

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

Adrian Fabisiewicz (328935)

15 stycznia 2025

## Spis treści

# 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\MyProject12.gdb"
2 arcpy.env.workspace = "in_memory"
3 arcpy.env.outputCoordinateSystem = arcpy.SpatialReference("ETRS_1989_Poland_CS92")
4 arcpy.env.extent = f'{geobaza}\gmina_buffer'
5 arcpy.env.mask = f'{geobaza}\gmina_buffer'
6 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 celu złączenia 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

Jako dane wejściowe do rozpatrzenia kryterium pierwszego wykorzystano warstwy zawierające obiekty o kodzie OT\_SWRS\_L z BDOT10k, reprezentujące rzeki i strumienie, oraz obiekty o kodzie OT\_PTWP\_A, reprezentujące wody powierzchniowe. W poprzednim, kroku, przygotowując dane, wokół obiektów liniovych stworzono centymetrowy bufor, aby zamienić je na obiekty poligonalne i móc połączyć je w jedną klasę z obiektami reprezentującymi wody powierzchniowe.

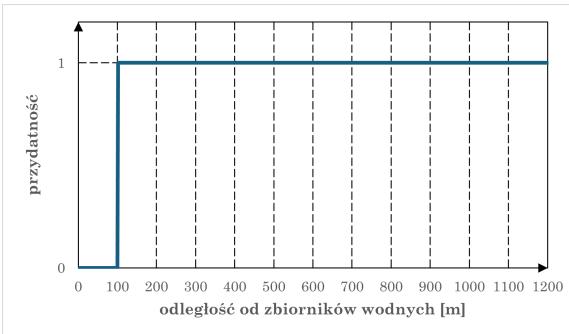
Pierwszym krokiem analizy dla tego kryterium było wykonanie mapy odległości od wód z użyciem narzędzia *DistanceAccumulation* z zestawu narzędzi *Spatial Analyst*.

```
1 out_distance_accumulation_raster = arcpy.sa.DistanceAccumulation(in_source_data=water)
```

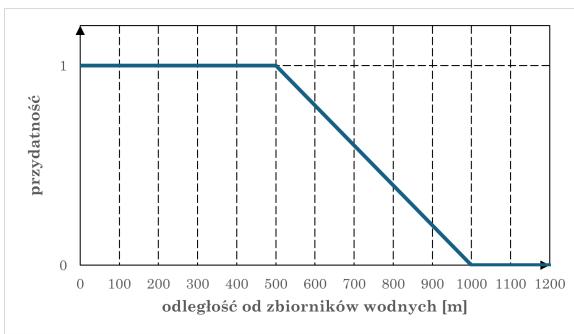
Następnie, z użyciem narzędzia *FuzzyMembership* zbadano członkostwo poszczególnych komórek, wykorzystując do tego utworzony w poprzednim kroku raster. W pierwszej funkcji, rosnącej, przyporządkowano komókom o wartościach od 0 do 100 wartość 0, komókom o wartościach od 100 do 102 przyporządkowano wartości rosnące liniowo do 1, a wartości od 102 w górę zreklasyfikowano na 1. Druga funkcja, malejąca, przyporządkowywała komórkom odległym od wód o ponad 1000 metrów wartość 0, komókom odległym od 1000 do 500 metrów - rosnąco wartości od 0 do 1, a komórki bliższe niż 500 metrów od wód przyjęły wartość 1.

```
1 woda_rosnaca = arcpy.sa.FuzzyMembership(out_distance_accumulation_raster,
    ↪ fuzzy_function="LINEAR 100 102")
2 woda_malejaca = arcpy.sa.FuzzyMembership(out_distance_accumulation_raster,
    ↪ fuzzy_function="LINEAR 1000 500")
```

Reklasyfikacji dokonano zgodnie z poniższymi rysunkami.



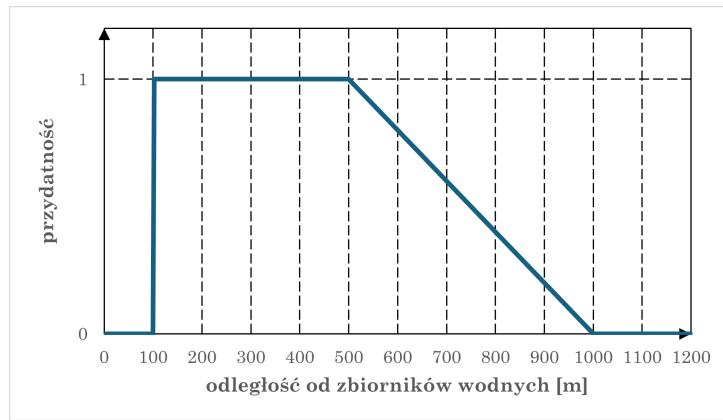
Rysunek 1: Funkcja rosnąca



Rysunek 2: Funkcja malejąca

Ostateczny wynik kryterium pierwszego uzyskano po połączeniu obu warstw utworzonych wcześniej przy użyciu fuzzy logic. Wykorzystano do tego narzędzie *FuzzyOverlay*.

```
1 woda_mapa = arcpy.sa.FuzzyOverlay([woda_rosnaca, woda_malejaca], 'AND')
2 woda_mapa.save(f'{geobaza}\\{kryterium_1}')
```



Rysunek 3: Reklasyfikacja dla kryterium 1.

Wynikiem analizy dla kryterium pierwszego była poniższa mapa. Wyklucza ona z przydatności wąską część obszaru położoną wokół wód. Reszta obszaru, ze względu na dosyć szeroki zakres odległości od wody, dla których przydatność reklasyfikowano na 1, ma korzystne warunki do inwestycji.

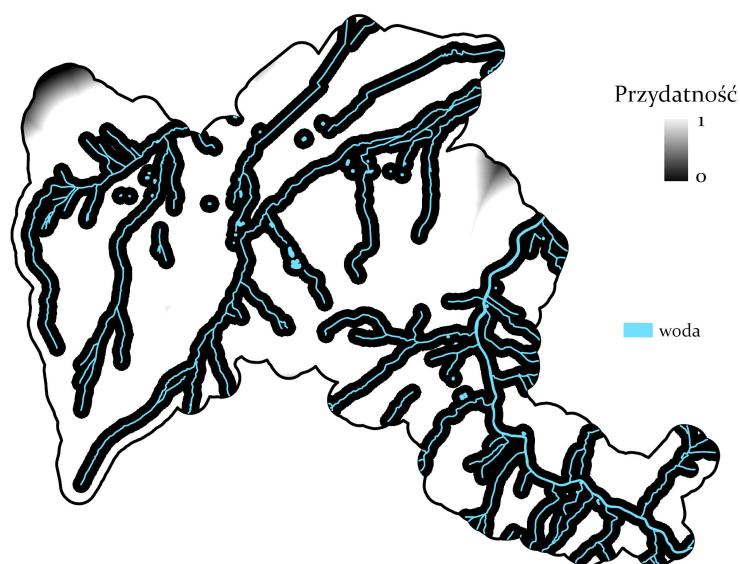


Rysunek 4: Mapa przydatności dla kryterium 1.

Mapę zapisano do geobazy w celu użycia w późniejszym etapie.

```
1 woda_mapa.save(f'{geobaza}\\\\kryterium_1')
```

Poniżej zaprezentowano mapę przedstawiającą rzeki i wody powierzchniowe na tle utworzonej mapy przydatności.



Rysunek 5: Mapa przydatności dla kryterium 1. zawierająca rzeki oraz zbiorniki wodne

### 4.3 Kryterium 2: odległość od budynków mieszkalnych

W celu rozpatrzenia kryterium odległościowego od budynków mieszkalnych, pobrano z bazy BDOT10k poligonową warstwę o kodzie OT\_BUBD\_A, reprezentującą budynki. Następnie wybrano z nich jedynie te będące budynkami mieszkalnymi. Skorzystano w tym celu z atrybutu *FOBUD* warstwy, informującego o funkcji ogólnej budynku. Budynki mieszkalne przyjmowały w tym polu wartość *budynki mieszkalne*. Skonstruowano więc zapytanie i wybrano zgodne obiekty, korzystając z funkcji *Select* z zestawu Analysis.

```
1 arcpy.analysis.Select(budynki, 'budynki_mieszkalne', "FOBUD = 'budynki mieszkalne'")
```

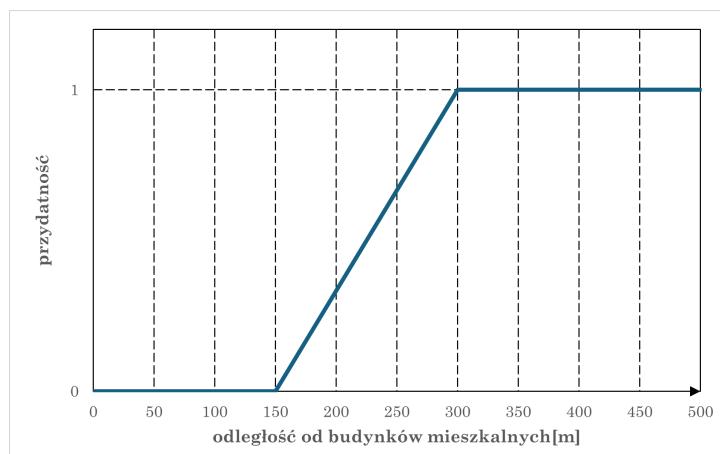
Następnie utworzono raster reprezentujący odległość od budynku mieszkalnego dla każdej komórki z użyciem narzędzia *DistanceAccumulation*.

```
1 out_distance_accumulation_buildings =
  arcpy.sa.DistanceAccumulation(in_source_data='budynki_mieszkalne')
```

Dla utworzonej warstwy dokonano określenia członkostwa z użyciem rosnącej funkcji liniowej. Funkcja każdej komórce odległej od budynku mieszkalnego o mniej niż 150 metrów przyporządkowywała zerową przydatność. Dla obszarów odległych od 150 do 300 metrów przydatność liniowo rosła, i od 300 metrów w górę przyjmowała maksymalną przydatność.

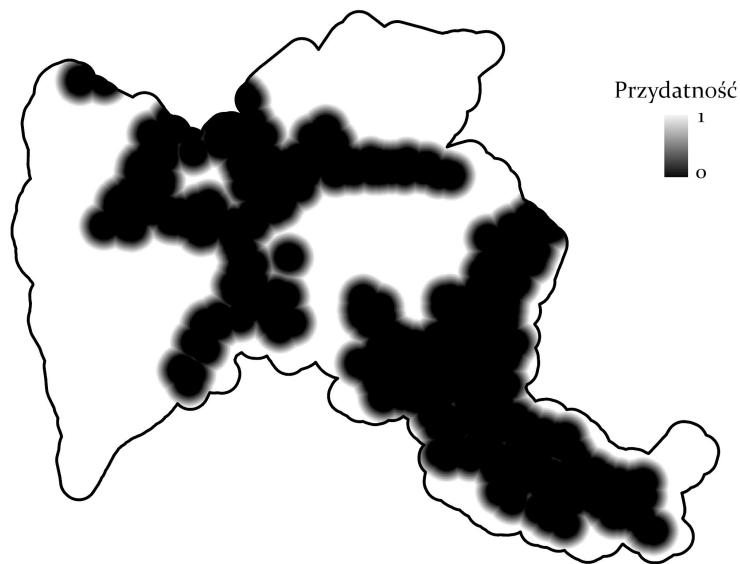
```
1 budynki_mieszkalne = arcpy.sa.FuzzyMembership(out_distance_accumulation_buildings,
  fuzzy_function="LINEAR 150 300")
```

Reklasyfikacji dokonano zgodnie z funkcją przedstawioną poniżej.



Reklasyfikacja dla kryterium 2.

Wynikiem analizy dla kryterium drugiego była poniższa mapa. Wyklucza ona z przydatności znaczną część obszaru położoną w południowej oraz środkowo-zachodniej części gminy. Północna, zachodnia oraz centralna część obszaru przyjmuje maksymalną przydatność.

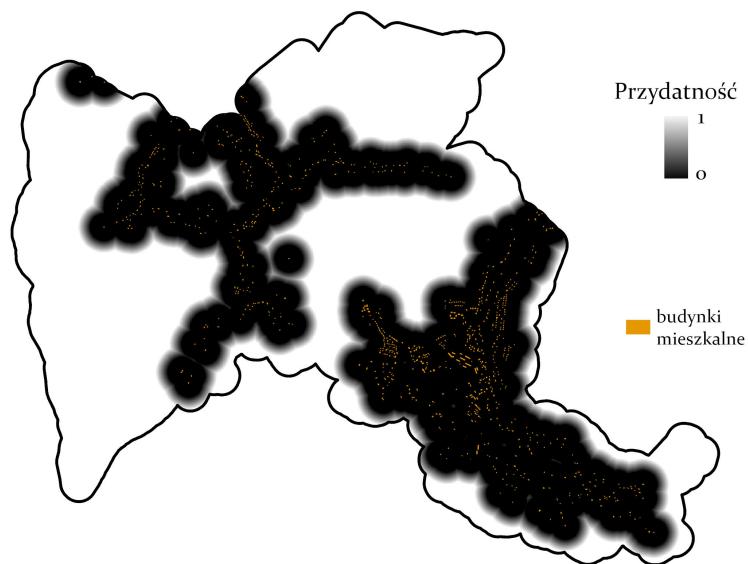


Rysunek 6: Mapa przydatności dla kryterium 2.

Mapę zapisano do geobazy w celu użycia w późniejszym etapie.

```
1 budynki_mieszkalne.save(f'{geobaza}\\kryterium_2')
```

Poniżej zaprezentowano mapę przedstawiającą budynki mieszkalne na tle utworzonej mapy przydatności.



Rysunek 7: Mapa przydatności dla kryterium 2. zawierająca budynki mieszkalne

#### 4.4 Kryterium 3: pokrycie terenu

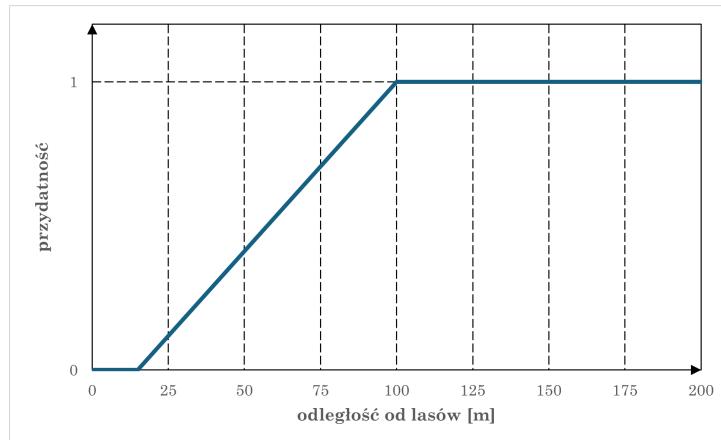
Warstwą wejściową w kryterium trzecim, była warstwa OT\_PTLZ\_A zawierająca tereny leśne i zadrzewione. W celu obliczenia odległości komórek od terenów leśnych, wykorzystano funkcję *DistanceAccumulation*.

```
1 out_distance_accumulation_ptlz = arcpy.sa.DistanceAccumulation(in_source_data=ptlz)
```

Następnie określono poziom członkostwa z użyciem *FuzzyMembership*. Funkcja każdej komórce odległej od lasu o mniej niż 15 metrów przyporządkowywała zerową przydatność. Dla obszarów odległych od 15 do 100 metrów przydatność liniowo rosła, przyjmując od 100 metrów w góre maksymalną przydatność.

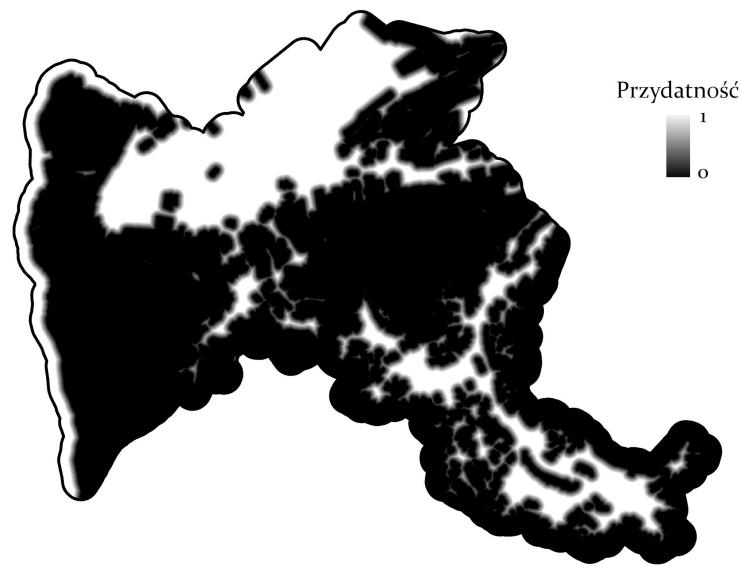
```
1 lasy_fuzzy = arcpy.sa.FuzzyMembership(out_distance_accumulation_ptlz,
    ↴ fuzzy_function="LINEAR 15 100")
```

Reklasyfikacji dokonano zgodnie z funkcją przedstawioną poniżej.



Reklasyfikacja dla kryterium 3.

Wynikiem analizy dla kryterium drugiego była poniższa mapa. Ze względu na wysoką lesistość lub niewielką odległość od lasu większość obszaru gminy została wyeliminowana. Przydatny pozostał obszar w północnej części gminy oraz częściowo w południowo-wschodniej.

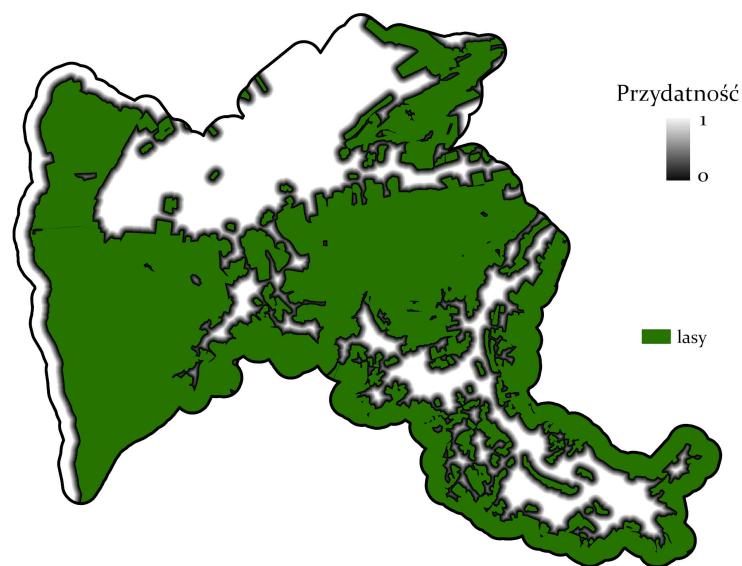


Rysunek 8: Mapa przydatności dla kryterium 3.

Mapę zapisano do geobazy w celu użycia w późniejszym etapie.

```
1 budynki_mieszkalne.save(f'{geobaza}\\{kryterium_3}')
```

Poniżej zaprezentowano mapę przedstawiającą lasy na tle utworzonej mapy przydatności.



Rysunek 9: Mapa przydatności dla kryterium 3. zawierająca lasy

## 4.5 Kryterium 4: dostęp do dróg utwardzonych

Drogi pobrano z bazy BDOT10k, wybierając obiekty o klasie OT\_SKDR\_L. Ze wszystkich dróg wybrano jedynie te utwardzone, tj. których materiał nawierzchni to beton, bruk, kostka kamienna, kostka prefabrykowana, masa bitumiczna lub płyty betonowe.

```
1 drogi_utwardzone = arcpy.analysis.Select(drogi, 'drogi_utwardzone', "MATE_NAWIE IN
→ ('beton', 'bruk', 'kostka kamienna', 'kostka prefabrykowana', 'masa bitumiczna',
→ 'płyty betonowe')")
```

Korzystając z narzędzia *LineDensity* utworzono raster z gęstością dróg utwardzonych na km<sup>2</sup>.

```
1 density = arcpy.sa.LineDensity(
2     in_polyline_features=drogi_utwardzone,
3     population_field=None,
4     cell_size=5,
5     search_radius=1000,
6     area_unit_scale_factor="SQUARE_KILOMETERS",
7 )
```

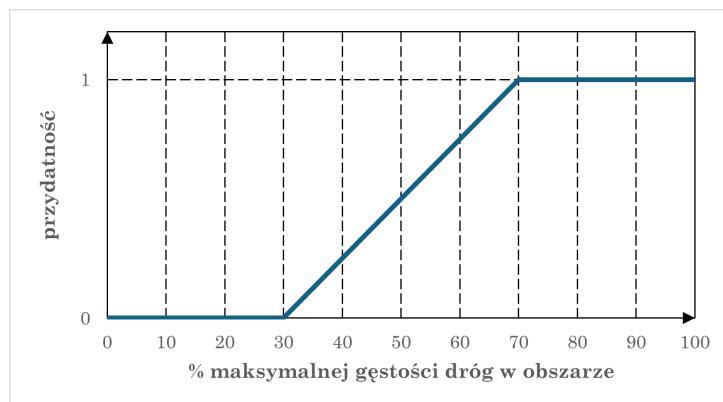
Aby móc uzależnić funkcję reklasyfikacyjną od wartości rastra, liczy się maksymalna wartość gęstości dróg w obszarze.

```
1 arcpy.management.CalculateStatistics(density)
2 max_value = density.maximum
```

Zostaje dokonana reklasyfikacja zgodnie z funkcją liniową. Wartości gęstości od 0 do 30% przyjmują zerową przydatność, od 30% do 70% liniowo rosną do maksymalnej przydatności, od 70% w górę przyjmując maksymalną przydatność.

```
1 kryterium_4 = arcpy.sa.RescaleByFunction(
2     in_raster=density,
3     transformation_function=f"LINEAR {0.3 * max_value} {0.7 * max_value}",
4     from_scale=0,
5     to_scale=1
6 )
```

Reklasyfikacji dokonano zgodnie z funkcją przedstawioną poniżej.

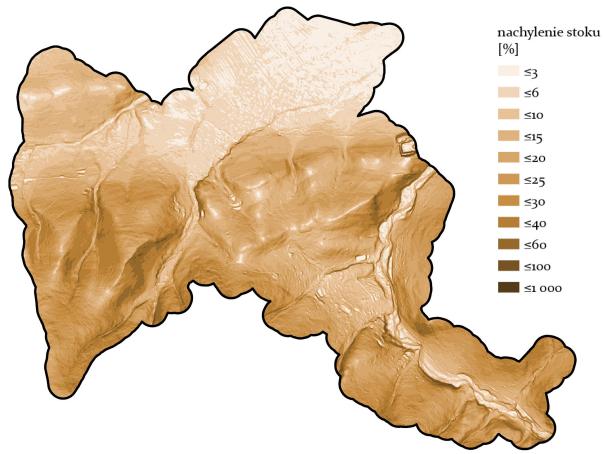


Rysunek 10: Reklasyfikacja dla kryterium 4.

## 4.6 Kryterium 5: nachylenie stoków

W celu realizacji kryterium pobrano kafelki NMT dla obszaru gminy. Połączono je z użyciem narzędzia *Mosaic To New Raster*. Dla nowo powstałego rastra obliczono procentowe nachylenia stoków z użyciem narzędzie *Slope* z zestawu *3D Analyst*.

```
1 arcpy.dsd.Slope(nmt, "slope", "PERCENT_RISE", 1)
```

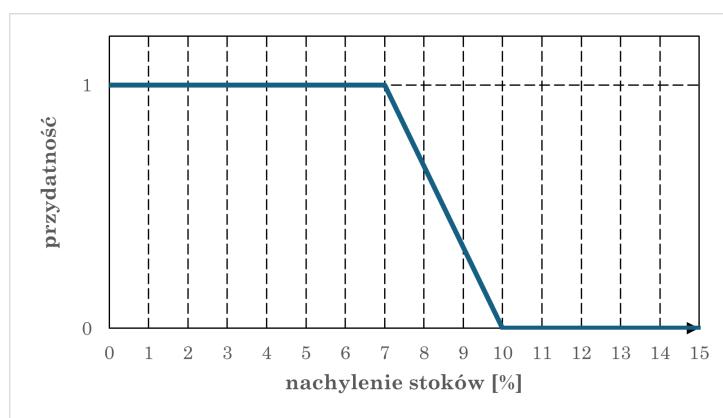


Rysunek 11: Mapa nachyleń stoków powstała z użyciem narzędzie *Slope*

Powstały raster zreklasyfikowano, przyporządkowując stokom o nachyleniu od 0 do 7% maksymalną przydatność, od 7 do 10 % stopniowo spadając, przyjmując dla stoków o nachyleniu powyżej 10% zerową przydatność.

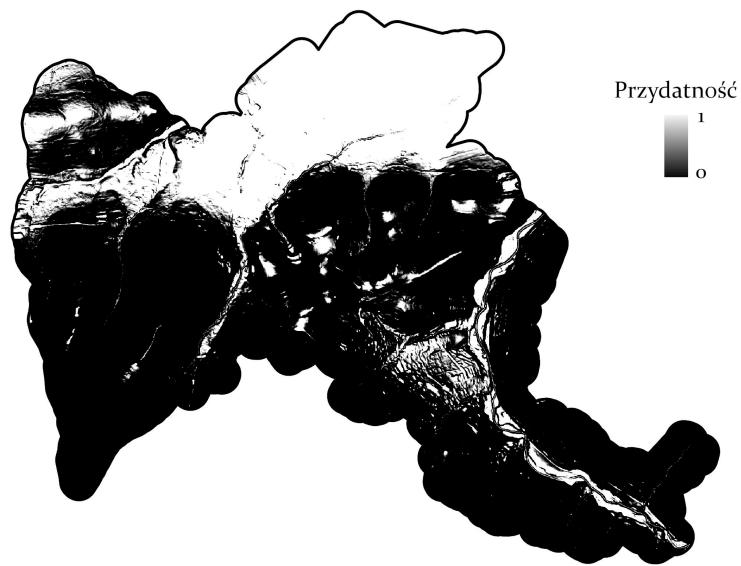
```
1 slope_fuzzy = arcpy.sa.FuzzyMembership("slope", fuzzy_function="LINEAR 10 7")
```

Reklasyfikacji dokonano zgodnie z funkcją przedstawioną poniżej.



Rysunek 12: Reklasyfikacja dla kryterium 5.

Kryterium eliminuje dużą część obszaru ze względu na dużą górzystość terenu. Przydatna pozostaje północna i północno-wschodnia część gminy oraz niewielkie obszary w pozostałej części gminy.



Rysunek 13: Mapa przydatności dla kryterium 5.

Mapę zapisano do geobazy w celu użycia w późniejszym etapie.

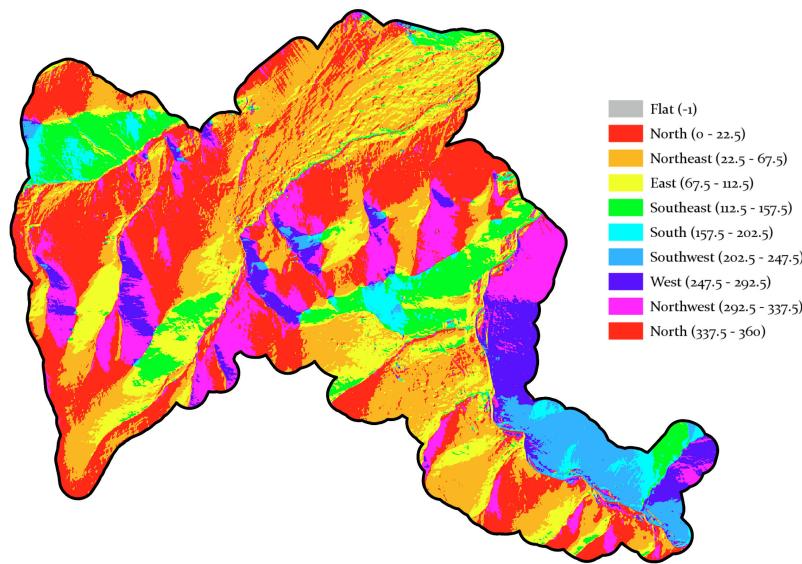
```
1 woda_mapa.save(f'{geobaza}\\{kryterium_5}')
```

## 4.7 Kryterium 6: dostęp do światła słonecznego

Do tego kryterium również wykorzystano NMT, a także narzędzie *Aspect* z zestawu *3D Analyst*.

```
1 aspect = arcpy.ddd.Aspect(nmt)
```

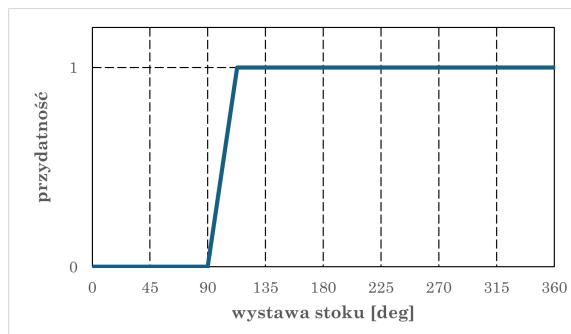
Efektem działania funkcji był poniższy raster, przedstawiający, na którą stronę świata wystawiony jest stok.



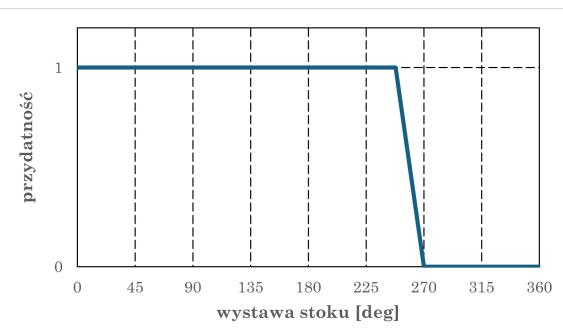
Rysunek 14: Mapa przydatności dla kryterium 6. zawierająca stopień wystawy słonecznej

Przydatne w kontekście naszego projektu są stoki południowo-zachodnie, południowe i południowo-wschodnie. Stworzono więc dwie funkcje reklasyfikacyjne: rosnącą od 90 do 113 stopni oraz malejącą od 248 do 270 stopni.

```
1 aspect_fuzzy = arcpy.sa.FuzzyMembership(aspect, fuzzy_function="LINEAR 90 113")
2 aspect_fuzzy_1 = arcpy.sa.FuzzyMembership(aspect, fuzzy_function="LINEAR 270 248")
```



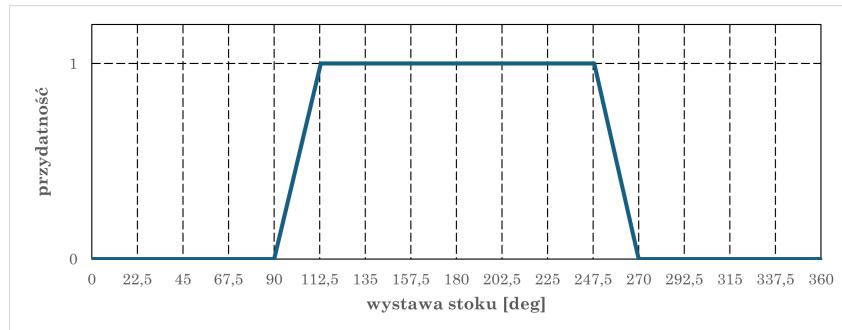
Rysunek 15: Funkcja rosnąca



Rysunek 16: Funkcja malejąca

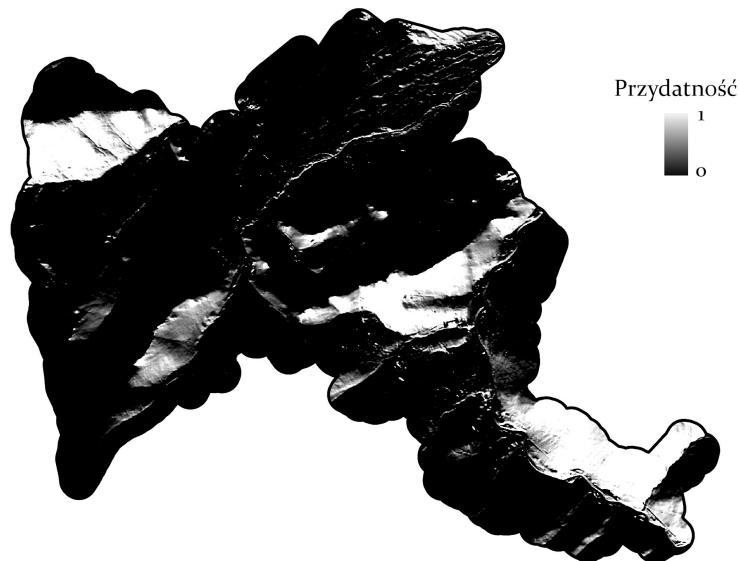
Ostateczny wynik kryterium szóstego uzyskano po połączeniu obu warstw utworzonych wcześniej przy użyciu fuzzy logic. Wykorzystano do tego narzędzie *FuzzyOverlay*.

```
1 aspect_overlay = arcpy.sa.FuzzyOverlay([aspect_fuzzy, aspect_fuzzy_1], 'AND')
```



Rysunek 17: Reklasyfikacja dla kryterium 6.

Kryterium wyłoniło tereny przydatne w północno-zachodniej, środkowej oraz południowo-wschodniej części obszaru. Znaczna część terenów charakteryzuje się niekorzystną wystawą słoneczną.



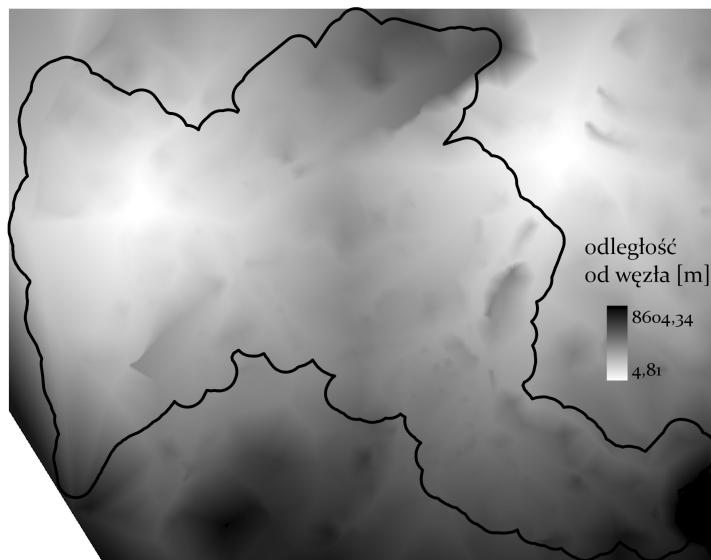
Mapa przydatności dla kryterium 6.

Mapę zapisano do geobazy w celu użycia w późniejszym etapie.

```
1 aspect_overlay.save(f'{geobaza}\\kryterium_6')
```

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

Warstwę rasterową zawierającą istotne drogowe węzły komunikacyjne utworzono manualnie na podstawie warstwy OT\_SKDR\_L z bazy BDOT10k. Przefiltrowano drogi, pozostawiając jedynie najważniejsze z nich - krajowe, wojewódzkie. Utworzono warstwę punktową, na której umieszczono kilka punktów w miejscach styku tych dróg. Następnie, korzystając z wtyczki *QNEAT3 - QGIS Network Analysis Toolbox* 3 oraz dostępnego w niej algorytmu *Iso-Area as Interpolation (from Layer)*. Na podstawie sieci dróg, utworzyła ona raster zawierający dla każdej komórki odległość od węzła komunikacyjnego.

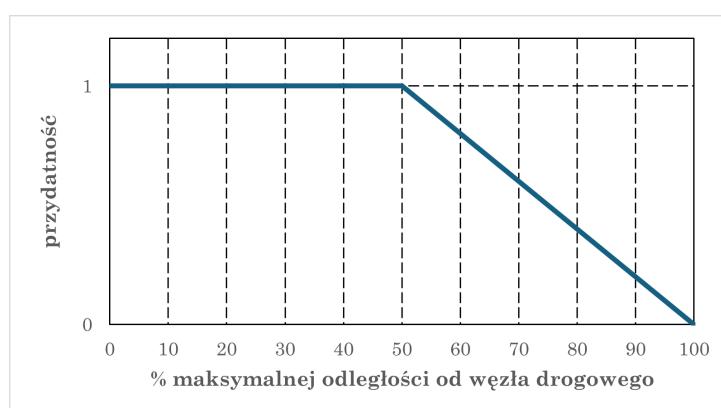


Rysunek 18: Raster utworzony poprzez wtyczkę *QNEAT3*

Dla powyższego rastra obliczono maksymalną odległość od węzła w celu wykorzystania wartości dla funkcji w fuzzy membership.

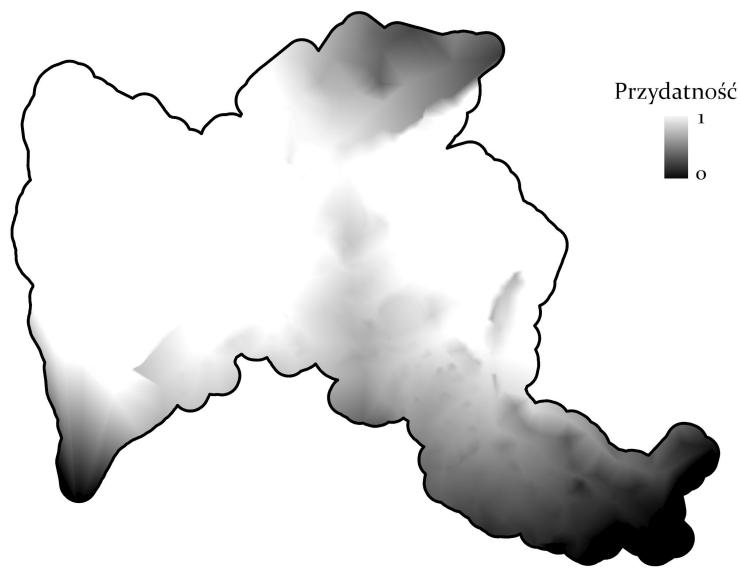
```
1 wezly_max = float(arcpy.management.GetRasterProperties(wezly, "MAXIMUM")[0].replace(',', '.'))
```

Sprawdzono członkostwo na podstawie funkcji liniowej, komórkom od maksymalnej odległości od węzła do połowy maksymalnej odległości od węzła przypisując odpowiednią wartość funkcji, a komórkom położonym bliżej przyporządkowując maksymalną przydatność.



Rysunek 19: Reklasyfikacja dla kryterium 7.

Tereny położone w centralnej części rastra okazały się być najbardziej przydatne. Skraje najbardziej na południe oraz północno-wschodnie mają najniższą przydatność.



Rysunek 20: Mapa przydatności dla kryterium 7.

Mapę zapisano do geobazy w celu użycia w późniejszym etapie.

```
1 wezly_fuzzy.save(f'{geobaza}\\{kryterium_7}')
```

## 4.9 Ocena przydatności terenu

Poniższy kod najpierw tworzy tabelę zawierającą wagę dla każdego z kryteriów, zmienne w zależności od wariantu.

```

1 tabela_kryteriow = arcpy.sa.WSTable([[f'{geobaza}\kryterium_1', "VALUE", waga_woda],
2                                     [f'{geobaza}\kryterium_2', "VALUE", waga_budynki], [f'{geobaza}\kryterium_3',
3                                     "VALUE", waga_lasy], [f'{geobaza}\kryterium_4', "VALUE", waga_drogi],
4                                     [f'{geobaza}\kryterium_5', "VALUE", waga_wysokosc], [f'{geobaza}\kryterium_6',
5                                     "VALUE", waga_aspect], [f'{geobaza}\kryterium_7', "VALUE", waga_wezly]])

```

Następnie tworzy sumę ważoną wszystkich z kryteriów.

```

1 weighted_sum = arcpy.sa.WeightedSum(tabela_kryteriow)
2 weighted_sum.save(f'{geobaza}\{variant}_suma_rozmyte')

```

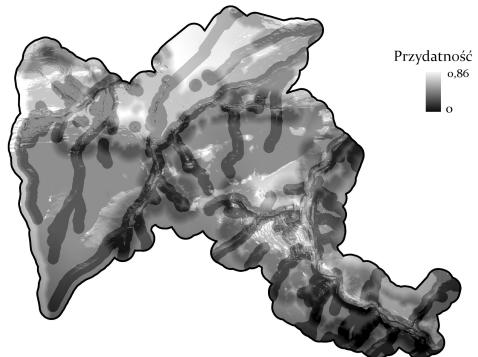
Od teraz do końca sekcji przedstawiane będą podwójne wyniki - pierwszy wynik będzie dla przypadku, gdzie dla każdego kryterium przyjęto tę samą wagę, a drugi dla przypadku, gdzie wagi są różne. Poniżej przedstawiono przyjęte wagi w obu podejściach.

Kryterium	Waga
1 (odległość od rzek i zbiorników wodnych)	0.142857
2 (odległość od budynków mieszkalnych)	0.142857
3 (pokrycie terenu)	0.142857
4 (dostęp do dróg utwardzonych)	0.142857
5 (nachylenie stoków)	0.142857
6 (dostęp światła słonecznego)	0.142857
7 (dobry dojazd do istotnych węzłów komunikacyjnych)	0.142857

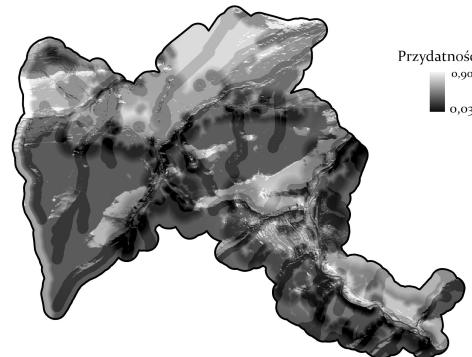
Tabela 2: Tabela z równymi wagami dla kryteriów

Kryterium	Waga
1 (odległość od rzek i zbiorników wodnych)	0.10
2 (odległość od budynków mieszkalnych)	0.15
3 (pokrycie terenu)	0.20
4 (dostęp do dróg utwardzonych)	0.10
5 (nachylenie stoków)	0.15
6 (dostęp światła słonecznego)	0.25
7 (dobry dojazd do istotnych węzłów komunikacyjnych)	0.05

Tabela 3: Tabela z różnymi wagami dla kryteriów



Rysunek 21: Suma kryteriów rozmytych - wagi równe



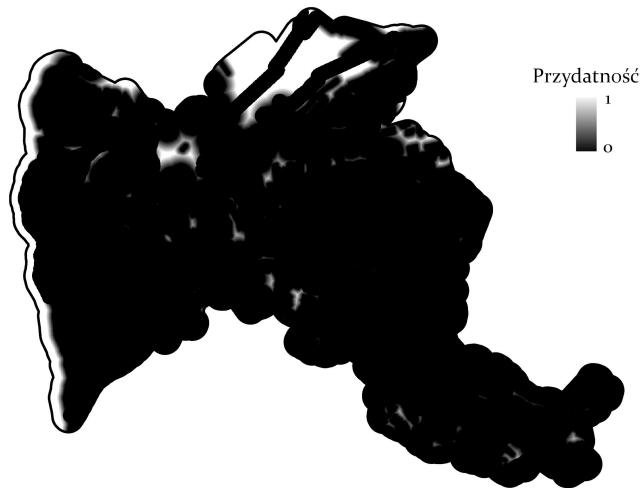
Rysunek 22: Suma kryteriów rozmytych - wagi różne

Później brane są pod uwagę kryteria ostre, tj. 100-metrowa strefa ochronna od wód, 150-metrowa odle-

głość od budynków mieszkalnych oraz 15-metrowa odległość od lasów. W funkcji *FuzzyOverlay* wybrano parametr *AND*, dzięki czemu zostaje utworzony raster, który przyjmuje najniższą możliwą wartość dla komórki.

```

1 kryteria_ostre = arcpy.sa.FuzzyOverlay([woda_rosnaca, f"geobaza]\\kryterium_2",
   ↪ f"geobaza]\\kryterium_3"], 'AND')
2 kryteria_ostre.save(f'geobaza]\\kryteria_ostre')
```

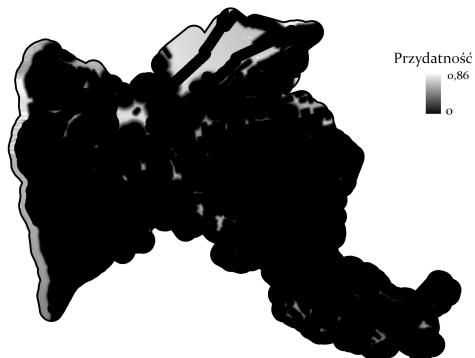


Rysunek 23: Suma kryteriów ostrych

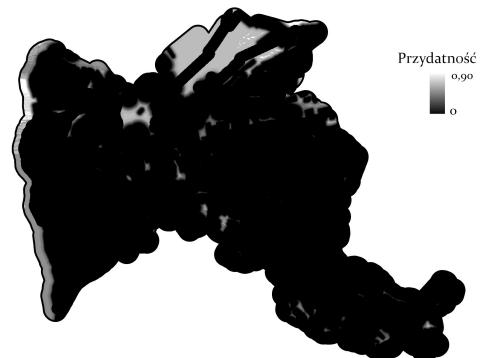
Następnie zostaje utworzony iloczyn kryteriów ostrych i rozmytych. W ten sposób eliminowane z dalszych analiz są komórki wykluczone przez któreś z kryteriów ostrych.

```

1 iloczyn = arcpy.sa.FuzzyOverlay([kryteria_ostre, weighted_sum], 'AND')
2 iloczyn.save(f'geobaza]\\variant_wynik')
```



Rysunek 24: Wynik łączenia kryteriów ostrych i rozmytych - wagi równe



Rysunek 25: Wynik łączenia kryteriów ostrych i rozmytych - wagi różne

Przed reklasyfikacją, obliczana jest maksymalna przydatność na obszarze.

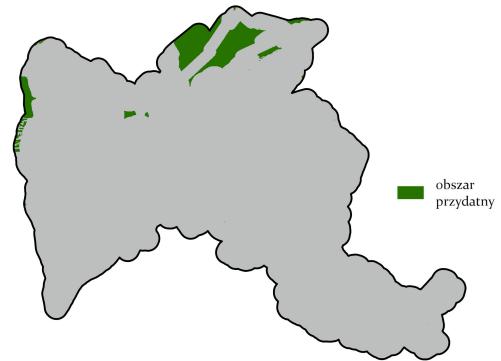
```
1 arcpy.management.CalculateStatistics(iloczyn)
2 max_przydatnosc = iloczyn.maximum
```

Utworzony przez połączenie kryteriów ostrych i rozmytych raster reklasyfikujemy, przyporządkowując komórkom o przydatności powyżej określonego progu przydatności wartość 1, a pozostałym komórkom - wartość 0.

```
1 wynik_reclassified = arcpy.sa.Reclassify(iloczyn, "VALUE", arcpy.sa.RemapRange([[0,
2   ↪ prog_przydatnosci * max_przydatnosc, 0], [prog_przydatnosci * max_przydatnosc, 1,
3   ↪ 1]]))
2 wynik_reclassified.save(f'{geobaza}\\{wariant}_wynik_reclassified')
```



Rysunek 26: Wynik łączenia kryteriów ostrych i rozmytych po reklasyfikacji - wagi równe



Rysunek 27: Wynik łączenia kryteriów ostrych i rozmytych po reklasyfikacji - wagi różne

## 4.10 Wybór przydatnych działek

W pierwszym kroku w tym etapie zamieniamy rastrową warstwę z przydatnością terenu na warstwę poligonową.

```
1 arcpy.conversion.RasterToPolygon(f'{geobaza}\{wariant}_wynik_reclassified',
    ↪ 'poligon_przydatnosci', "NO_SIMPLIFY", "VALUE")
```

Aby wybrać jedynie te poligony, które są terenami przydatnymi, korzystam z narzędzia *Select Layer By Attribute*, wybierając obiekty o wartości 1 w polu *gridcode*.

```
1 arcpy.management.MakeFeatureLayer('poligon_przydatnosci', "poligon_przydatnosci_layer")
2 arcpy.management.SelectLayerByAttribute("poligon_przydatnosci_layer", "NEW_SELECTION",
    ↪ "gridcode = 1")
```

Warstwa z przydatnymi poligonami jest zapisywana w geobazie.

```
1 arcpy.management.CopyFeatures("poligon_przydatnosci_layer",
    ↪ f'{geobaza}\{wariant}_poligon_przydatnosci')
```

Następnie z użyciem funkcji *Summarize Within*, dla każdej działki jest liczona powierzchnia, jaką stanowią dla niej poligony przydatne.

```
1 arcpy.analysis.SummarizeWithin(
2     in_polygons=dzialki,
3     in_sum_features=f'{geobaza}\{wariant}_poligon_przydatnosci',
4     out_feature_class=f'{geobaza}\{wariant}_summarized_within',
5     keep_all_polygons="ONLY_INTERSECTING",
6     shape_unit="SQUAREMETERS",
7     add_group_percent="NO_PERCENT",
8 )
```

Dla każdej działki liczone jest pole *pow\_przyd* informujące o tym, jaki procent działki stanowi teren przydatny, na podstawie pola *sum\_Area\_squaremeters* utworzonego w poprzednim kroku. Wykorzystano do tego narzędzie *Calculate Field*.

```
1 arcpy.management.CalculateField(
2     in_table=f'{geobaza}\{wariant}_summarized_within',
3     field="pow_przyd",
4     expression="100*!sum_Area_squaremeters!/!Shape_Area!",
5     expression_type="PYTHON3",
6     code_block="",
7     field_type="FLOAT"
8 )
```

Utworzone pole jest wykorzystane w narzędziu *Select Layer By Attribute*, gdzie wybierane są działki o procentowej powierzchni przydatnej wyższej od określonego progu.

```
1 dzialki_przydatne_powyzej_progu =
    ↪ arcpy.management.SelectLayerByAttribute(f'{geobaza}\{wariant}_summarized_within',
    ↪ "NEW_SELECTION", f"pow_przyd >= {prog_przydatnosci}")
2 arcpy.management.CopyFeatures(dzialki_przydatne_powyzej_progu,
    ↪ f'{geobaza}\{wariant}_dzialki_przydatne_powyzej')
```

Zgodnie z założeniami projektu, przydatne działki, które ze sobą sąsiadują, są łączone w jedną grupę przy użyciu narzędzi *Dissolve*.

```

1 arcpy.management.Dissolve(
2     in_features=f"{geobaza}\\"{wariant}_dzialki_przydatne_powyzej",
3     out_feature_class=f"{geobaza}\\"{wariant}_dzialki_przydatne_powyzej_dissolve",
4     multi_part="SINGLE_PART",
5     unsplit_lines="DISSOLVE_LINES",
6     concatenation_separator=""
7 )

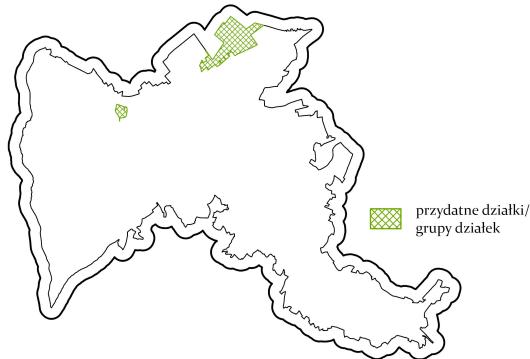
```

W ostatnim kroku wybierane są działki lub grupy działek o łącznej powierzchni powyżej 2ha (20 000 m<sup>2</sup>).

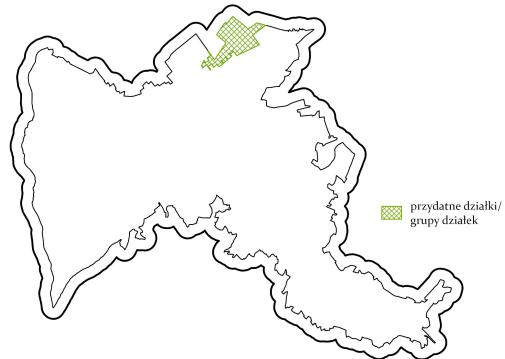
```

1 arcpy.management.MakeFeatureLayer(f"{geobaza}\\"{wariant}_dzialki_przydatne_powyzej_dissolve",
2     f"{geobaza}\\"{wariant}_dzialki_przydatne_powyzej_dissolve_layer")
2 arcpy.management.SelectLayerByAttribute(f"{geobaza}\\"{wariant}_dzialki_przydatne_powyzej_dissolve_layer",
3     "NEW_SELECTION", "Shape_Area >= 20000")
3 arcpy.management.CopyFeatures(f"{geobaza}\\"{wariant}_dzialki_przydatne_powyzej_dissolve_layer",
4     f"{geobaza}\\"{wariant}_grupy_dzialek_przydatne_powyzej_{prog_przydatnosci}")

```



Rysunek 28: Mapa przedstawiająca przydatne działki - wagi równe



Rysunek 29: Mapa przedstawiająca przydatne działki - wagi różne



Rysunek 30: Mapa przedstawiająca przydatne działki na ortofotomapie - wagi równe

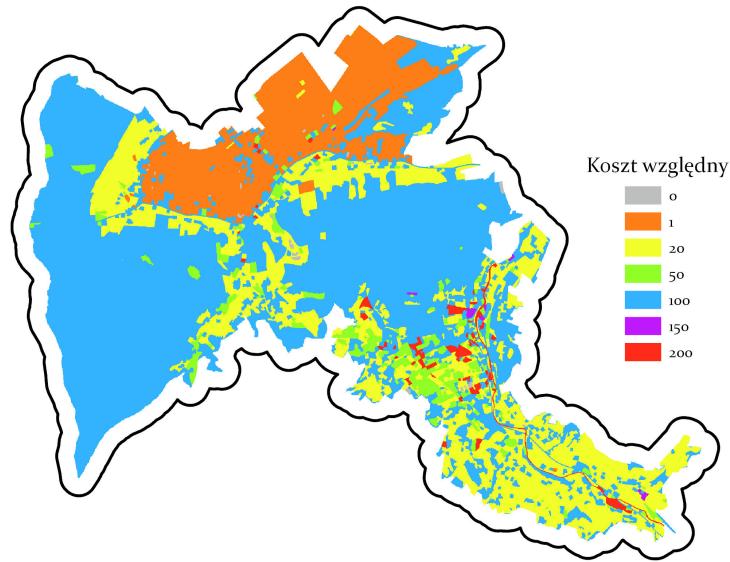


Rysunek 31: Mapa przedstawiająca przydatne działki na ortofotomapie - wagi różne

## 4.11 Koszt przyłącza do sieci SN

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.



Rysunek 32: Mapa kosztów względnych

```

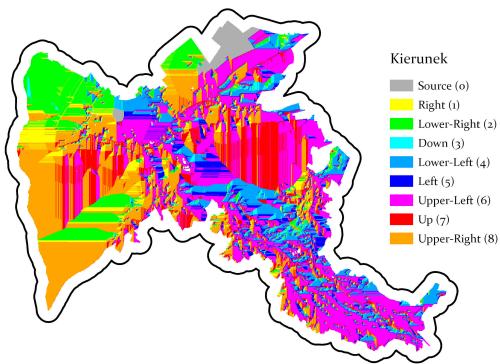
1 pt_merged_layer = "pt_merged_layer"
2 arcpy.management.MakeFeatureLayer(pt_merged, pt_merged_layer)
3 arcpy.management.AddField(pt_merged_layer, "cost", "FLOAT")
4 arcpy.management.CalculateField(
5     in_table=pt_merged_layer,
6     field="cost",
7     expression="costs.get(!x_kod!, 0)",
8     expression_type="PYTHON3",
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,

```

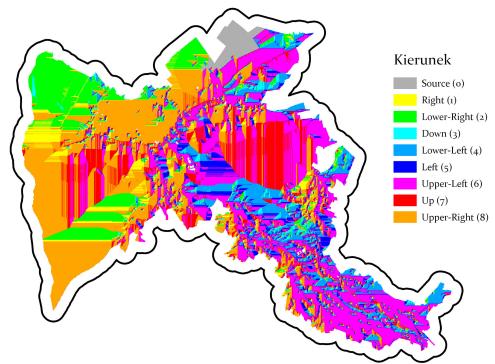
```

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,
    out_polyline_features=f"{geobaza}\\"{wariant}_cost_path_vector")

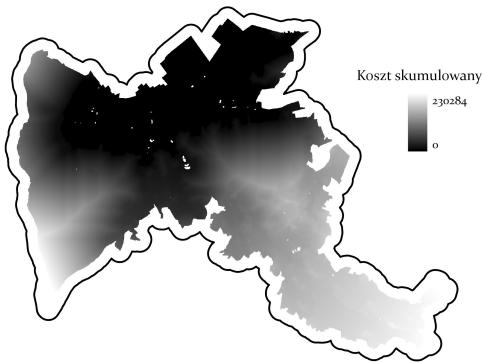
```



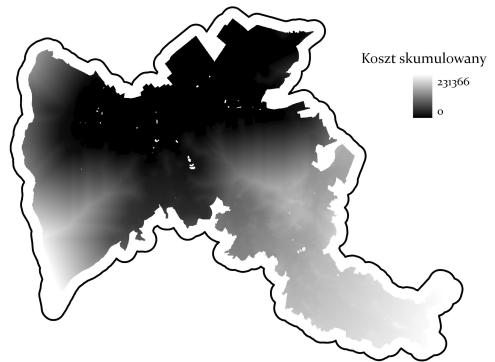
Rysunek 33: Mapa kierunków (backlink) - równe wagi



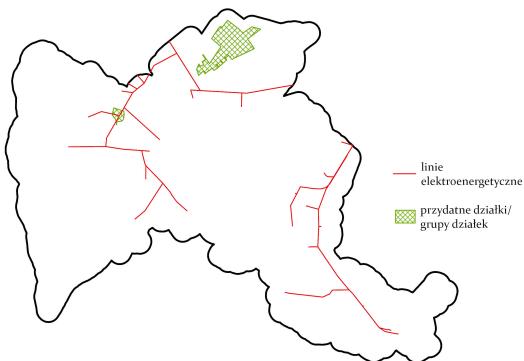
Rysunek 34: Mapa kierunków (backlink) - różne wagi



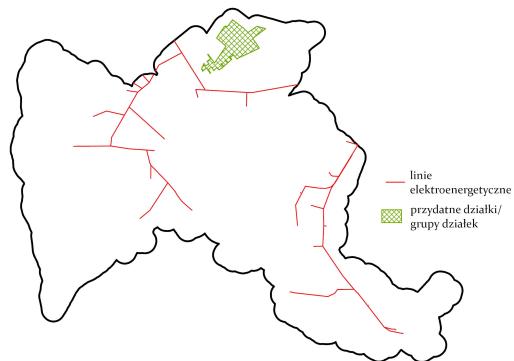
Rysunek 35: Mapa kosztów skumulowanych - równe wagi



Rysunek 36: Mapa kosztów skumulowanych - różne wagi

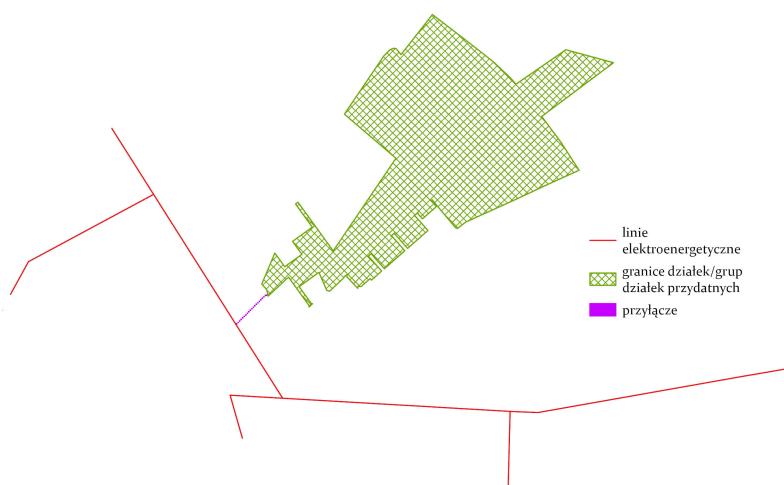


Rysunek 37: Mapa przedstawiająca przydatne działki oraz linie elektroenergetyczne - równe wagi

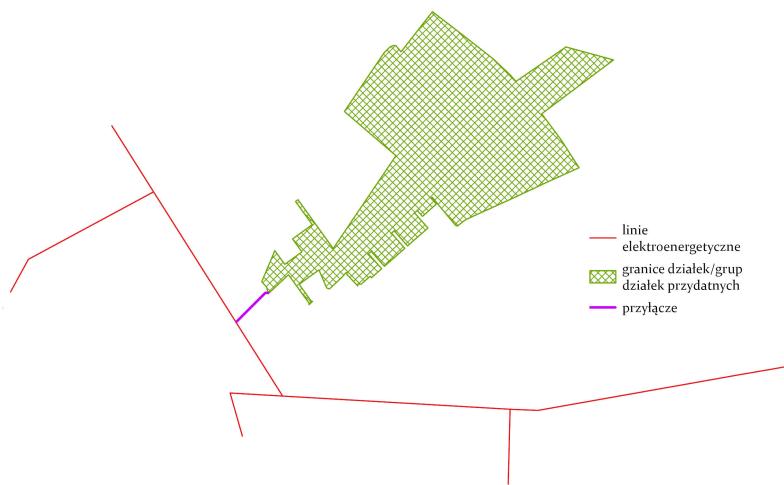


Rysunek 38: Mapa przedstawiająca przydatne działki oraz linie elektroenergetyczne - różne wagi

W przypadku wersji z równymi wagami, linia elektroenergetyczna przechodzi przez jedną z wytypowanych grup działek. W związku z tym żadna ścieżka przyłącza nie została utworzona. Poniżej znajduje się wersja przyłącza utworzona dla wersji z różnymi wagami dla kryteriów.



Rysunek 39: Mapa przedstawiaj±ca utworzon± ścieżk± w postaci rastrowej - ró¿ne wagi

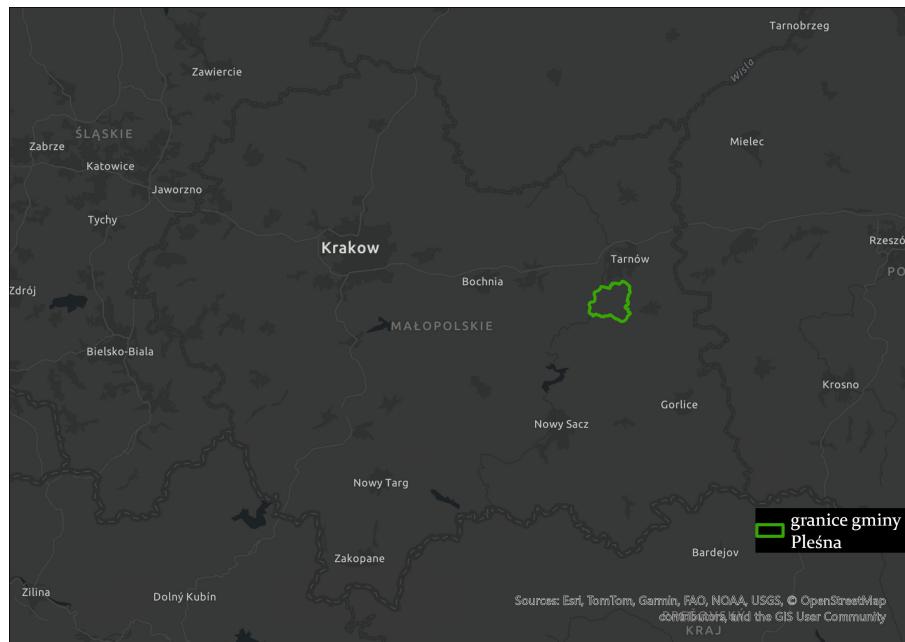


Rysunek 40: Mapa przedstawiaj±ca utworzon± ścieżk± w postaci wektorowej - ró¿ne wagi

## 5 Test modelu na danych z innego obszaru

### 5.1 Opis obszaru

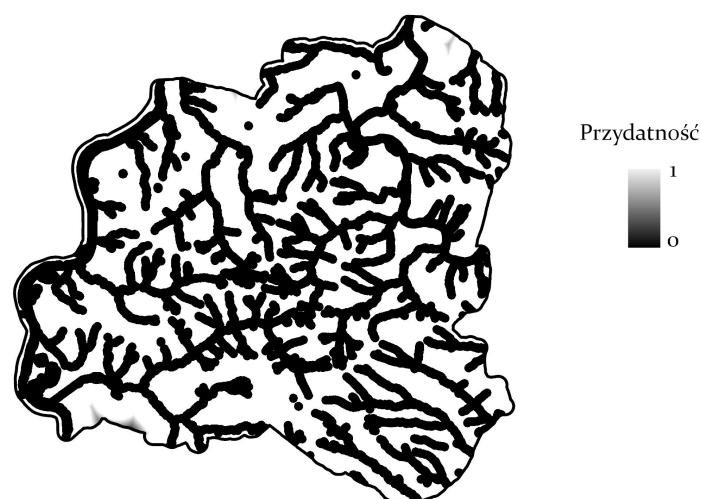
Stworzony model należało przetestować na innym obszarze. Wybrano do tego celu gminę Pleśna - wiejską gminę w powiecie tarnowskim w województwie małopolskim.



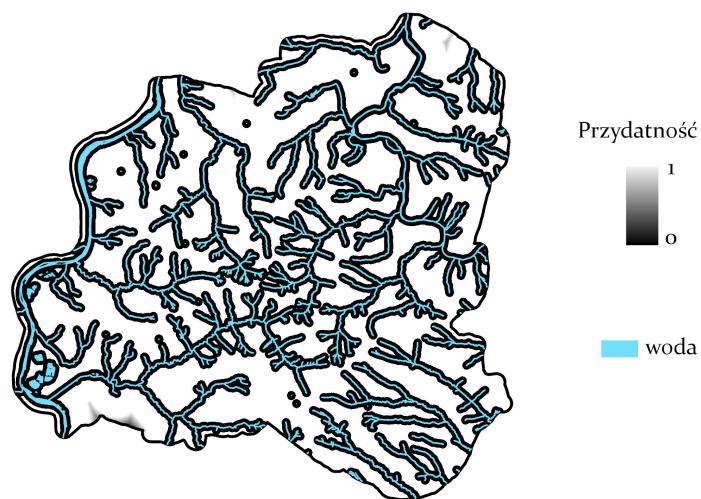
Rysunek 41: Położenie gminy na mapie województwa małopolskiego

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

W gminie występuje duża gęstość rzek i zbiorników wodnych, lecz ze względu na łagodnie postawione granice dla tego kryterium, duża część obszaru gminy cechuje się dużą przydatnością.



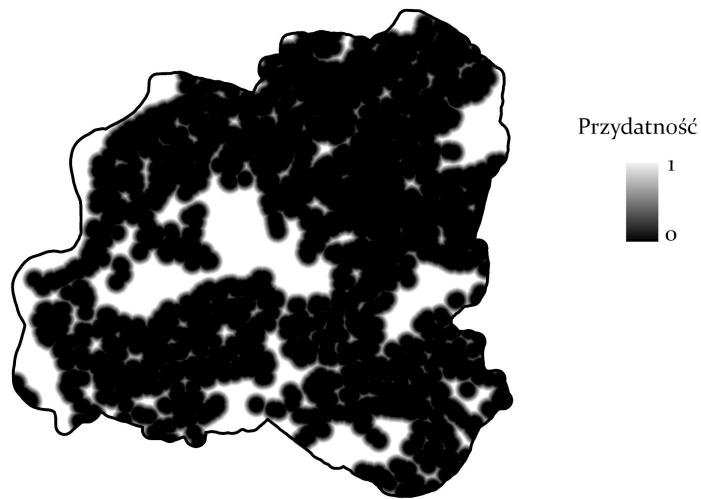
Rysunek 42: Mapa przydatności dla kryterium 1.



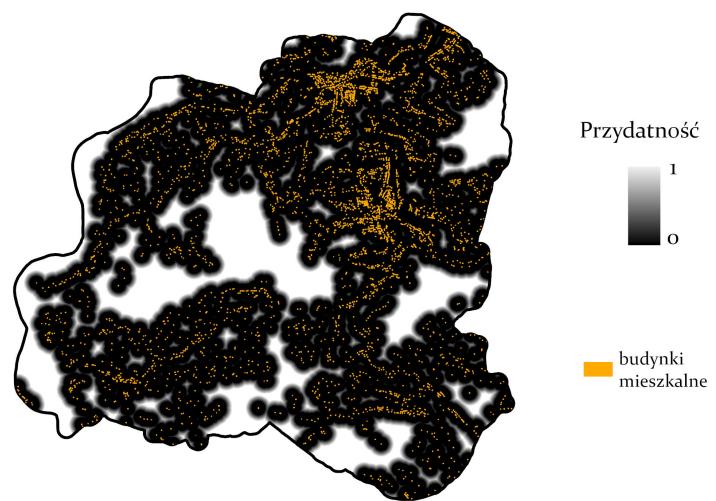
Rysunek 43: Mapa przydatności dla kryterium 1. zawierająca rzeki oraz zbiorniki wodne

### 5.3 Kryterium 2: odległość od budynków mieszkalnych

Jak widać poniżej, w gminie znajduje się wiele budynków mieszkalnych, rozmieszczone mniej więcej równomiernie po całym jej obszarze. Duża część terenu zostaje już na tym etapie wyeliminowana ze względu na obecność kryterium ostrego odległości powyżej 150 metrów od budynków mieszkalnych.



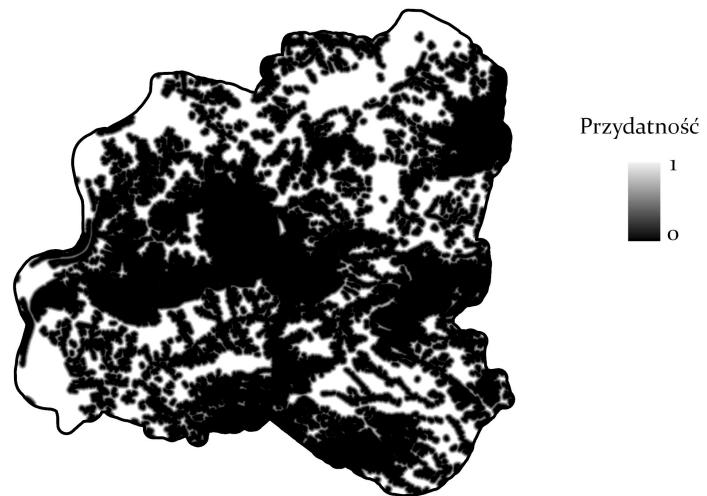
Rysunek 44: Mapa przydatności dla kryterium 2.



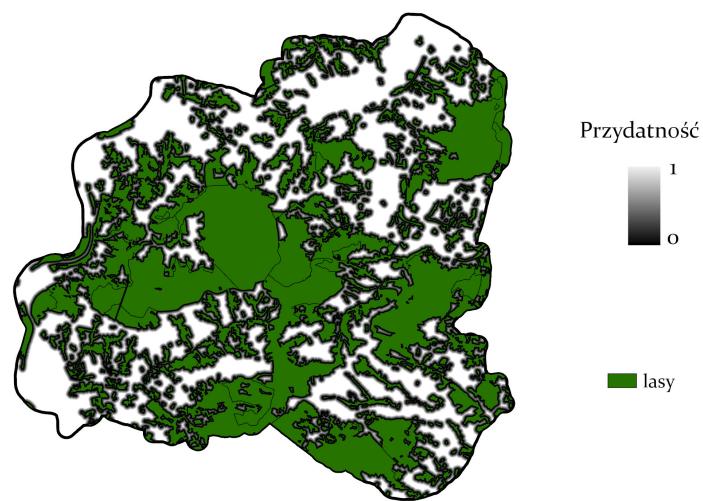
Rysunek 45: Mapa przydatności dla kryterium 2. zawierająca budynki mieszkalne

#### 5.4 Kryterium 3: pokrycie terenu

Duża część obszaru, która dla kryterium 2. przyjęła dużą przydatność, tutaj zostaje całkowicie wyeliminowana ze względu na pojawienie się tam terenów leśnych.



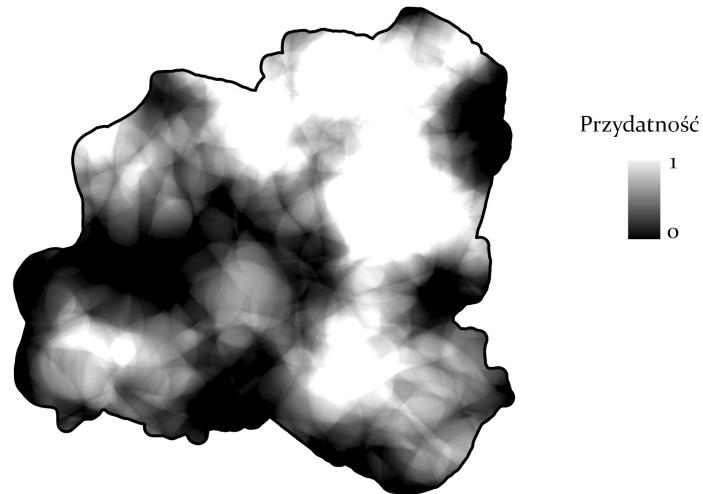
Rysunek 46: Mapa przydatności dla kryterium 3.



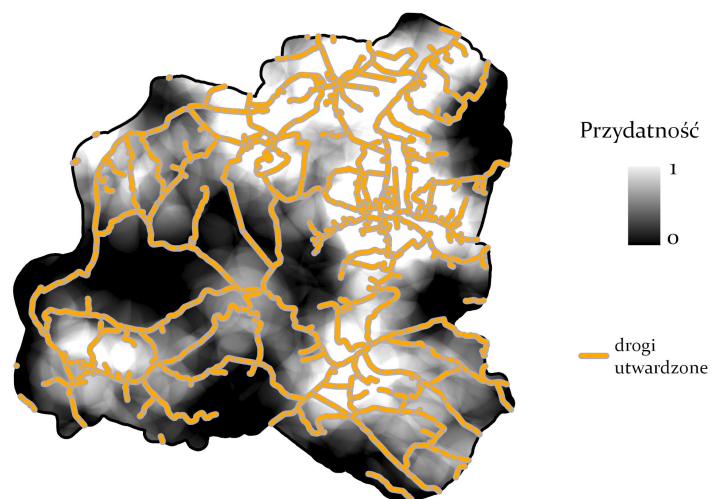
Rysunek 47: Mapa przydatności dla kryterium 3. zawierająca lasy

### 5.5 Kryterium 4: dostęp do dróg utwardzonych

Stosunkowo duża część obszaru, w porównaniu do poprzednich kryteriów, cechuje się dużą przydatnością dla tego kryterium.



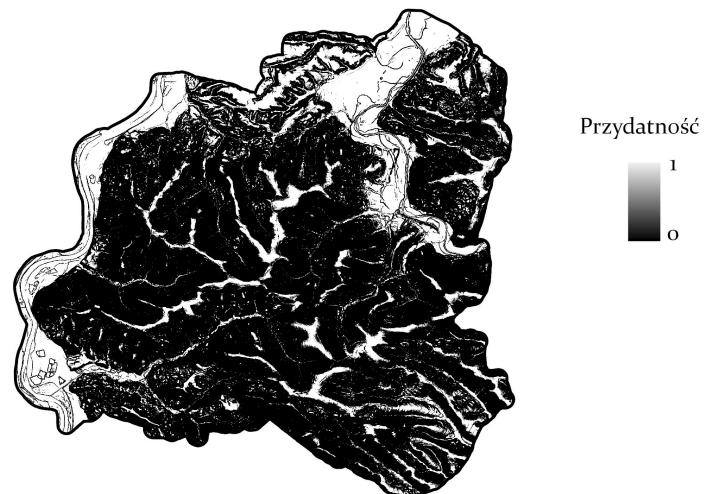
Rysunek 48: Mapa przydatności dla kryterium 4.



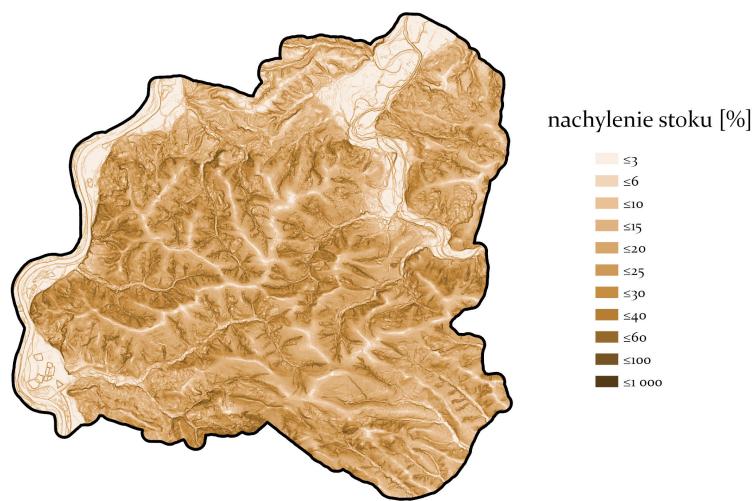
Rysunek 49: Mapa przydatności dla kryterium 4. zawierająca drogi utwardzone

## 5.6 Kryterium 5: nachylenie stoków

Gmina ta również jest gminą o stosunkowo dużych nachyleniach stoków. Lecz istnieją również w niej tereny o niższych stokach, na zachodzie oraz północnym wschodzie.



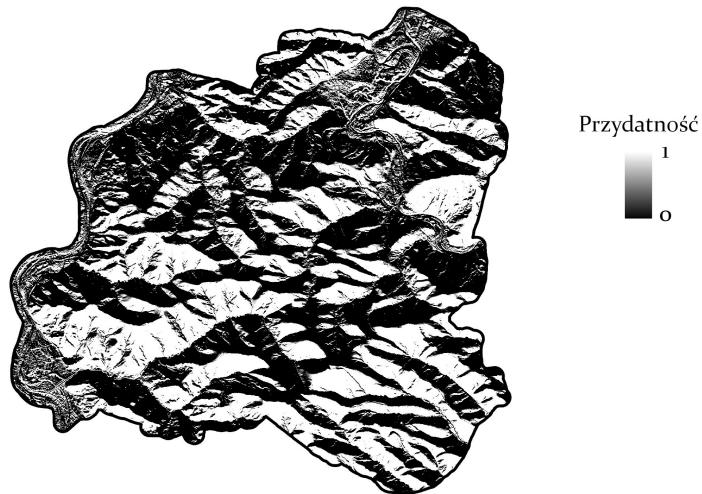
Rysunek 50: Mapa przydatności dla kryterium 5.



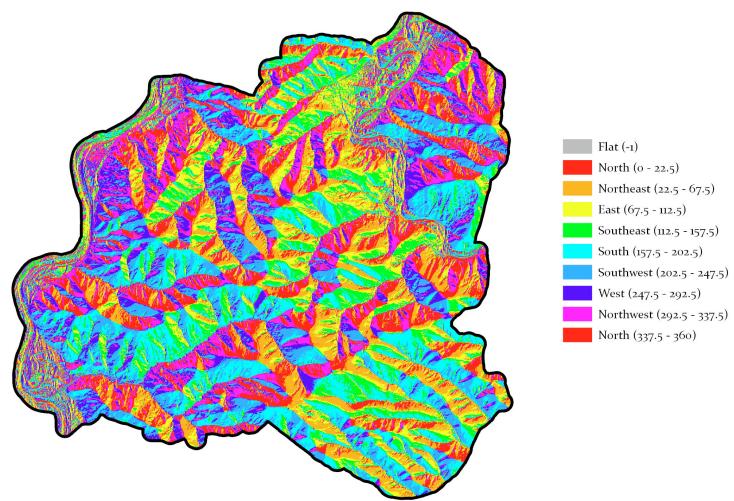
Rysunek 51: Mapa nachyleń stoków wykorzystana podczas sprawdzania kryterium 5.

### 5.7 Kryterium 6: dostęp do światła słonecznego

Pod względem tego kryterium gmina wydaje się w dużej części korzystnie uwarunkowana.

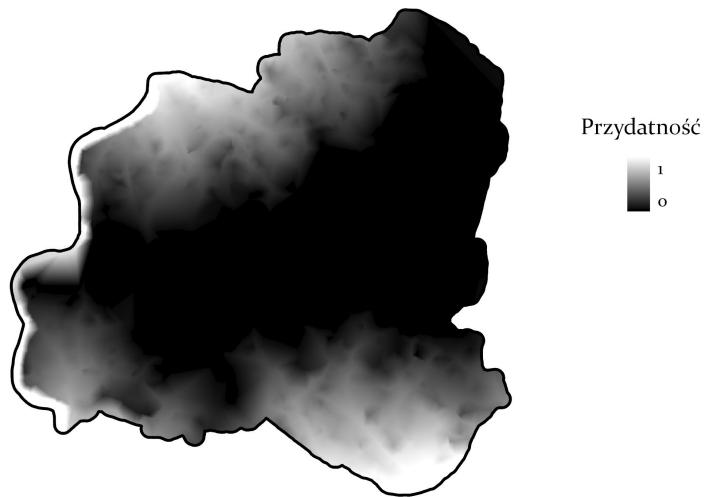


Rysunek 52: Mapa przydatności dla kryterium 6.

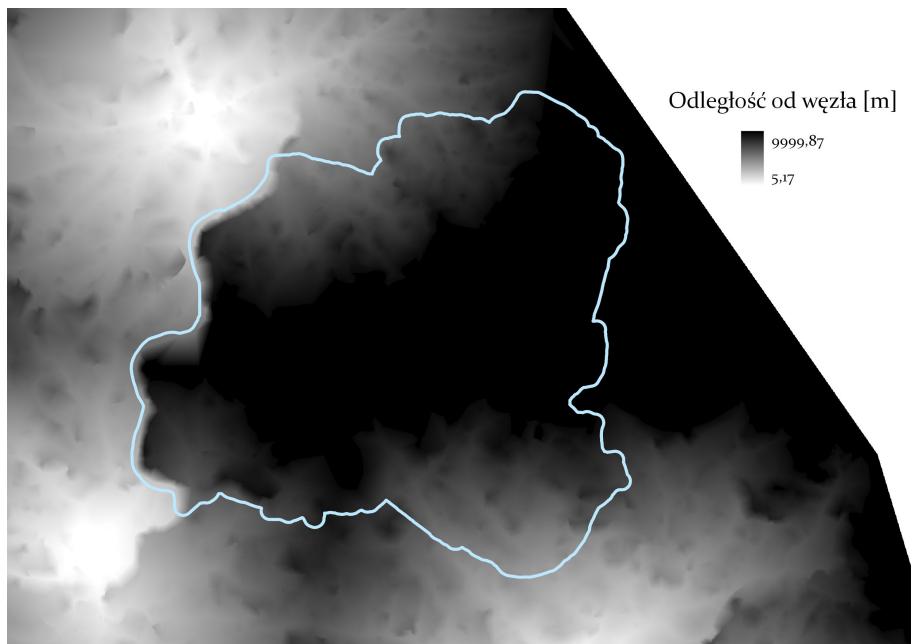


Rysunek 53: Mapa przydatności dla kryterium 6. zawierająca stopień wystawy słonecznej

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

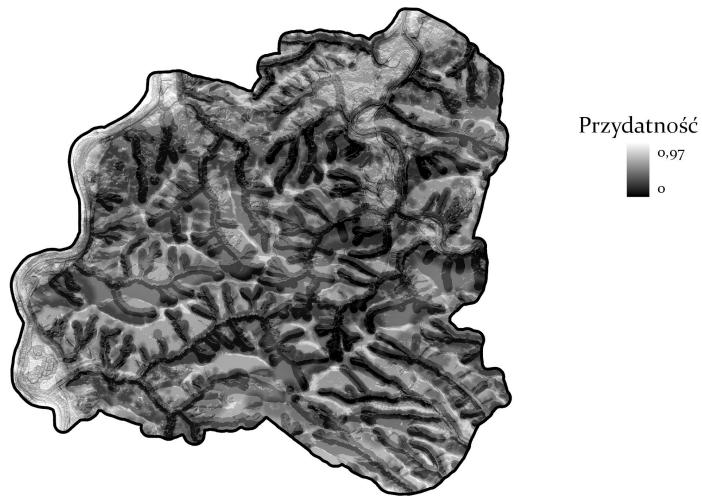


Rysunek 54: Mapa przydatności dla kryterium 7.



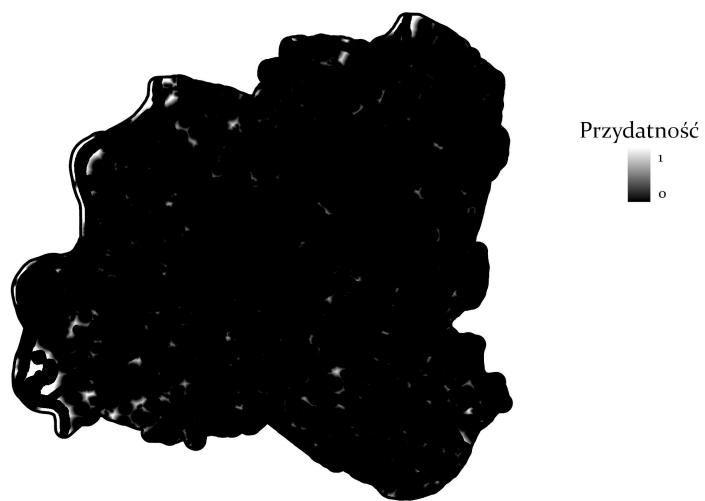
Rysunek 55: Mapa odległości od węzłów

## 5.9 Ocena przydatności terenu

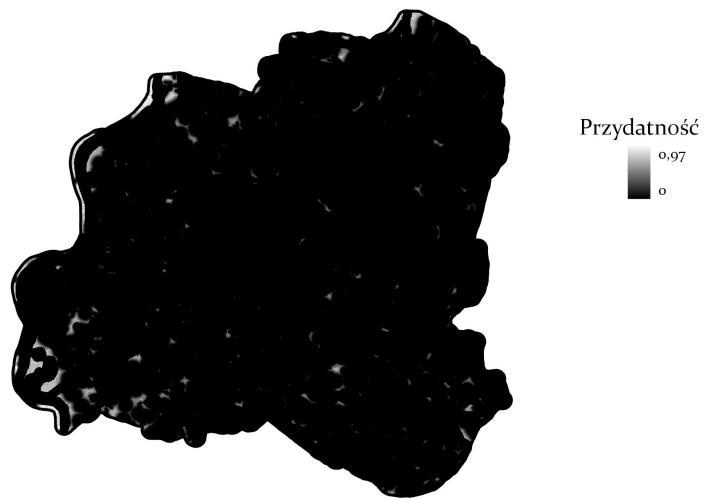


Rysunek 56: Suma kryteriów rozmytych

Ogromna część obszaru gminy jest zupełnie nieprzydatna do inwestycji ze względu na bliskość budynków mieszkalnych, lasów czy wody. Elementy te wzajemnie się uzupełniają i jeżeli na terenie nie ma budynków mieszkalnych, to istnieje tam las, albo przepływa tam rzeka. W wyniku tego niemal cała gmina jest czarną plamą oznaczającą zerową przydatność.

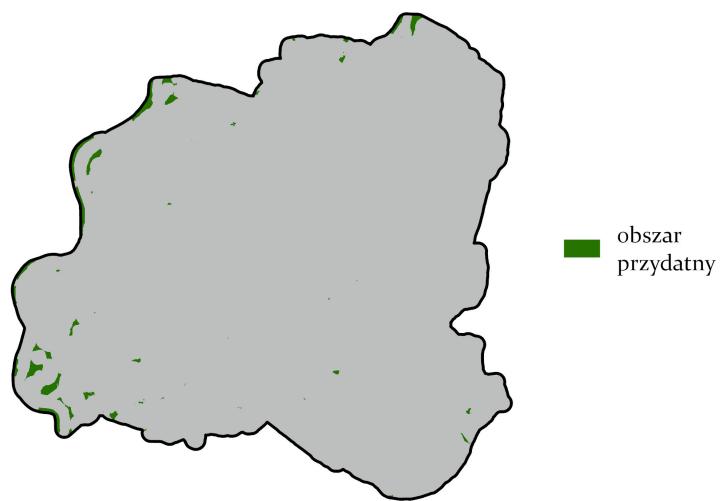


Rysunek 57: Suma kryteriów ostrych



Rysunek 58: Wynik łączenia kryteriów ostrych i rozmytych

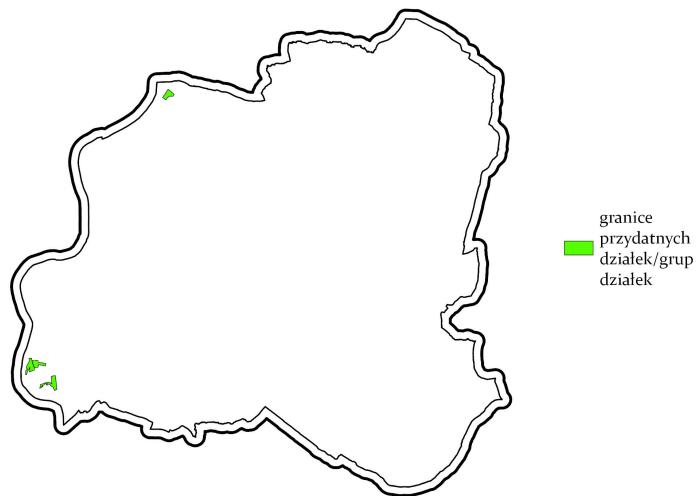
Niewielkie obszary przydatne, jakie pozostały po połączeniu kryteriów, znajdują się głównie w zachodniej części gminy Pleśna.



Rysunek 59: Wynik łączenia kryteriów ostrych i rozmytych po reklasyfikacji

## 5.10 Wybór przydatnych działek

Pomimo nikłej obecności terenów przydatnych pod tę inwestycję w gminie, udało się wyłonić 4 grupy działek (z czego 2 z nich są przecięte jedynie drogą, więc można je połączyć).



Rysunek 60: Mapa przedstawiająca przydatne działki

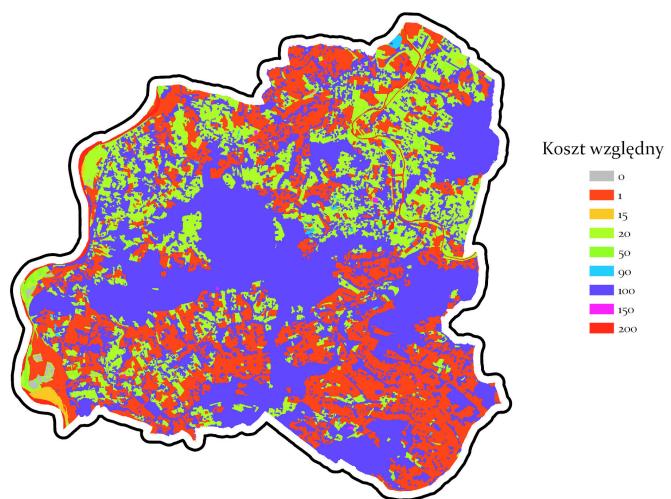


Rysunek 61: Mapa przedstawiająca przydatne działki na ortofotomapie

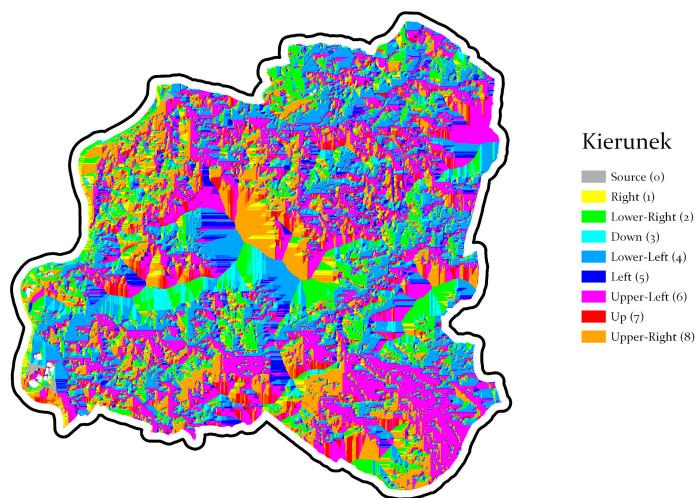


Rysunek 62: Mapa przedstawiająca przydatne działki na ortofotomapie

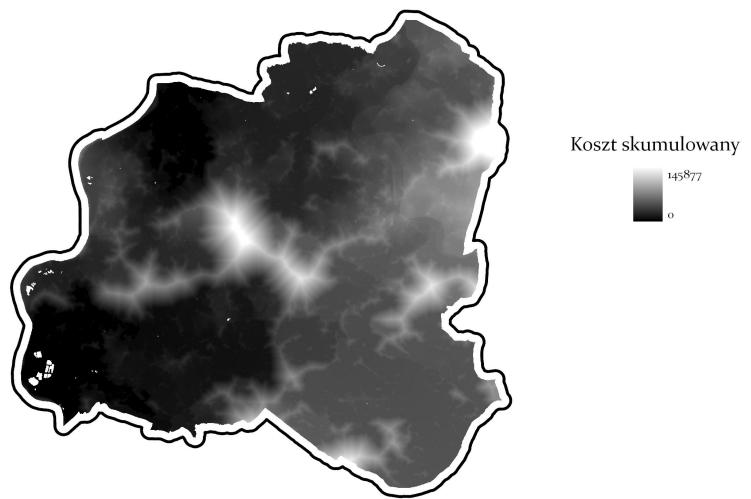
## 5.11 Przyłącze do sieci SN



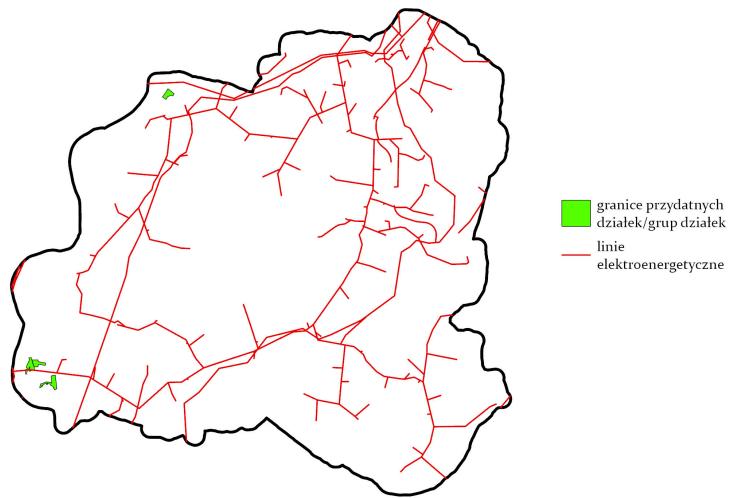
Rysunek 63: Mapa kosztów względnych



Rysunek 64: Mapa kierunków (backlink)

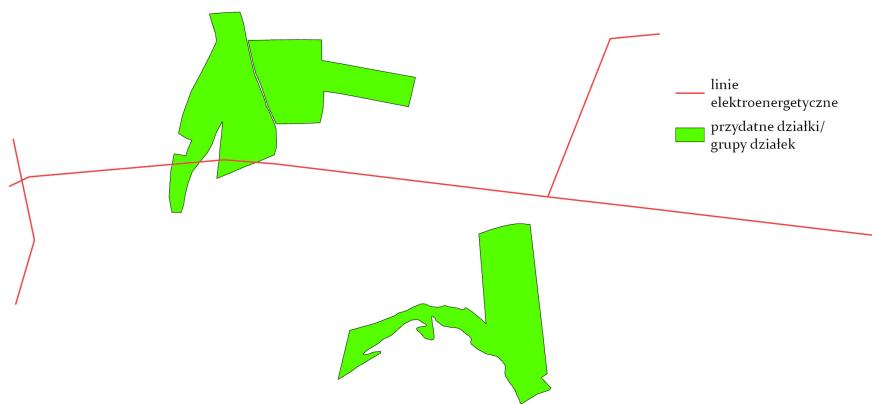


Rysunek 65: Mapa kosztów skumulowanych



Rysunek 66: Mapa przedstawiaj±ca przydatne dzia³ki oraz linie elektroenergetyczne

Przez jedną z wytypowanych grup dzia³ek przechodzi ju¿ linia elektroenergetyczna, w zwi±zku z czym scie¿ka przy³±cza nie zosta³a utworzona.



Rysunek 67: Mapa przedstawiająca linie elektroenergetyczne przechodzące przez jedną z grup działek