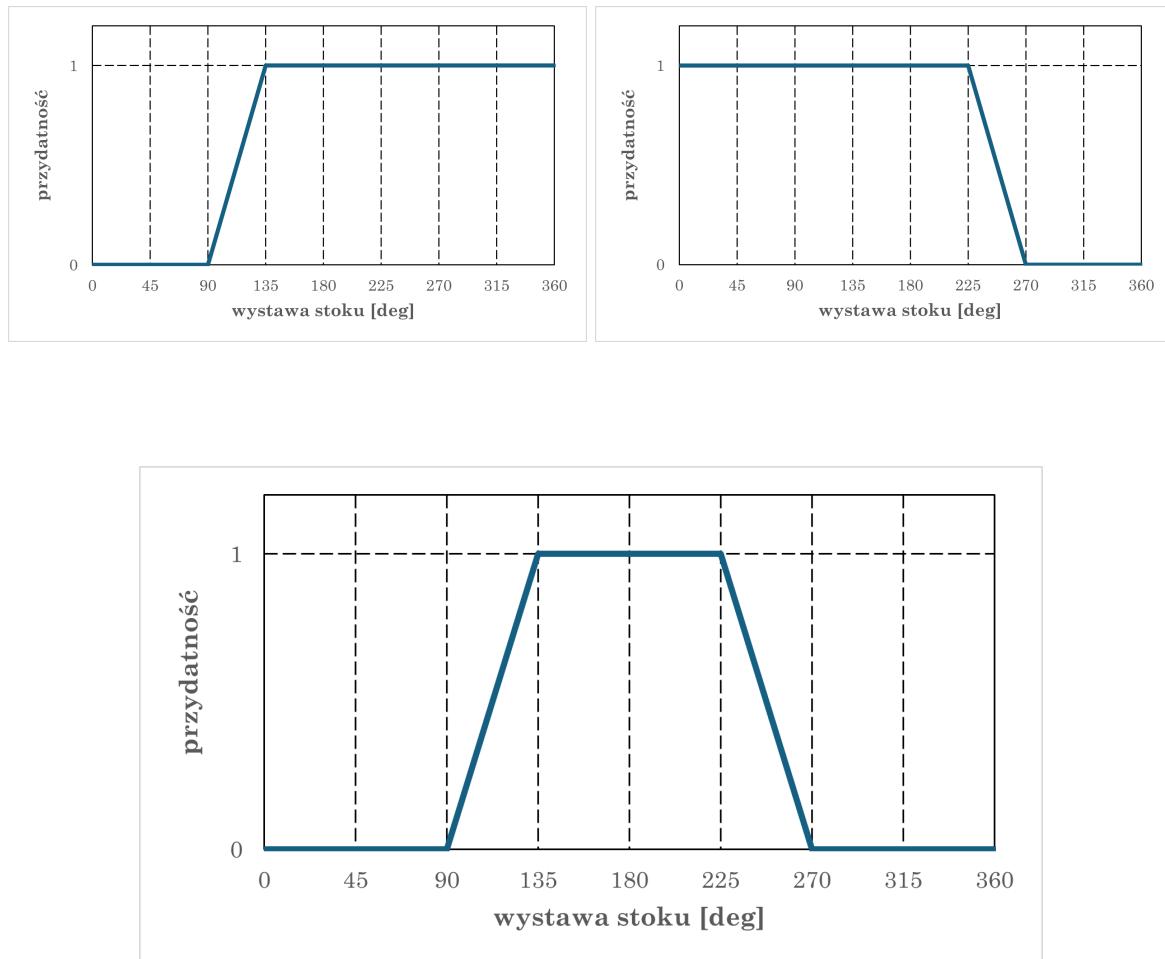
Mapa przydatności dla kryterium 5.



Mapa nachyleń stoków wykorzystana podczas sprawdzania kryterium 5.

## 4.7 Kryterium 6: dostęp do światła słonecznego

### 4.7.1 Opis działania

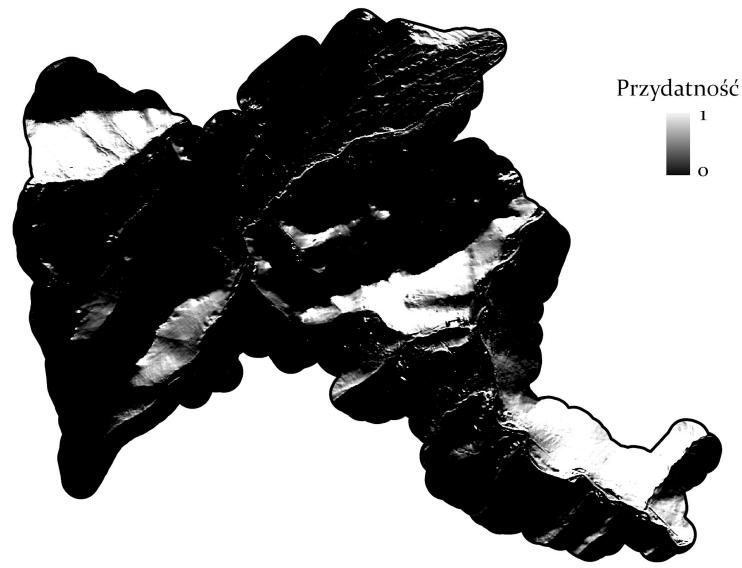


Reklasyfikacja dla kryterium 6.

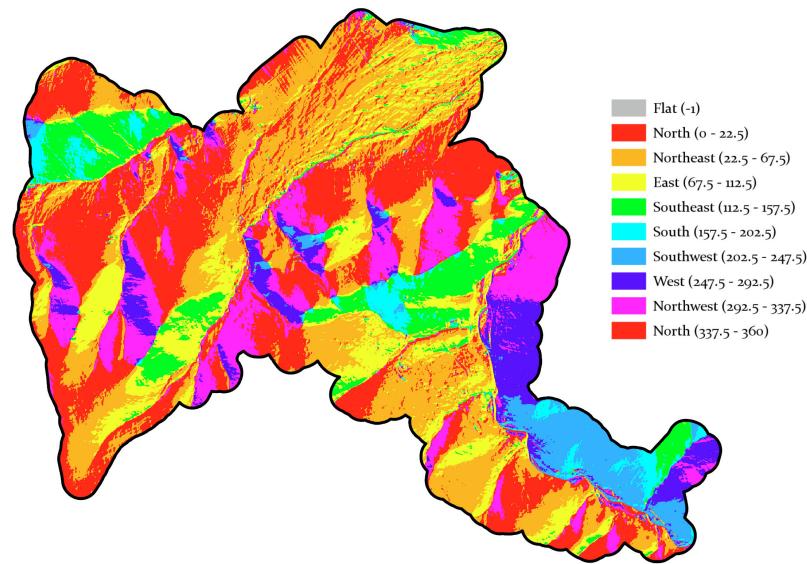
### 4.7.2 Kod

```
1 aspect = arcpy.dsd.Aspect(nmt)
2 aspect_fuzzy = arcpy.sa.FuzzyMembership(aspect, fuzzy_function="LINEAR 90 135")
3 aspect_fuzzy_1 = arcpy.sa.FuzzyMembership(aspect, fuzzy_function="LINEAR 270
   225")
4 aspect_overlay = arcpy.sa.FuzzyOverlay([aspect_fuzzy, aspect_fuzzy_1], 'AND')
5 aspect_overlay.save(f'{geobaza}\\kryterium_6')
```

#### 4.7.3 Wynik



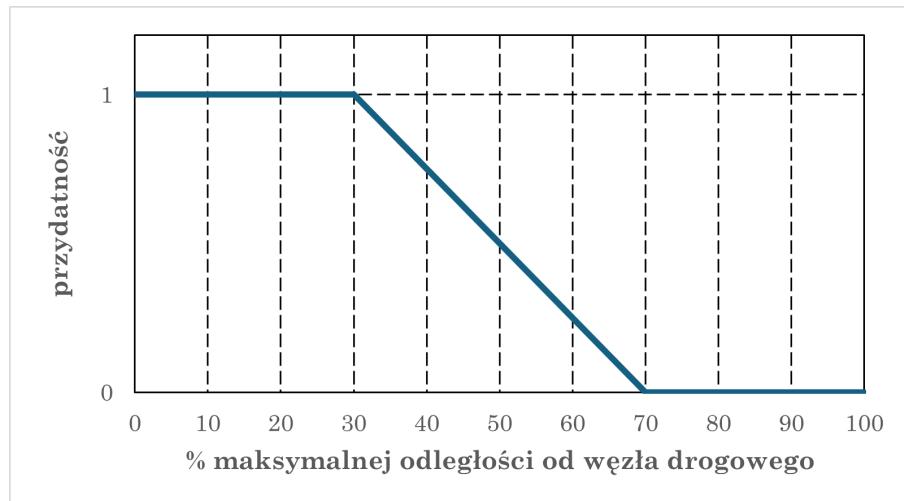
Mapa przydatności dla kryterium 6.



Mapa przydatności dla kryterium 6. zawierająca stopień wystawy słonecznej

## 4.8 Kryterium 7: dojazd do istotnych drogowych węzłów komunikacyjnych

### 4.8.1 Opis działania

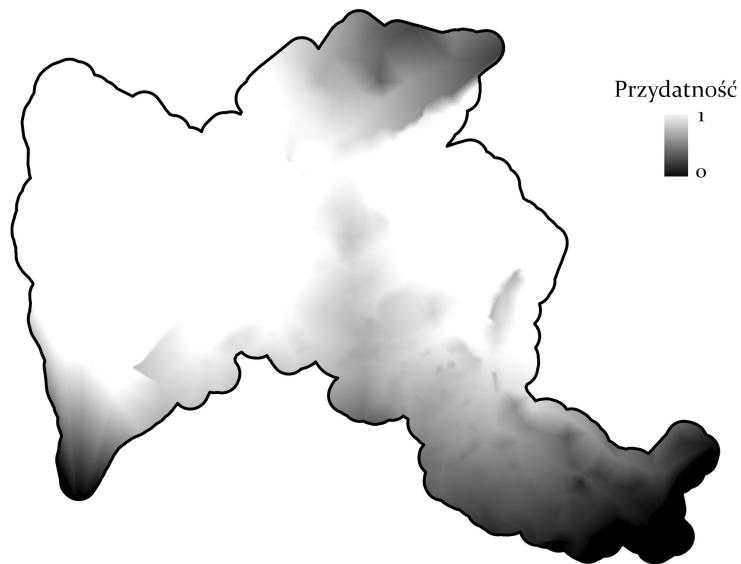


Reklasyfikacja dla kryterium 7.

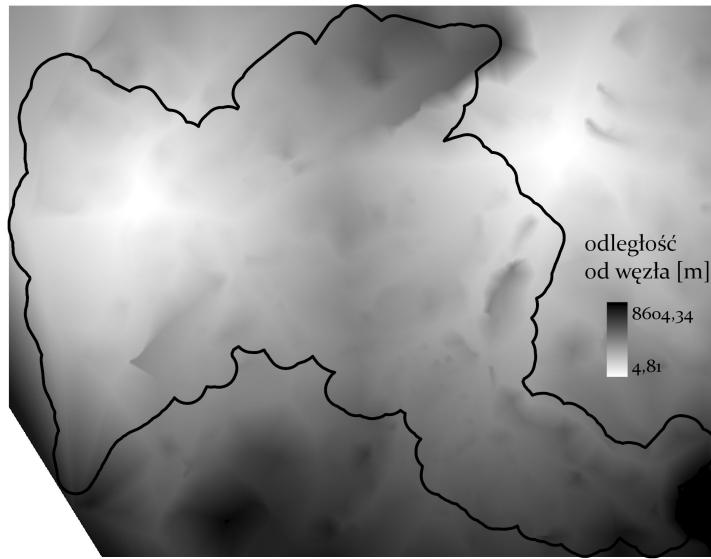
### 4.8.2 Kod

```
1 wezly_max = float(arcpy.management.GetRasterProperties(wezly, "MAXIMUM")[0].  
    replace(',', '.'))  
2 wezly_fuzzy = arcpy.sa.FuzzyMembership(wezly, fuzzy_function=f"LINEAR {0.7 *  
    wezly_max} {0.3 * wezly_max}")  
3 wezly_fuzzy.save(f'{geobaza}\kryterium_7')
```

### 4.8.3 Wynik



Mapa przydatności dla kryterium 7.



Mapa odległości od węzłów

## 4.9 Ocena przydatności terenu

### 4.9.1 Opis działania

Powyższy kod najpierw tworzy tabelę zawierającą wagi dla każdego z kryteriów, zmienne w zależności od wariantu. Następnie tworzy sumę ważoną dla wszystkich z kryteriów.

Później brane są pod uwagę kryteria ostre, tj. 100-metrowa strefa ochronna od wód, 150-metrowa odległość od budynków mieszkalnych oraz 15-metrowa odległość od lasów. Zostaje utworzony raster, który w pikselu, dla którego przydatność dla któregokolwiek z kryterium wyniosła 0, przyjmuje również przydatność równą 0. Dzięki temu miejsca wyeliminowane przez którekolwiek z kryteriów ostrzych nie będą brane pod uwagę przy wyborze lokalizacji farmy.

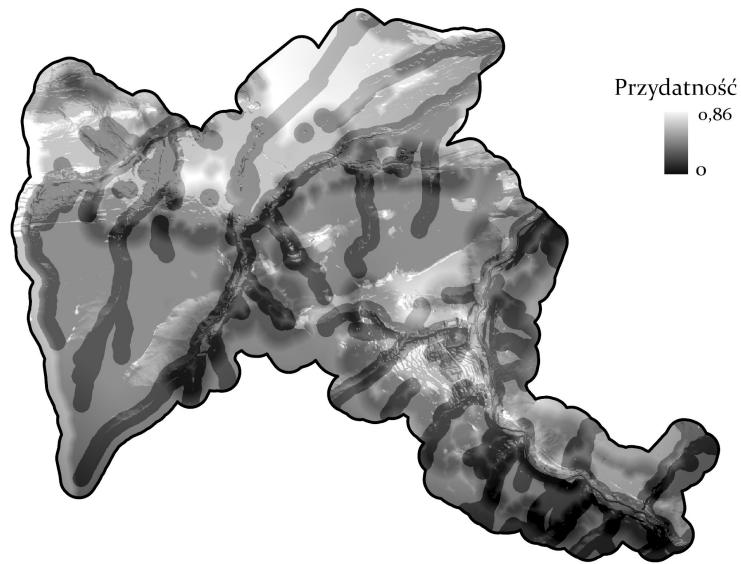
Następnie zostaje utworzona suma warstwy kryteriów ostrzych i rozmytych. Piksele z przydatnością powyżej progu przydatności zostały zremapowane na 1, pozostałe na 0.

### 4.9.2 Kod

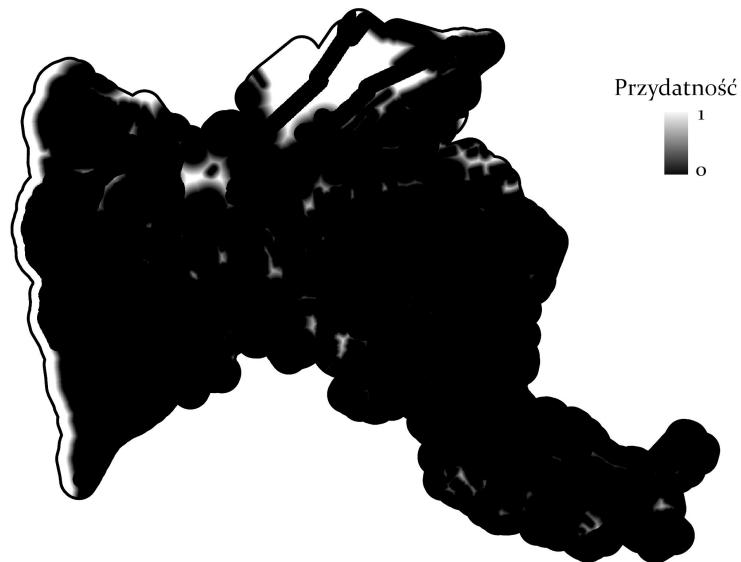
```

1 tabela_kryteriow = arcpy.sa.WSTable([[f'{geobaza}\kryterium_1', "VALUE",
2 waga_woda], [f'{geobaza}\kryterium_2', "VALUE", waga_budynki], [f'{geobaza}
3 }\kryterium_3', "VALUE", waga_lasyl], [f'{geobaza}\kryterium_4', "VALUE",
4 waga_drogil], [f'{geobaza}\kryterium_5', "VALUE", waga_wysokosc], [f'{geobaza}\kryterium_6', "VALUE", waga_aspect], [f'{geobaza}\kryterium_7', "
5 VALUE", waga_wezly]])
6 weighted_sum = arcpy.sa.WeightedSum(tabela_kryteriow)
7 weighted_sum.save(f'{geobaza}\{wariant}_suma_rozmyte')
8
9 kryteria_ostre = arcpy.sa.FuzzyOverlay([woda_rosnaca, f'{geobaza}\kryterium_2"
10 , f'{geobaza}\kryterium_3"], 'AND')
11 kryteria_ostre.save(f'{geobaza}\kryteria_ostre')
12
13 suma = arcpy.sa.FuzzyOverlay([kryteria_ostre, weighted_sum], 'AND')
14 suma.save(f'{geobaza}\{wariant}_wynik')
15
16 wynik_reclassified = arcpy.sa.Reclassify(suma, "VALUE", arcpy.sa.RemapRange
17 ([[0, prog_przydatnosci, 0], [prog_przydatnosci, 1, 1]]))
18 wynik_reclassified.save(f'{geobaza}\{wariant}_wynik_reclassified')
```

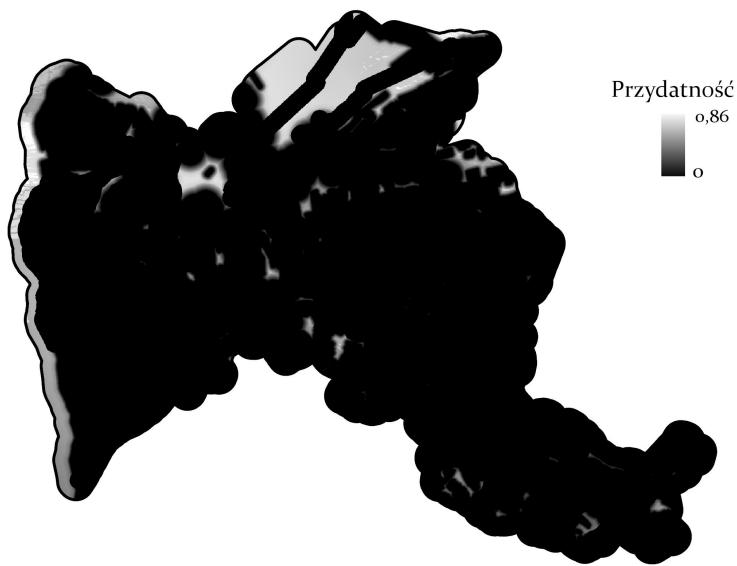
#### 4.9.3 Wynik



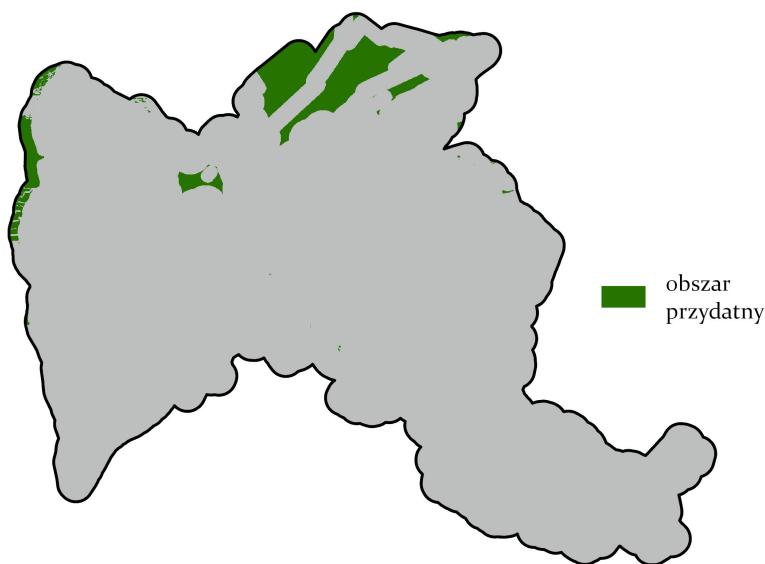
Suma kryteriów rozmytych



Suma kryteriów ostrych



Wynik łączenia kryteriów ostrych i rozmytych



Wynik łączenia kryteriów ostrych i rozmytych po reklasyfikacji

## 4.10 Wybór przydatnych działek

### 4.10.1 Opis działania

Kod najpierw zamienia uzyskany wcześniej raster na warstwę poligonową. Następnie zostają wybrane tylko te poligony, które stanowią teren przydatny.

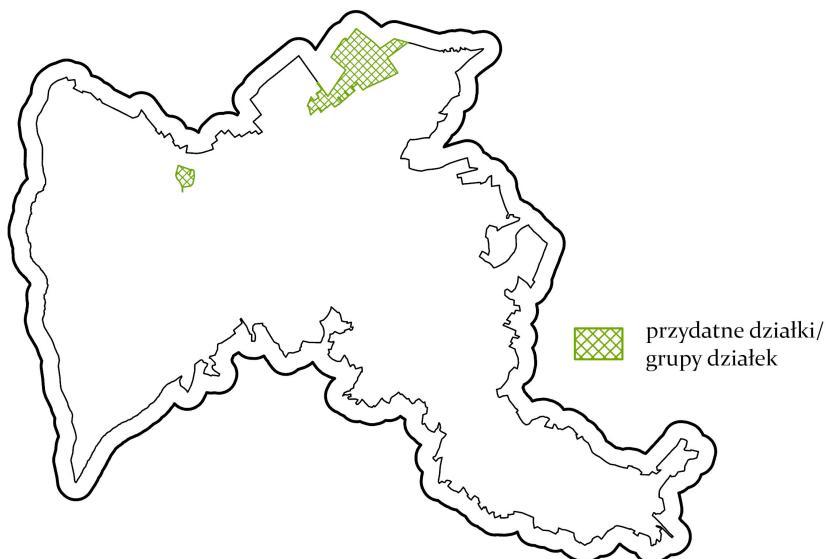
Później warstwa z działkami zostaje przecięta powstałymi poligonami. Dzięki temu działka zostaje podzielona na części przydatne i nieprzydatne. Kolejno, aby uzyskać procentową przydatność każdej z działek, pętla przechodzi przez warstwę z działkami przeciętymi oraz pierwotną warstwę z działkami. Jeżeli pętla natrafi na część przydatną działki, zostaje znaleziona odpowiednia działka w warstwie pierwotnej, i pole części przydatnej działki zostaje dodane do pola `pole_przydatne`.

Dzięki temu możemy obliczyć przydatność działki - stosunek powierzchni przydatnej do powierzchni cał-

kowitej. Wybieramy jedynie te działki, których przydatność jest wyższa od wybranego progu przydatności oraz takie, które mają powierzchnię co najmniej 2ha.

#### 4.10.2 Kod

```
1  polygon_przydatnosci = "polygon_przydatnosci"
2  arcpy.conversion.RasterToPolygon(f'{geobaza}\{wariant}_wynik_reclassified',
3      polygon_przydatnosci, "NO_SIMPLIFY", "VALUE")
4
5  polygon_przydatnosci_layer = "polygon_przydatnosci_layer"
6  arcpy.management.MakeFeatureLayer(polygon_przydatnosci,
7      polygon_przydatnosci_layer)
8  arcpy.management.SelectLayerByAttribute(polygon_przydatnosci_layer, "NEW_SELECTION", "gridcode = 1")
9  arcpy.management.CopyFeatures(polygon_przydatnosci_layer, f'{geobaza}\{wariant}_poligon_przydatnosci')
10
11 dzialki_przeciete = "dzialki_przeciete"
12 arcpy.analysis.Intersect([dzialki, polygon_przydatnosci], dzialki_przeciete, "ALL")
13 arcpy.management.CopyFeatures(dzialki_przeciete, f'{geobaza}\{wariant}_dzialki_przeciete_pole')
14
15 arcpy.management.AddField(dzialki_przeciete, "pole", "DOUBLE")
16 arcpy.management.CalculateField(dzialki_przeciete, "pole", "!shape.area@SQUAREMETERS!", "PYTHON3")
17
18 arcpy.management.AddField(dzialki, "pole", "DOUBLE")
19 arcpy.management.CalculateField(dzialki, "pole", "!shape.area@SQUAREMETERS!", "PYTHON3")
20
21 arcpy.management.AddField(dzialki, "pole_przydatne", "DOUBLE")
22 arcpy.management.CalculateField(dzialki, "pole_przydatne", 0, "PYTHON3")
23
24 with arcpy.da.UpdateCursor(dzialki_przeciete, ["ID_DZIAŁKI", "pole", "gridcode"]) as cursor:
25     for row in cursor:
26         if row[2] == 1:
27             with arcpy.da.UpdateCursor(dzialki, ["ID_DZIAŁKI", "pole", "pole_przydatne"]) as cursor2:
28                 for row2 in cursor2:
29                     if row[0] == row2[0]:
30                         row2[2] += row[1]
31                         cursor2.updateRow(row2)
32
33 arcpy.management.AddField(dzialki, "przydatnosc", "DOUBLE")
34 arcpy.management.CalculateField(dzialki, "przydatnosc", "(!pole_przydatne! / !pole!) * 100", "PYTHON3")
35
36 dzialki_layer = "dzialki_layer"
37 arcpy.management.MakeFeatureLayer(dzialki, dzialki_layer)
38 arcpy.management.SelectLayerByAttribute(dzialki_layer, "NEW_SELECTION", f"przydatnosc > {prog_przydatnosci} AND shape_area > 20000 ")
39 arcpy.management.CopyFeatures(dzialki_layer, f'{geobaza}\{wariant}_dzialki_przydatne_powyzej_{prog_przydatnosci}')
```



Mapa przedstawiająca przydatne działki



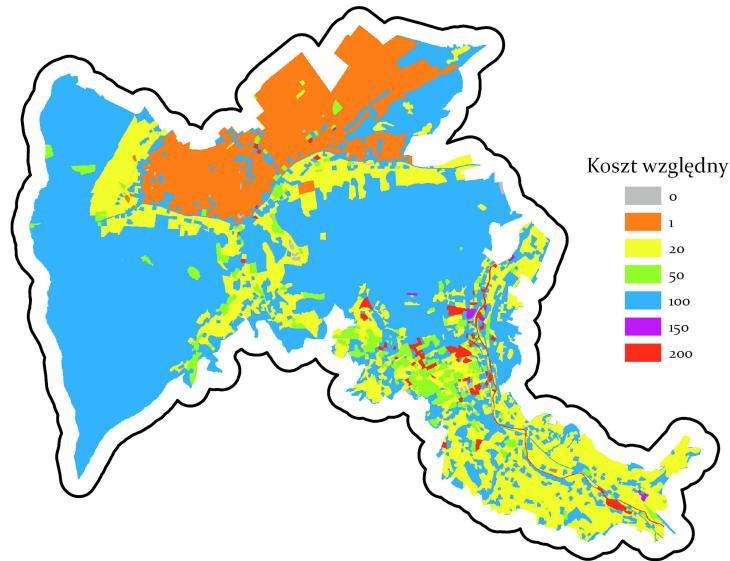
Mapa przedstawiająca przydatne działki na ortofotomapie

## 4.11 Koszt przyłącza do sieci SN

### 4.11.1 Opis działania

Przed rozpoczęciem analiz należało stworzyć tzw. mapę kosztów względnych(jednostkowych), która przedstawia faktyczną lub umowną wartość kosztu zbudowania przyłącza przez dany obszar. Dla mapy kosztów względnych w postaci rastrowej, będzie to wzajemny koszt budowy przyłącza na obszarze o powierzchni odpowiadającej rozmiarowi piksela. W ćwiczeniu przyjęto, że koszt = 1 jest odniesiony do obszarów rolniczych (najmniejszy koszt prac ziemnych / budowlanych dla 1 piksela). Koszty względne dla innych obszarów są obliczane jako wielokrotność kosztów dla terenów rolniczych. Przypisane koszty dla wszystkich kategorii użytkowania terenu dla doprowadzenia do farmy przyłącza przedstawia tabela.

Następnie, dla każdego obiektu połączonej warstwy zawierającej pokrycie terenu gminy dodano odpowiedni koszt, na podstawie pola `x_kod`, mówiącego o typie terenu, jaki stanowi. Warstwę następnie zamieniamy na raster, wykorzystując do tego nowo dodane pole z kosztem. Tworzymy mapę kosztów skumulowanych (cost map) oraz mapę kierunków (backlink raster) z użyciem narzędzia Cost Distance. Wykorzystując utworzone mapy, tworzymy ścieżkę przyłącza o najmniejszym koszcie (cost path). Rasterowa ścieżka zostaje przekonwertowana na warstwę wektorową.



Mapa kosztów względnych

#### 4.11.2 Kod

```

1 pt_merged_layer = "pt_merged_layer"
2 arcpy.management.MakeFeatureLayer(pt_merged, pt_merged_layer)
3 arcpy.management.AddField(pt_merged_layer, "cost", "FLOAT")
4 arcpy.management.CalculateField(
5     in_table=pt_merged_layer,
6     field="cost",
7     expression="costs.get(!x_kod!, 0)",
8     expression_type="PYTHON3",
9     code_block="""costs = {
10     "PTWP01": 0,
11     "PTWP02": 200,
12     "PTWP03": 0,
13     "PTZB02": 100,
14     "PTZB01": 200,
15     "PTZB05": 50,
16     "PTZB04": 200,
17     "PTZB03": 200,
18     "PTLZ01": 100,
19     "PTLZ02": 50,
20     "PTLZ03": 50,
21     "PTRK01": 15,
22     "PTRK02": 15,
23     "PTUT03": 100,
24     "PTUT02": 90,
25     "PTUT04": 20,
26     "PTUT05": 20,
27     "PTUT01": 0,
28     "PTTR02": 1,

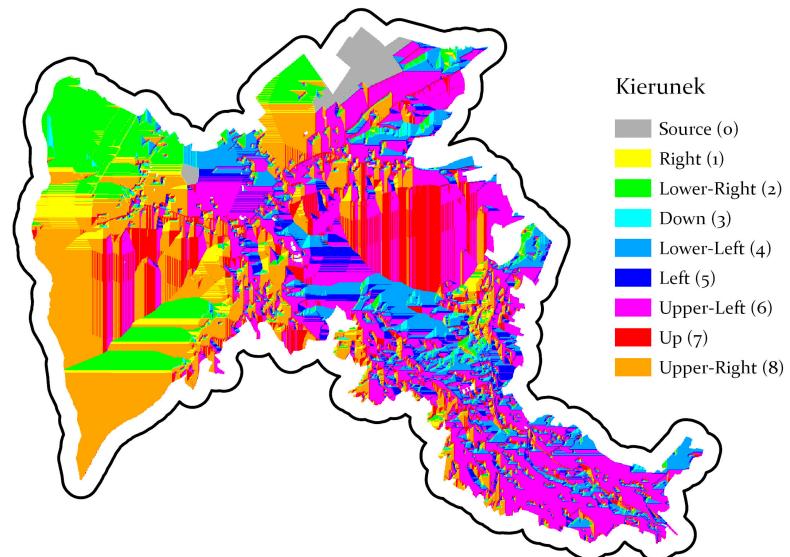
```

```

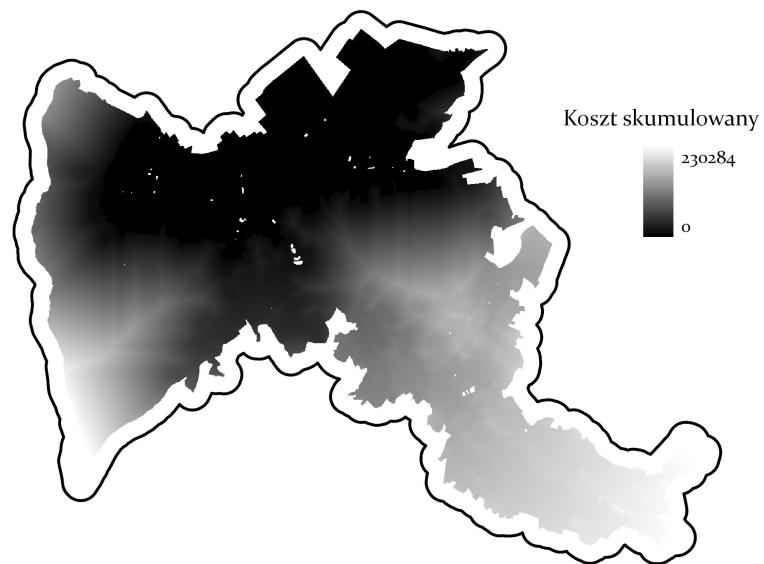
29     "PTTR01": 20,
30     "PTKM02": 200,
31     "PTKM01": 100,
32     "PTKM03": 200,
33     "PTKM04": 0,
34     "PTGN01": 1,
35     "PTGN02": 1,
36     "PTGN03": 1,
37     "PTGN04": 1,
38     "PTPL01": 50,
39     "PTS001": 0,
40     "PTS002": 0,
41     "PTWZ01": 0,
42     "PTWZ02": 0,
43     "PTNZ01": 150,
44     "PTNZ02": 150
45   }"""
46 )
47
48 out_cost = arcpy.conversion.PolygonToRaster(
49   in_features=pt_merged_layer,
50   value_field="cost",
51   out_rasterdataset=f"{geobaza}\\"{wariant}_cost_raster",
52   cell_assignment="CELL_CENTER",
53   cellsize=5
54 )
55
56 out_distance = arcpy.sa.CostDistance(
57   in_source_data=f"{geobaza}\\"{wariant}_dzialki_przydatne_powyzej_{
      prog_przydatnosci}",
58   in_cost_raster=f"{geobaza}\\"{wariant}_cost_raster",
59   maximum_distance=None,
60   out_backlink_raster=f"{geobaza}\\"{wariant}_cost_backlink",
61   source_cost_multiplier=None,
62   source_start_cost=None,
63   source_resistance_rate=None,
64   source_capacity=None,
65   source_direction=""
66 )
67 out_distance.save(f"{geobaza}\\"{wariant}_cost_distance")
68
69 out_path = arcpy.sa.CostPath(
70   in_destination_data=linie_elektroenergetyczne ,
71   in_cost_distance_raster=f"{geobaza}\\"{wariant}_cost_distance",
72   in_cost_backlink_raster=f"{geobaza}\\"{wariant}_cost_backlink",
73   path_type="BEST_SINGLE",
74   force_flow_direction_convention="INPUT_RANGE"
75 )
76 out_path.save(f"{geobaza}\\"{wariant}_cost_path")
77
78 path_vector = arcpy.conversion.RasterToPolyline(in_raster=out_path,
79   out_polyline_features=f"{geobaza}\\"{wariant}_cost_path_vector")

```

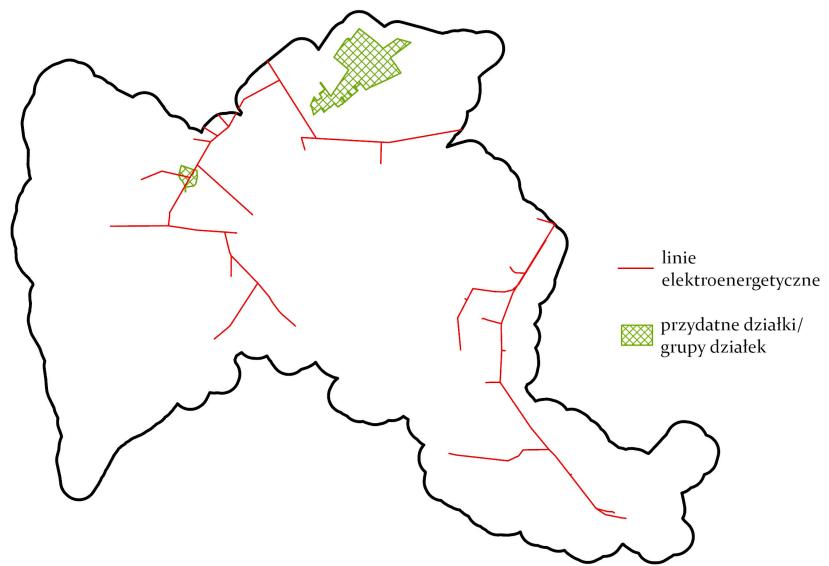
#### 4.11.3 Wynik



Mapa kierunków (backlink)



Mapa kosztów skumulowanych



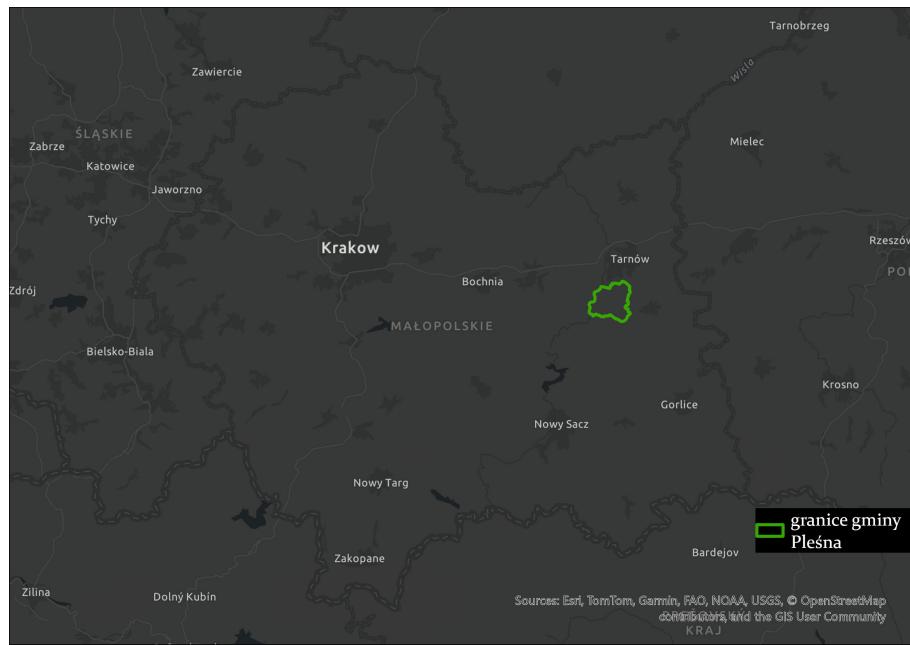
Mapa przedstawiająca przydatne działki oraz linie elektroenergetyczne

#### 4.11.4 Ostateczny wybór działki

## 5 Test modelu na danych z innego obszaru

### 5.1 Opis obszaru

Stworzony model należało przetestować na innym obszarze. Wybrano do tego celu gminę Pleśna - wiejską gminę w powiecie tarnowskim w województwie małopolskim.

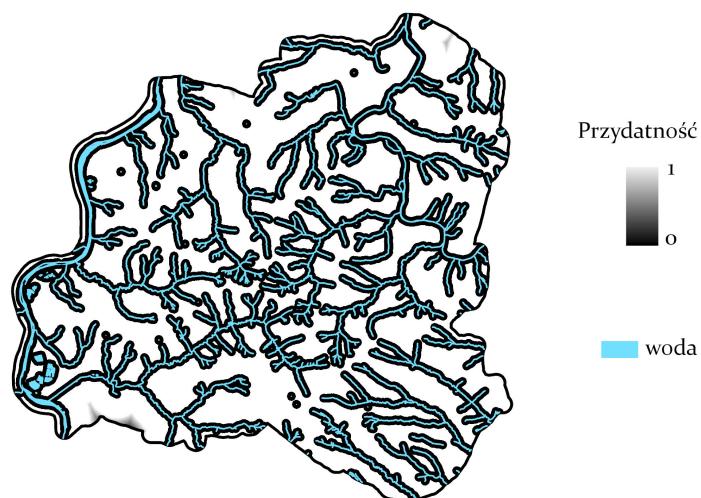


Położenie gminy na mapie województwa małopolskiego

## 5.2 Kryterium 1: odległość od rzek i zbiorników wodnych

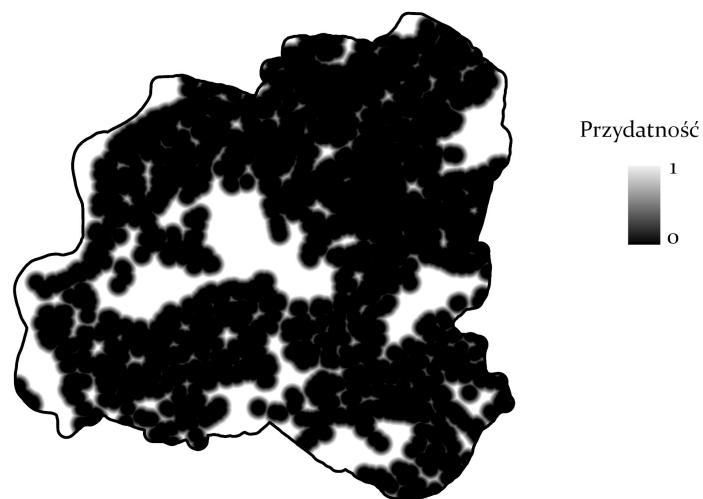


Mapa przydatności dla kryterium 1.

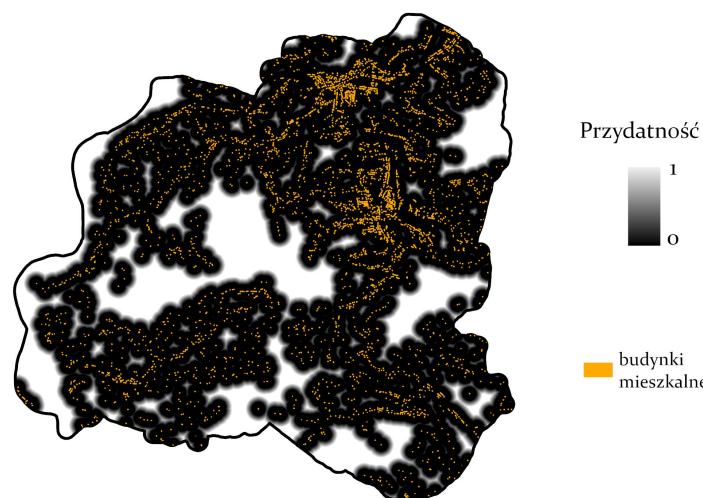


Mapa przydatności dla kryterium 1. zawierająca rzeki oraz zbiorniki wodne

### 5.3 Kryterium 2: odległość od budynków mieszkalnych

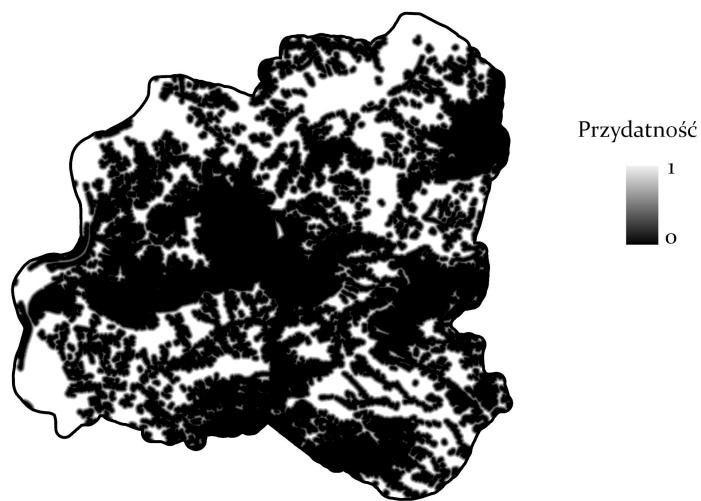


Mapa przydatności dla kryterium 2.

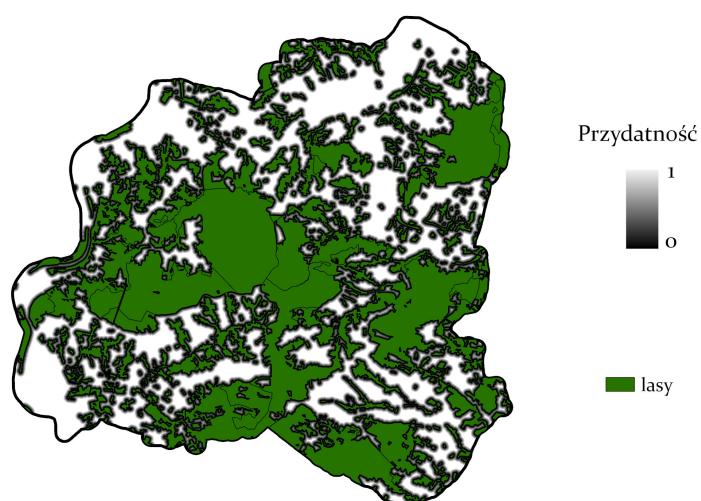


Mapa przydatności dla kryterium 2. zawierająca budynki mieszkalne

#### 5.4 Kryterium 3: pokrycie terenu

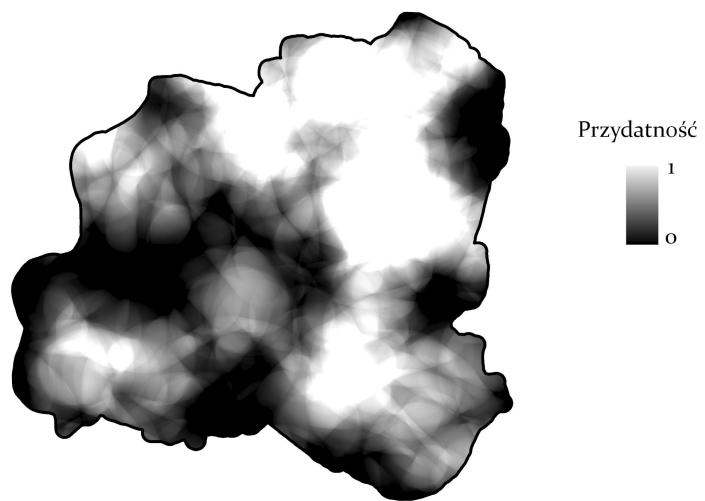


Mapa przydatności dla kryterium 3.

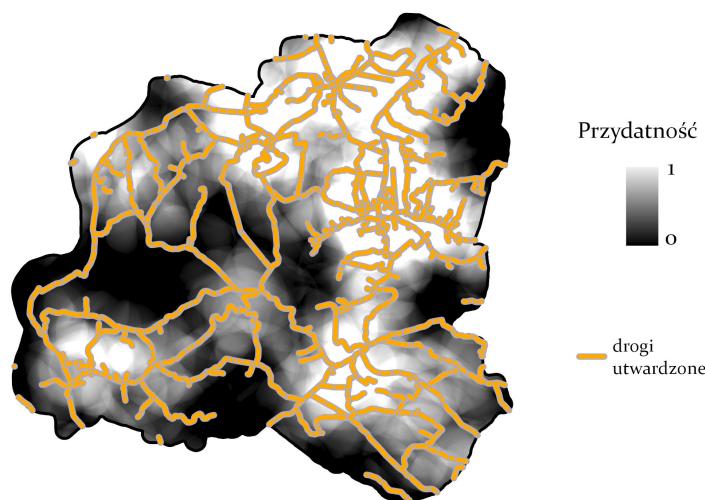


Mapa przydatności dla kryterium 3. zawierająca lasy

## 5.5 Kryterium 4: dostęp do dróg utwardzonych

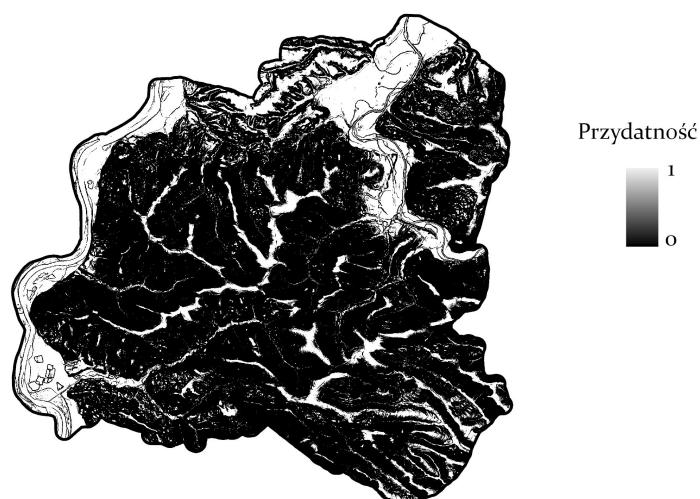


Mapa przydatności dla kryterium 4.

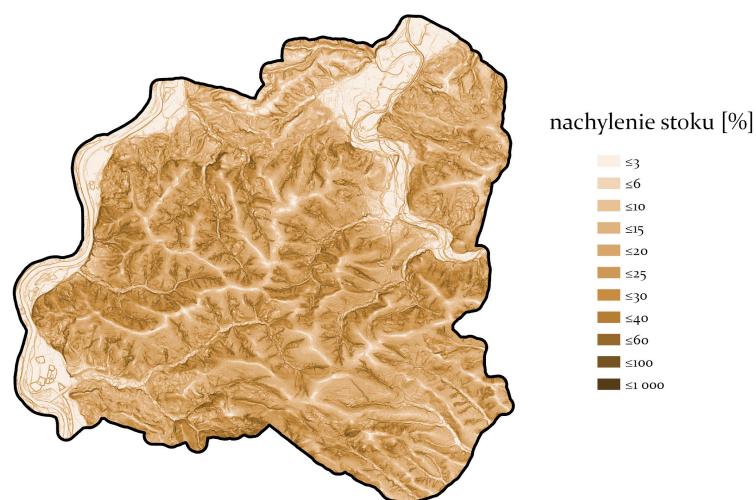


Mapa przydatności dla kryterium 4. zawierająca drogi utwardzone

## 5.6 Kryterium 5: nachylenie stoków

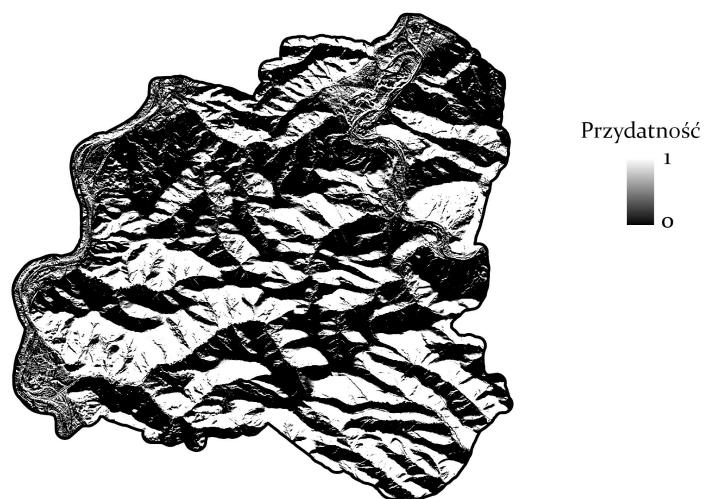


Mapa przydatności dla kryterium 5.

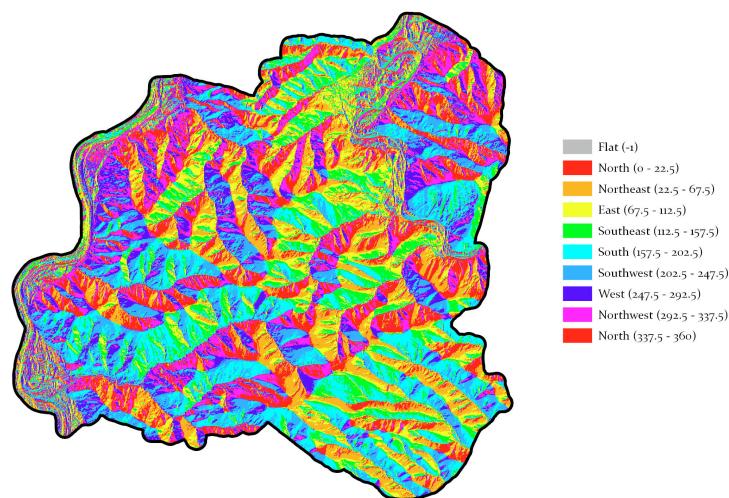


Mapa nachyleń stoków wykorzystywana podczas sprawdzania kryterium 5.

## 5.7 Kryterium 6: dostęp do światła słonecznego

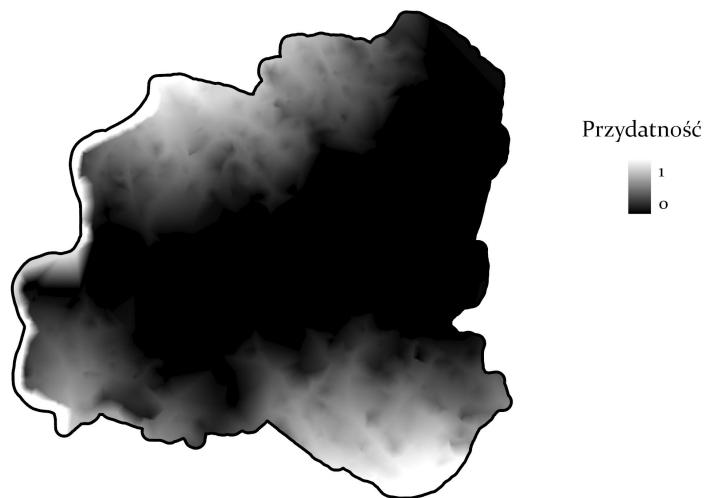


Mapa przydatności dla kryterium 6.

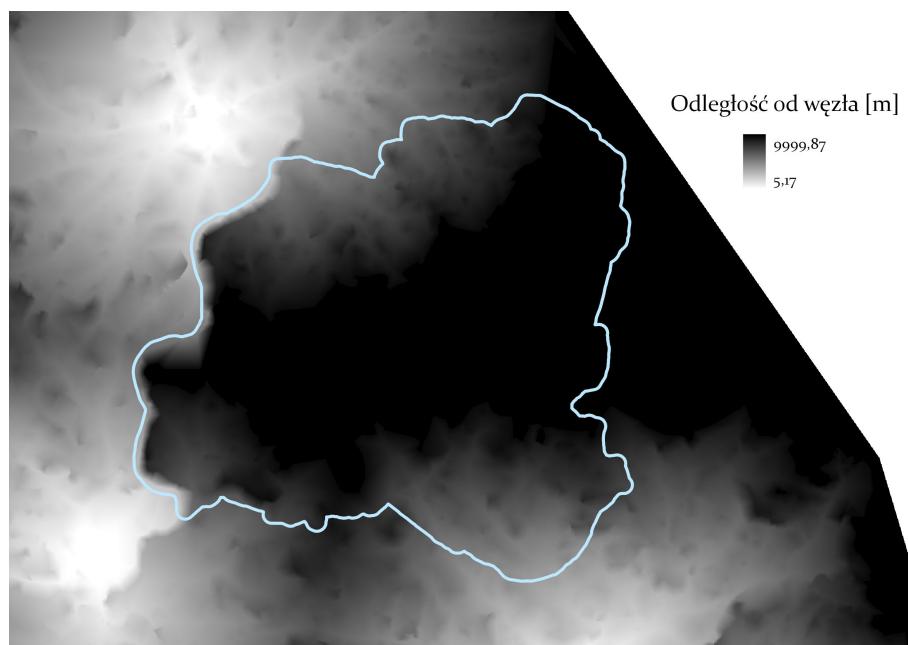


Mapa przydatności dla kryterium 6. zawierająca stopień wystawy słonecznej

## 5.8 Kryterium 7: dojazd do istotnych drogowych węzłów komunikacyjnych

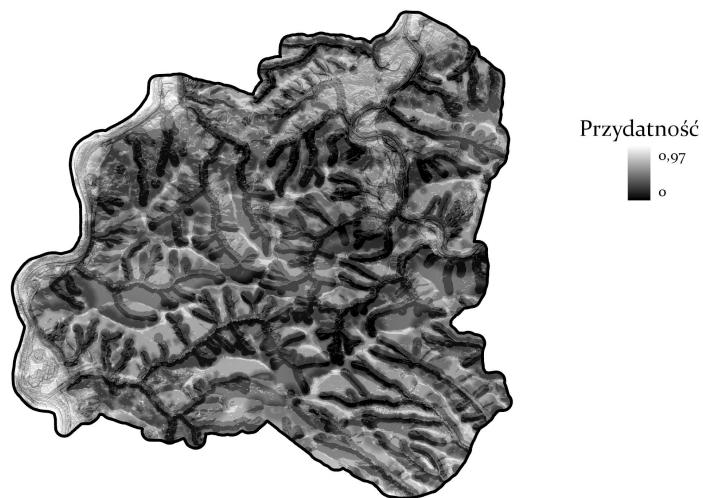


Mapa przydatności dla kryterium 7.

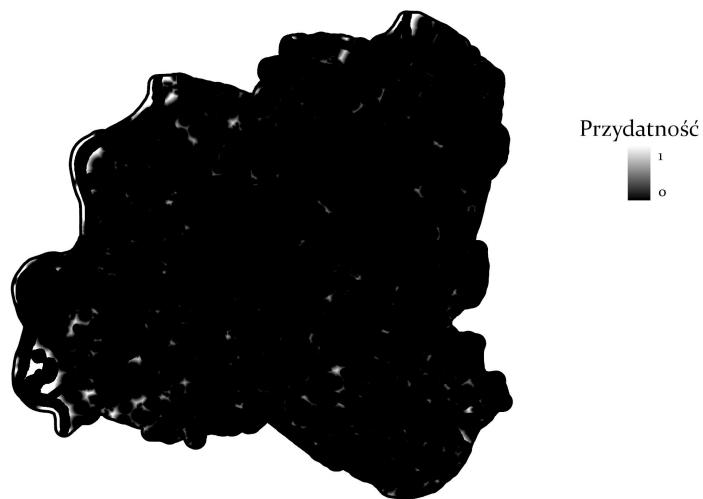


Mapa odległości od węzłów

## 5.9 Ocena przydatności terenu



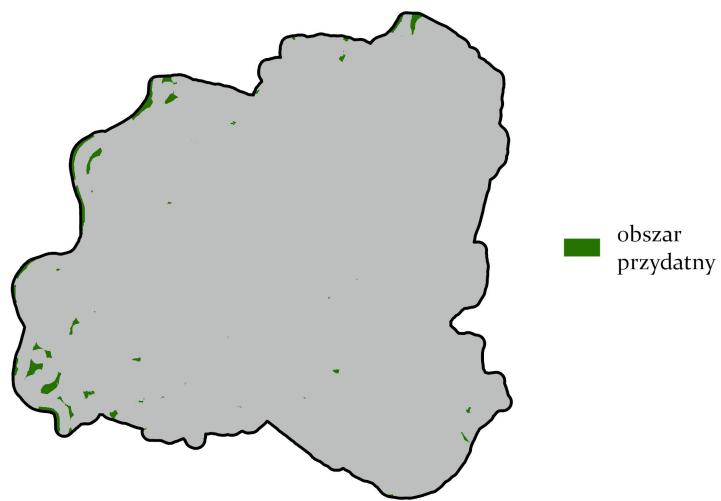
Suma kryteriów rozmytych



Suma kryteriów ostrych



Wynik łączenia kryteriów ostrych i rozmytych



Wynik łączenia kryteriów ostrych i rozmytych po reklasyfikacji

## 5.10 Wybór przydatnych działek



Mapa przedstawiająca przydatne działki

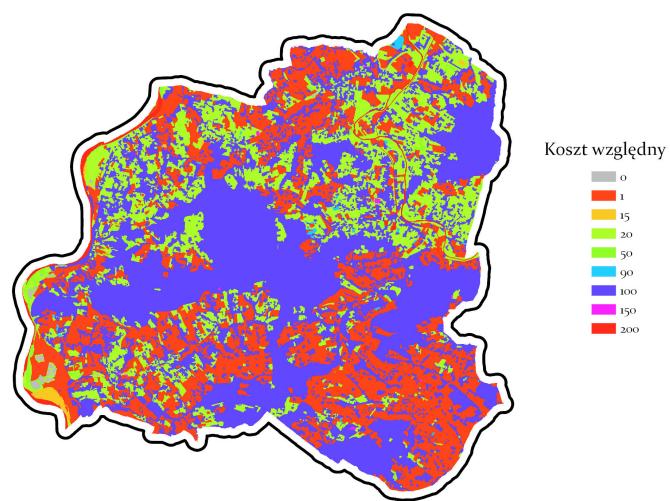


Mapa przedstawiająca przydatne działki na ortofotomapie

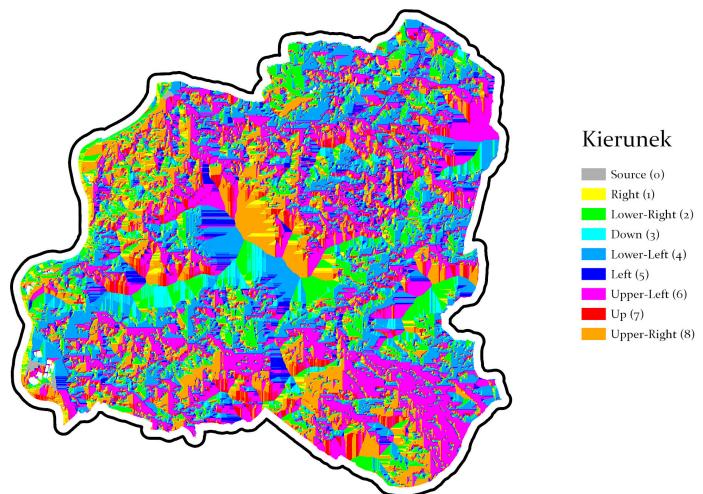


Mapa przedstawiająca przydatne działki na ortofotomapie

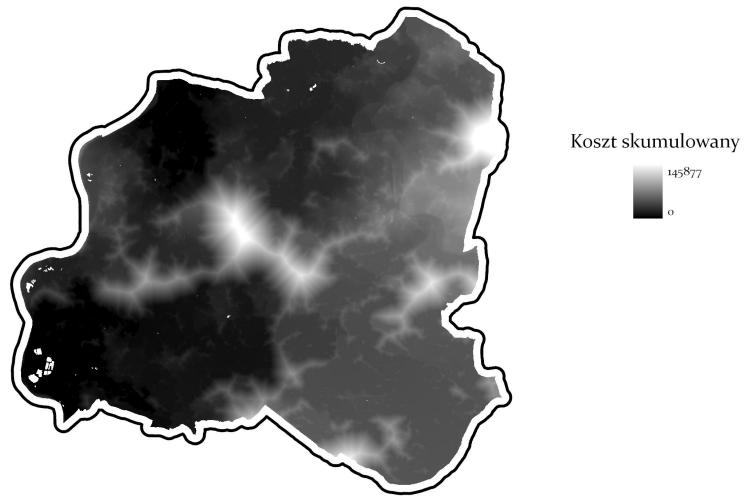
### 5.11 Przyłącze do sieci SN



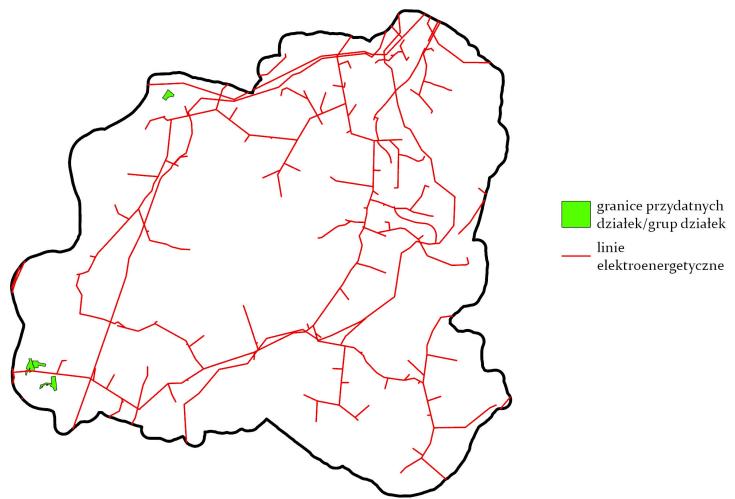
Mapa kosztów względnych



Mapa kierunków (backlink)



Mapa kosztów skumulowanych



Mapa przedstawiająca przydatne działki oraz linie elektroenergetyczne