

Vlaanderen
is omgeving

Verstedelijkt, randstedelijk en landelijk Vlaanderen

Indeling op basis van statistische sectoren

DEPARTEMENT
OMGEVING

omgevingvlaanderen.be

In dit document wordt beschreven hoe de statistische sectoren via een generiek algoritme ingedeeld worden in drie types: verstedelijkt – randstedelijk - landelijk

COLOFON

Verantwoordelijke uitgever:

Departement Omgeving
Vlaams Planbureau voor Omgeving

Koning Albert II-laan 20 bus 8
1000 Brussel

vpo.omgeving@vlaanderen.be

www.omgevingvlaanderen.be

Bronverwijzing: Vermeiren Karolien, Pisman Ann, Vanacker Stijn, Willems Peter, Loris Isabelle, en Engelen Guy (2018), *Verstedelijkt, randstedelijk en landelijk Vlaanderen. Indeling op basis van statistische sectoren*, studie uitgevoerd in opdracht van het Vlaams Planbureau voor Omgeving.

PARTNERS



VITO Rapport nummer 2017/RMA/R/1377

Inhoud

Algoritme stedelijke gebieden 5
Doelstelling 5
Inleiding 5
Stap 0 – Voorbereiding 6
Stap 1 – Ruimtelijke variabele 1: ruimtebeslag 6
Stap 2 – Ruimtelijke variabele 2 en 3: bevolkings- of tewerkstellingsdichtheid 7
Stap 3 – Gaten in 'verstedelijkt' opvullen 8
Stap 4 – Ruimtelijke variabele 4: voldoende aantal inwoners per cluster 9
Stap 5 – Randstedelijke sectoren identificeren 9
Stap 6 – Uitbreiden van aantal randstedelijke sectoren 10
Stap 7 – Gaten in randstedelijk opvullen 10
Stap 8 – 'Eilanden' in verstedelijkt gebied 11
Bijlage: metadata 13
Abstract 13
Verklaring van de inhoud van de kolommen van de polygonenlaag en de spreadsheet 14
Lijst kaartbestanden 16
Verwerkingsalgoritme "stedelijkegebieden.py" 17

Algoritme stedelijke gebieden

Doelstelling

Het doel van deze opdracht bestaat erin om een indeling van het Vlaamse grondgebied te maken in ‘verstedelijkte’, ‘randstedelijke’ en ‘landelijke’ gebieden op het detailniveau van de statistische sectoren. De indeling gebeurt volautomatisch aan de hand van een speciaal ervoor ontwikkeld algoritme. Dit algoritme is beschikbaar als een script dat ook met andere grenswaarden gerund kan worden. In principe zijn ook de ruimtelijke variabelen aanpasbaar op voorwaarde dat ze redelijk gelijkaardig zijn aan de 4 variabelen die momenteel toegepast worden.

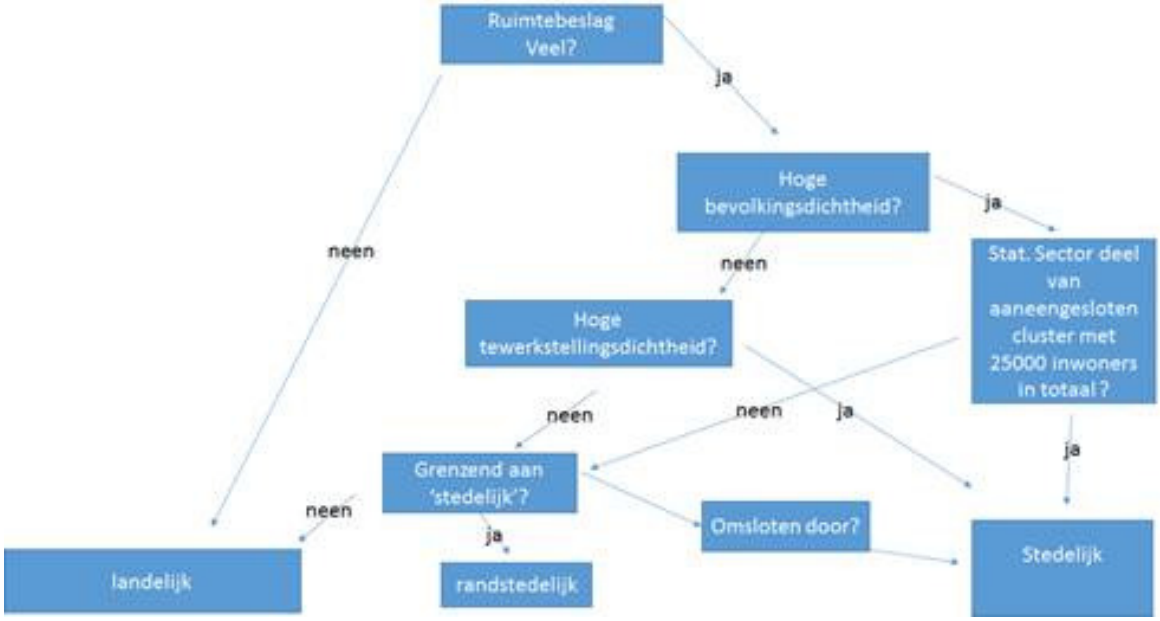
Deze opdeling op basis van statistische sectoren heeft niet tot doel om direct toepasbaar te zijn voor het ruimtelijk beleid, maar eerder om een ruimtelijke-typologisch kader te bieden om analyses over het Vlaamse grondgebied uit te voeren.

Inleiding

Met het algoritme ‘stedelijke gebieden’ worden alle statistische sectoren in Vlaanderen ingedeeld in drie types, namelijk ‘verstedelijkt’, ‘randstedelijk’ of ‘landelijk’. Deze indeling gebeurt op basis van een aantal ruimtelijke variabelen en het al dan niet aanliggend zijn van de sectoren. In dit document wordt stap per stap beschreven hoe het algoritme rekenkundig omgaat met de ruimtelijke variabelen en andere randvoorwaarden.

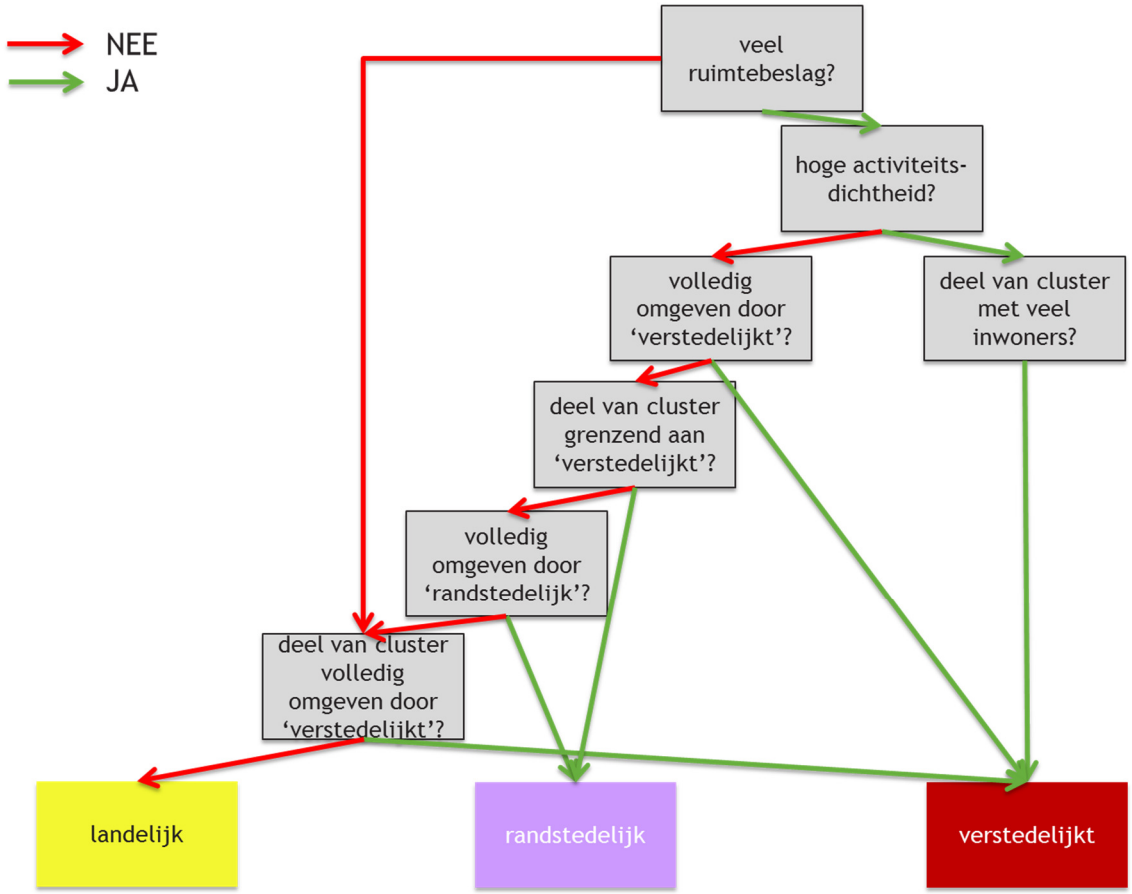
De gehanteerde ruimtelijke variabelen zijn ruimtebeslag, tewerkstellingsdichtheid, inwonersdichtheid en totaal aantal inwoners per cluster van sectoren. Per ruimtelijke variabele wordt een drempelwaarde ingesteld gebaseerd op de statistische verdeling van de waarden gemeten in alle statistische sectoren van Vlaanderen. De keuze voor de ene of andere mogelijke drempelwaarde is op een iteratieve wijze tot stand gekomen. Varianten werden daarvoor doorgerekend en met de betrokken experten van het Vlaams Planbureau voor Omgeving geëvalueerd op hun capaciteit om gekende voorbeelden van stedelijke, randstedelijke en landelijke gebieden op een voldoende discriminerende wijze in kaart te brengen.

Onderstaand schema (zie Figuur 1) omvat de originele beslisboom die de leidraad van het algoritme vormt.



Figuur 1: oorspronkelijke beslisboom die het algoritme implementeert

De beslisboom werd gaande het traject lichtelijk bijgesteld en resulteerde finaal in het onderstaande schema (zie Figuur 2).



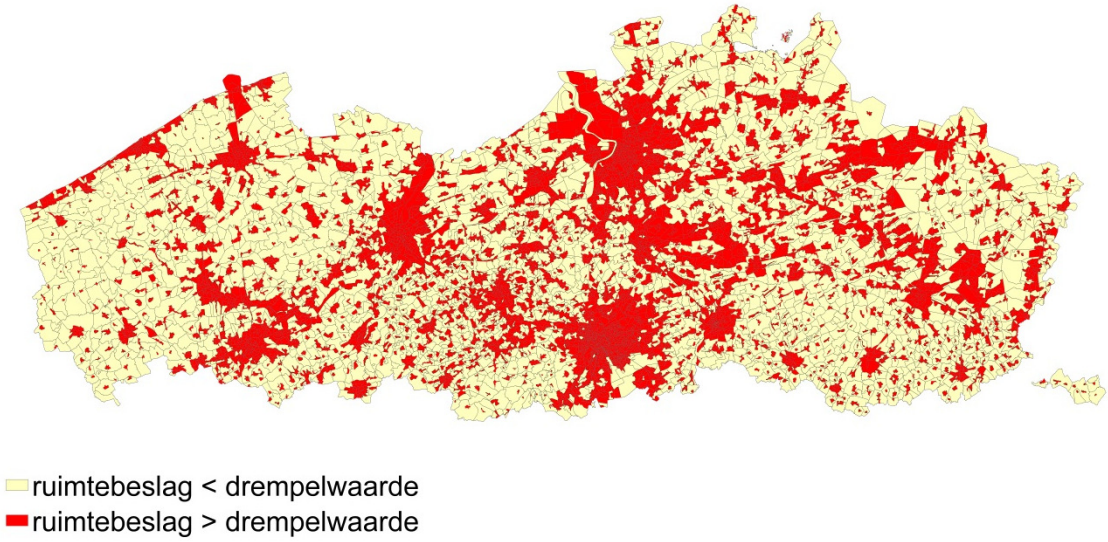
Figuur 2: uiteindelijke beslisboom die het algoritme implementeert

Stap 0 – Voorbereiding

Vooraleer aan de indeling te beginnen, wordt voor elke statistische sector de waarde van alle ruimtelijke variabelen berekend: het ruimtebeslag, aantal tewerkgestelden en het aantal inwoners. Deze berekening gebeurt met behulp van een GIS op basis van het landgebruiksbestand en de afgeleide indicatoren op 10m resolutie (2013). Dit resultaat vormt de input van het algoritme, zijnde een polygonenkaart van alle statistische sectoren in Vlaanderen met hun ruimtebeslag (uitgedrukt in percentage ten opzichte van de oppervlakte), tewerkstellingsdichtheid, totaal aantal inwoners en inwonersdichtheid. Deze gegevens zijn beschikbaar en opgeleverd in de MS Exceltabel 'versted_rand_land.xlsx'. De verklaring en inhoud van de verschillende kolommen is opgenomen in Bijlage 1.

Stap 1 – Ruimtelijke variabele 1: ruimtebeslag

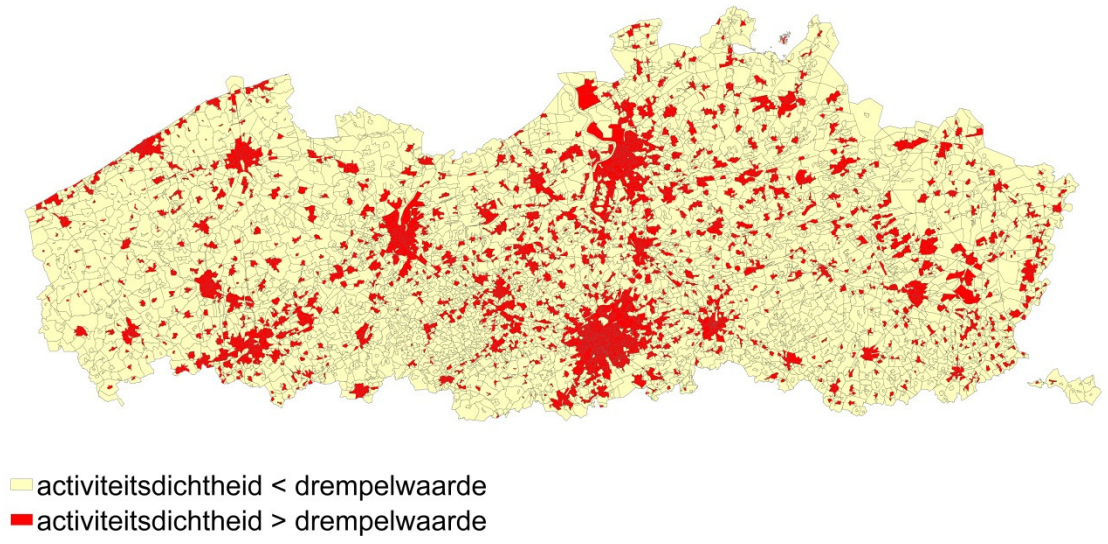
De eerste ruimtelijke variabele waarop elke statistische sector getest wordt in het algoritme is het percentage ruimtebeslag. **De drempelwaarde voor ruimtebeslag is 32.5%, zijnde het gemiddelde ruimtebeslag in Vlaanderen** (nulmeting, berekend op basis van 10m resolutie). Alle sectoren met een hoger percentage ruimtebeslag, komen in aanmerking om als 'verstedelijkt' of 'randstedelijk' ingedeeld te worden. Alle sectoren met een lager ruimtebeslag worden 'landelijk' (zie: Figuur 3).



Figuur 3: statistische sectoren in Vlaanderen volgens ruimtebeslag (ifv drempelwaarde)

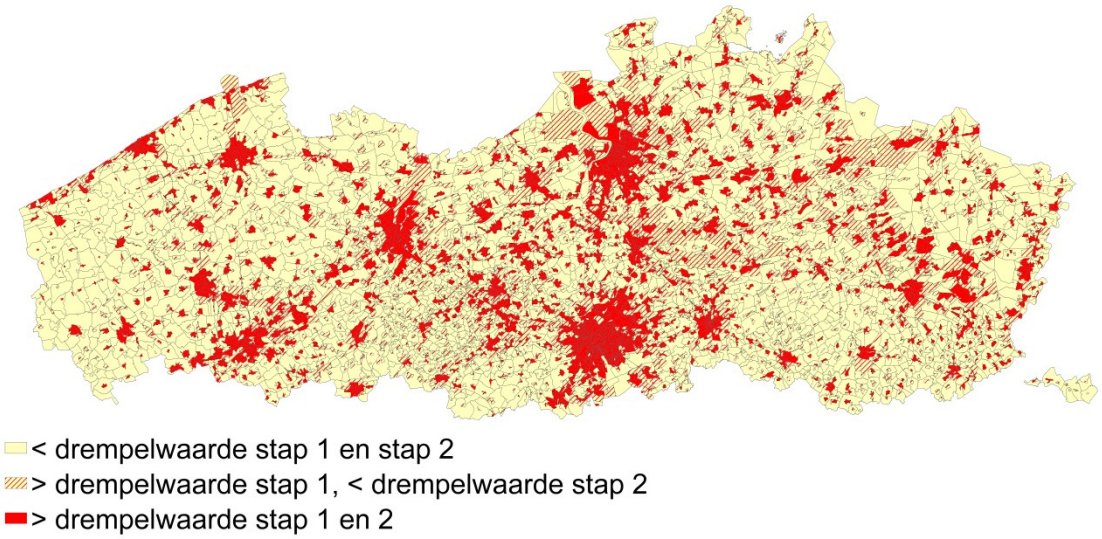
Stap 2 – Ruimtelijke variabele 2 en 3: bevolkings- of tewerkstellingsdichtheid

In de tweede stap worden ruimtelijke variabele 2 en 3 beschouwd, respectievelijk de inwonersdichtheid en de tewerkstellingsdichtheid. **De drempelwaarde voor ruimtelijke variabele 2 en 3 is het gemiddelde plus 0,5 keer de standaarddeviatie** (berekend op basis van 10m resolutie). Dit komt neer op **38% betrouwbaarheidsinterval, dus 38% van de cellen liggen boven deze drempelwaarde of ongeveer 2 op 7)** (berekend op basis van 10m resolutie). Voor de bevolkingsdichtheid is de grenswaarde **11,85 inwoners per ha** en voor de tewerkstellingsdichtheid is de grenswaarde **10,14 werkplekken per ha**. Indien statistische sectoren hoger dan de drempelwaarde scoren voor bevolkingsdichtheid ofwel tewerkstellingsdichtheid, spreken we van sectoren met een hoge activiteitsdichtheid. We selecteren hiermee dus sectoren die zich duidelijk manifesteren als plekken met een hoge dichtheid aan wonen of werken (Figuur 4).



Figuur 4: statistische sectoren in Vlaanderen volgens activiteiten (ifv drempelwaarde)

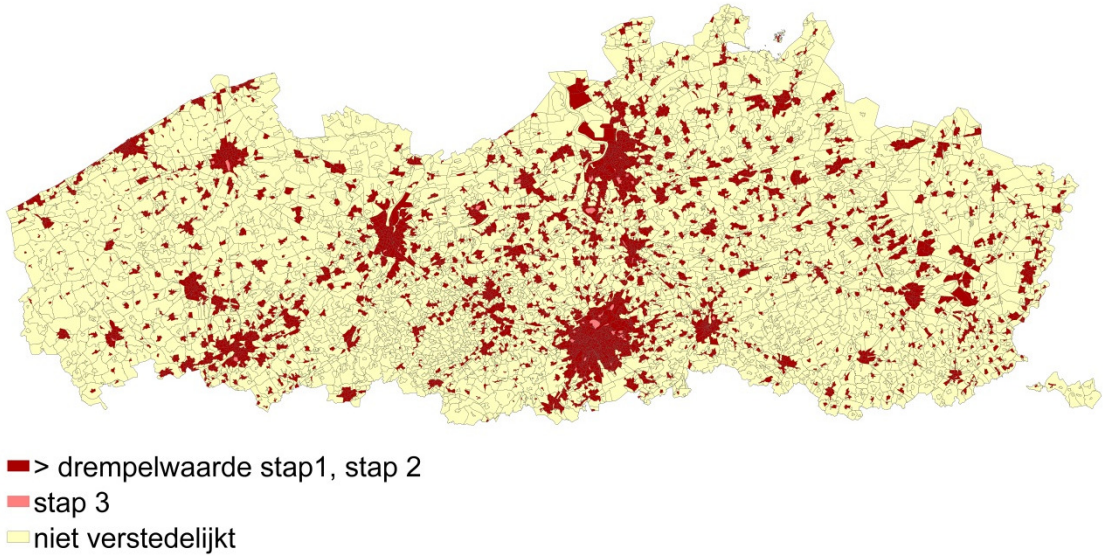
Sectoren die boven de drempelwaarde scoren zowel voor stap 1 als stap 2, komen in aanmerking voor de categorie ‘verstedelijkt’ (Figuur 5).



Figuur 5: statistische sectoren in Vlaanderen volgens ruimtebeslag en activiteiten (ifv drempelwaarden)

Stap 3 – Gaten in ‘verstedelijkt’ opvullen

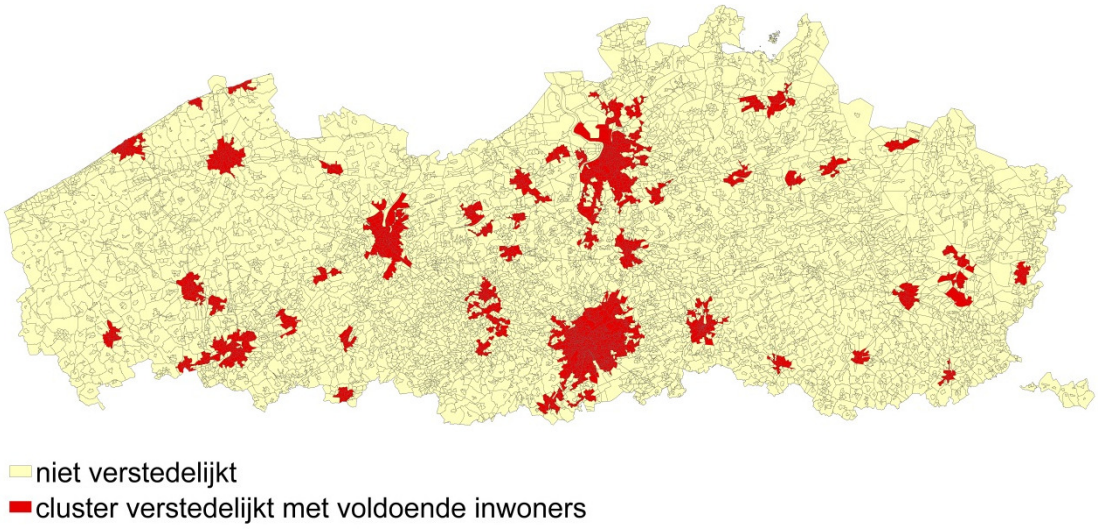
Statistische sectoren die niet als stedelijk weerhouden zijn, maar volledig omgeven zijn door verstedelijkte sectoren (= al hun buren zijn wél verstedelijkt), worden in stap 3 opgevuld en komen mee in aanmerking voor de categorie ‘verstedelijkt’. Ze verschijnen in het roze op de kaart van Figuur 6.



Figuur 6: toevoeging gaten in ‘verstedelijkt’ na stap 2

Stap 4 – Ruimtelijke variabele 4: voldoende aantal inwoners per cluster

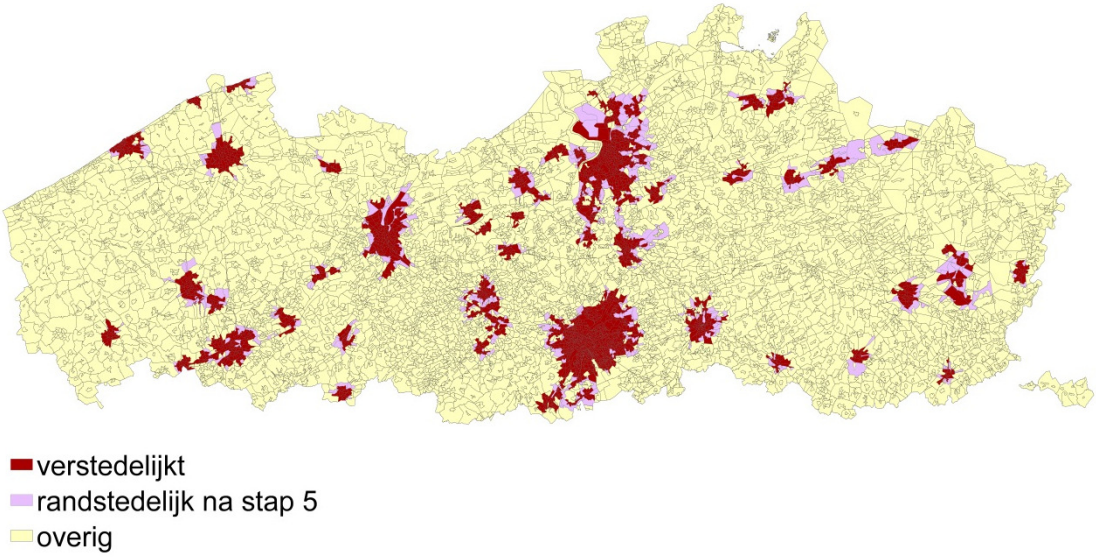
In stap 4 worden alle aan elkaar grenzende statistische sectoren die na stap 3 in aanmerking komen voor de categorie 'verstedelijkt' geclusterd. Per cluster wordt het totaal aantal inwoners berekend. Een cluster wordt finaal als 'verstedelijkt' weerhouden wanneer zijn inwoneraantal boven de **drempelwaarde van 15.000 inwoners** ligt (zie Figuur 7). In de laatste stap kunnen eventuele 'eilanden', zijnde meerdere sectoren die niet 'verstedelijkt' of 'randstedelijk' zijn maar wel volledig omgeven zijn door 'verstedelijkte sectoren', nog opgevuld worden tot 'verstedelijkt'.



Figuur 7: 'verstedelijkt' als zijnde clusters van statistische sectoren met voldoende inwoners (> drempelwaarde van 15.000)

Stap 5 – Randstedelijke sectoren identificeren

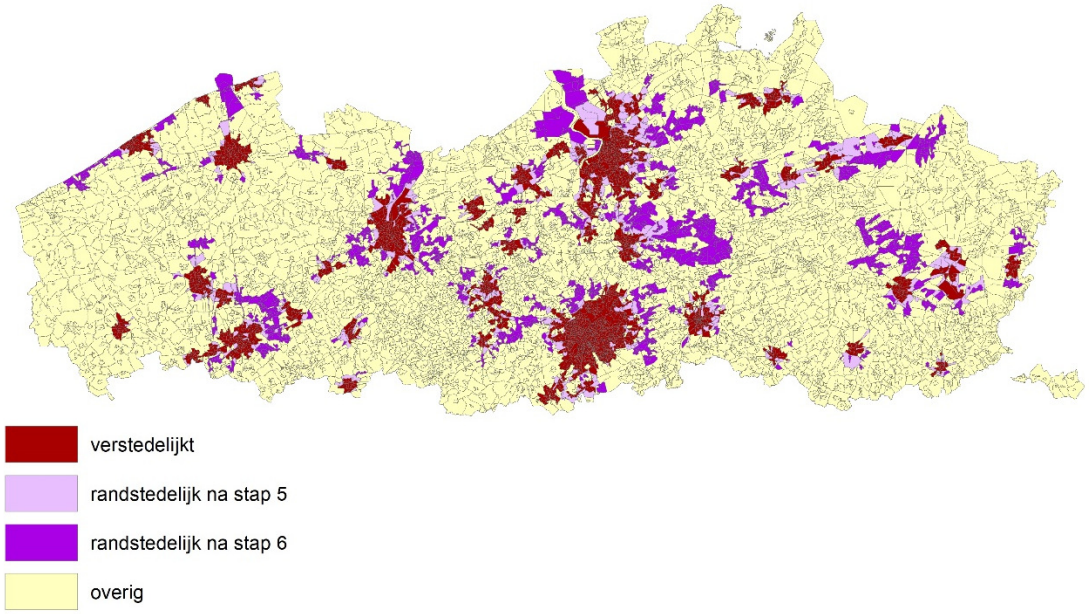
Om vervolgens 'randstedelijke' sectoren te definiëren worden de sectoren weerhouden die grenzen aan stedelijke sectoren én voldoende ruimtebeslag hebben, zijnde meer dan de grenswaarde 32,5%.



Figuur 8: selectie van 'randstedelijke' sectoren

Stap 6 – Uitbreiden van aantal randstedelijke sectoren

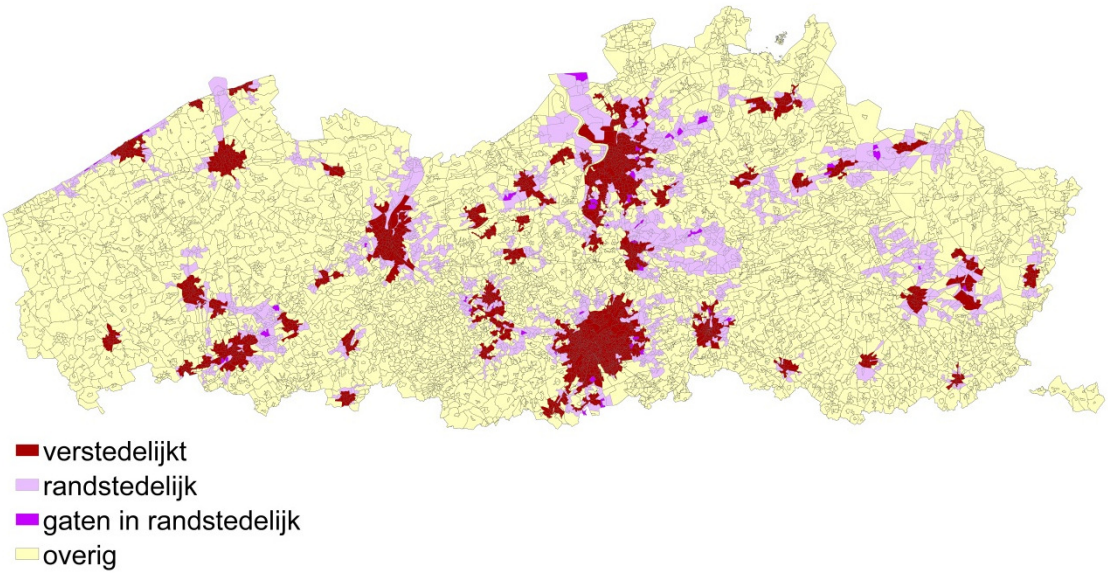
Naast de 'randstedelijke sectoren van de eerste graad' uit stap 5, worden in stap 6 de sectoren toegevoegd die deel uitmaken van clusters van aaneengesloten sectoren met voldoende ruimtebeslag en grenzen aan de randstedelijke sectoren van stap 5. Ze zijn te zien in lila in Figuur 9.



Figuur 9: uitbreiding 'randstedelijke' sectoren

Stap 7 – Gaten in randstedelijk opvullen

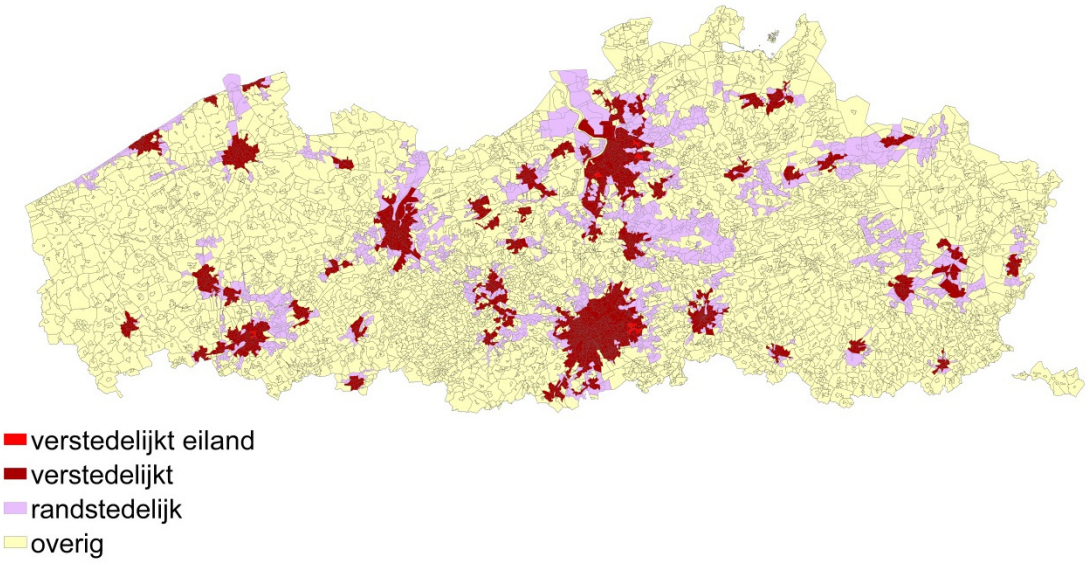
Analoog aan stap 3 worden in stap 7 gaten opgevuld. In stap 7 gaat het om gaten in het randstedelijke gebied. Indien alle buren van een statistische sector randstedelijk zijn, wordt de sector eveneens randstedelijk, zelfs als het ruimtebeslag beneden de grenswaarde van 32,5% ligt. Ze zijn te zien in lila in Figuur 10.



Figuur 10: gaten 'randstedelijk' vullen

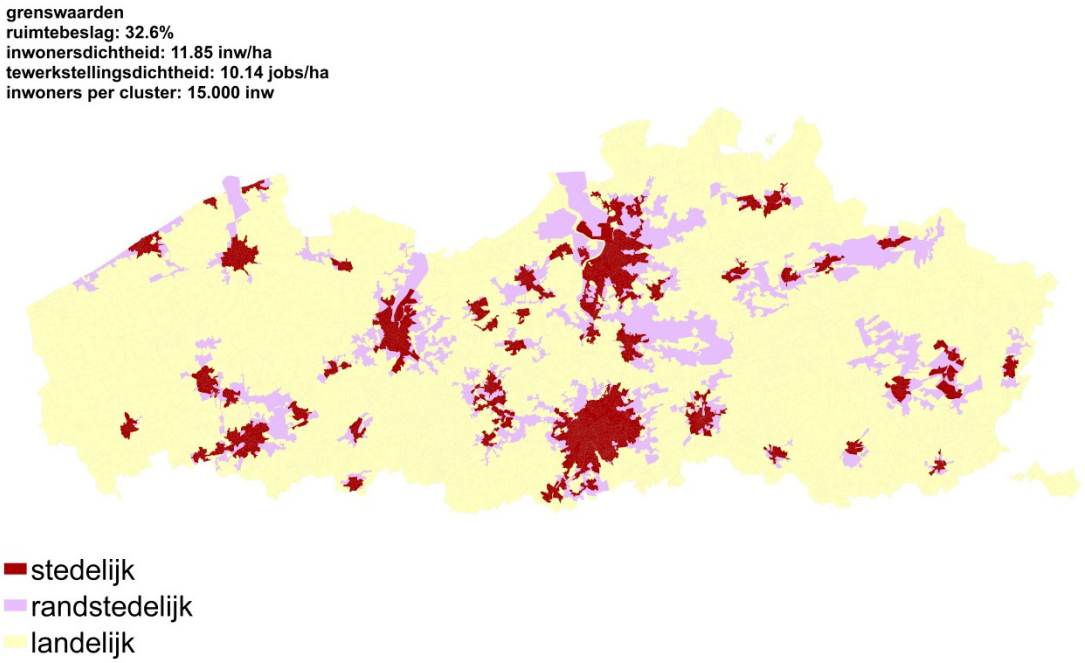
Stap 8 – ‘Eilanden’ in verstedelijkt gebied

Als achtste en laatste stap worden alle sectoren geclusterd per categorie zoals ze zijn toegewezen tot en met in stap 7 (‘verstedelijkt’, ‘randstedelijk’ of ‘overig’). Per cluster worden de buren geanalyseerd. Indien een cluster omgeven is door enkel ‘verstedelijkte sectoren’, is dit een ‘eiland’ binnen verstedelijkt gebied en wordt de cluster aan het verstedelijkt gebied toegevoegd. De betreffende eilanden zijn in fel rood weergegeven in Figuur 11.



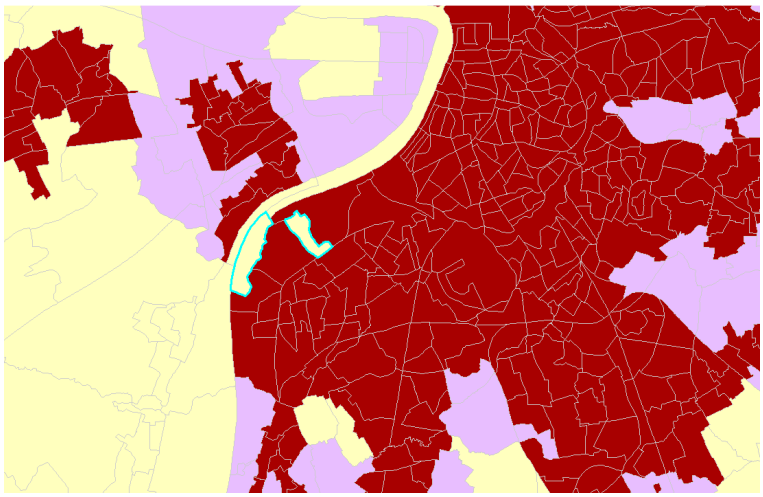
Figuur 11: ‘eilanden’ opvullen.

In het finale resultaat worden alle ‘overige’ sectoren gecategoriseerd als ‘landelijke sectoren’ (Figuur 12).



Figuur 12: finale indeling statistische sectoren als ‘verstedelijk’, ‘randstedelijk’ of ‘landelijk’

Opmerking: statistische sectoren kunnen grillige vormen aannemen. Visueel kan het op de kaart lijken dat een sector nog als 'gat' overblijft hoewel het omringd is door 'verstedelijkt'gebied (en bijgevolg volgens het stappenplan opgevuld had moeten worden). Maar sommige sectoren kunnen bestaan uit verschillende niet verbonden onderdelen (zie Figuur 13) waardoor meerdere polygonen behoren tot dezelfde statistische sector. In dit geval worden de buren van de onderdelen samen geanalyseerd wat ertoe kan leiden dat de sector niet per definitie ingevuld wordt.



Figuur 13: voorbeeld van een sector bestaande uit twee losstaande onderdelen

Bijlage: metadata

Abstract

De polygonenlaag ‘versted_rand_land_vlaa.shp’ bevat alle statistische sectoren van het Vlaamse Gewest inclusief de kolom ‘type’ die per sector een ruimtelijk type verstedelijkt, randstedelijk of landelijk toekent.

De polygonenlaag ‘versted_rand_land_vlabru.shp’ bevat alle statistische sectoren van het Vlaamse Gewest én van het Brusselse Hoofdstedelijk Gewest, inclusief de kolom ‘type’ die per sector een ruimtelijk type verstedelijkt, randstedelijk of landelijk toekent.

Verklaring van de inhoud van de kolommen van de polygonenlaag en de spreadsheet

CODSEC	NIS-code sector
NISCODE	NIS-code gemeente
SEC	code sector
OPPERVL	oppervlakte in m²
ID_statsec	Unieke id
inwoners	aantal inwoners
tewerkstel	aantal tewerkgestelden
RBpct	percentage ruimtebeslag
label	naam sector
inwdicht	inwonersdichtheid
werkdicht	tewerkstellingsdichtheid
type_code	indeling in cijfercode 0/1/2
type	indeling in type verstedelijkt/randstedelijk/landelijk
opp_ha	oppervlakte in ha

Lijst kaartbestanden

Input	Inwonersdichtheid 2013	Bron: Poelmans, L., Van Esch, L., Janssen, L., Engelen, G.(2016), Indicatoren ruimtelijk rendement, referentiejaar 2013.
	Tewerkstellingsdichtheid 2013	
	Ruimtebeslag 2013	
	Statistische sectoren met statistieken	Bron: Algemene Directie Statistiek - Statistics Belgium
Output	Indeling statistische sectoren op kaart (Vlaamse Gewest)	versted_rand_land_vlaa.shp
	Indeling statistische sectoren op kaart (Vlaams en Brussels Gewest)	versted_rand_land_vlabru.shp
	Indeling statistische sectoren in lijst	versted_rand_land.xlsx
Algoritme	Verwerking	stedelijkegebieden.py

Verwerkingsalgoritme “stedelijkegebieden.py”

```
# -*- coding: cp1252 -*-
# ---------------------------------------------------------------------------
# algoritme om statistische sectoren op basis van
# ruimtebeslag, inwoners en tewerkstelling
# in te delen in verstedelijkt, randstedelijk en landelijk
# ---------------------------------------------------------------------------

# Import arcpy module
import arcpy

# Variabelen
# waardes = variabele voor scenarionummer, inwonersdichtheid, tewerkstellingsdichtheid en aantal inwoners (per cluster)
waardes = [(20,11.85,10.14,15000)]

for waarde in waardes:
   ruimtebeslag = str(0.326)
   inwonersdichtheid = str(waarde[1])
   tewerkstellingsdichtheid = str(waarde[2])
   aantal_inw = str(waarde[3])
   resultnum = str(waarde[0])
   homef = "D:\\RV\\stedelijke_gebieden\\"

   #originele databank met statistische sectoren
   statistische_sectoren = homef + "statistische_sectoren_VL_BR_stats.shp"
   #temp bestanden
   statistische_sectoren_Polygo7 = homef+"buren1.dbf"
   statistische_sectoren_CopyFe2 = homef+"summary1.dbf"
   statistische_sectoren_CopyFe = homef+"statistische_sectoren_copy.shp"
   dissolve_shp = homef+"dissolve.shp"
   dissolve_shp2 = homef+"dissolve2.shp"
   dissolve_buffer_shp__2_ = homef+"dissolve_buffer.shp"
   join_shp = homef+"join.shp"
   sum_inwoners = homef+"sum_inwoners.dbf"
   test_dbf = homef+"test.dbf"
   test2_dbf = homef+"test2.dbf"
   selectie = homef + "selectie.shp"
   resultaatclusters = homef + "rb_clusters.shp"
   temp = homef+"stedgeb.gdb\\temp"
   temp_PolygonNeighbors1 = homef+"stedgeb.gdb\\temp_PolygonNeighbors1"
   temp_stat = homef+"temp_stat"
   temp4_shp = homef+"temp4.shp"
   temp4_shp__3_ = temp4_shp
   #output
   tussenresultaat = homef+"tussenresultaat.shp"
   resultaat = homef+"resultaat_"+resultnum+".shp"
   print('run'+resultnum+'ruimtebeslag='+ruimtebeslag+', inw='+inwonersdichtheid+', tewerkst='+tewerkstellingsdichtheid)

   arcpy.env.overwriteOutput = True

   # orginele input shapefile kopiëren naar tijdelijk werk bestand
   arcpy.CopyFeatures_management(statistische_sectoren, statistische_sectoren_CopyFe, "", "0", "0", "0")

   # velden toevoegen waarin resultaten van verschillende stappen worden weggeschreven
   arcpy.AddField_management(statistische_sectoren_CopyFe, "stap1", "SHORT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
   arcpy.AddField_management(statistische_sectoren_CopyFe, "stap2", "SHORT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
   arcpy.AddField_management(statistische_sectoren_CopyFe, "stap3", "SHORT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
   arcpy.AddField_management(statistische_sectoren_CopyFe, "stap4", "SHORT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

   # Stap 1: opsplitsing veel / weinig ruimtebeslag: indien ruimtebeslag> drempelwaarde --> stap 1 = 1
   print('bereken stap1')
   arcpy.CalculateField_management(statistische_sectoren_CopyFe, "stap1", "calc( !ruimtebesl!)", "PYTHON_9.3", "def calc(rb):\\n if
rb>"+ruimtebeslag+":\\n  return 1\\n else:\\n  return 0")

   # Stap 2: combinatie met inwonersdichtheid en tewerkstellingsdichtheid: indien inwdichtheid en tewdichtheid > dempelwaarden --
> stap2 = 1
   print('bereken stap2')
   arcpy.CalculateField_management(statistische_sectoren_CopyFe, "stap2", "calc( !stap1!, !inwdicht!, !werkdicht!)", "PYTHON_9.3",
"def calc(stap1, inw, tew):\\n if ((stap1 ==1 and inw > "+inwonersdichtheid+") or (stap1 ==1 and tew >
"+tewerkstellingsdichtheid+")):\\n  return 1\\n else:\\n  return 0")
```

```
   # Stap 3: aaneengrenzende stedelijke sectoren samennemen, gaten opvullen : indien alle buren stedelijk --> stap 3 = 1
   print('stap 3: bereken aantal buren')
   arcpy.PolygonNeighbors_analysis(statistische_sectoren_CopyFe, statistische_sectoren_Polygo7, "CODSEC;stap2",
"NO_AREA_OVERLAP", "BOTH_SIDES", "", "METERS", "SQUARE_METERS")
   arcpy.Statistics_analysis(statistische_sectoren_Polygo7, statistische_sectoren_CopyFe2, "nbr_stap2 SUM", "src_CODSEC")
   arcpy.JoinField_management(statistische_sectoren_CopyFe, "CODSEC", statistische_sectoren_CopyFe2, "src_CODSEC")

   print('stap3: gaten vullen')
   arcpy.CalculateField_management(statistische_sectoren_CopyFe, "stap3", "select( !FREQUENCY!, !SUM_nbr_st!, !stap2!)",
"PYTHON_9.3", "def select(freq,sum,stap2):\\n if (stap2 == 1 or freq ==sum):\\n  return 1\\n else:\\n  return 0")
   #stat_analy(input, output, frequency field, summary field)
   #=> field "freq" in dit geval = "nbr_stap2 SUM" = som van aantal buren die stap 2 zijn (aantal stedelijke buren)

   # Stap 4: inwonersaantal van clusters checken: indien inwonersaantal van stedelijke cluster > drempelwaarde: stap 4 = 1
   print('stap4: dissolve')
   arcpy.Dissolve_management(statistische_sectoren_CopyFe, dissolve_shp, "stap3", "", "SINGLE_PART", "DISSOLVE_LINES")
   arcpy.Dissolve_management(dissolve_shp, dissolve_shp2, "stap3", "", "SINGLE_PART", "DISSOLVE_LINES")
   arcpy.Buffer_analysis(dissolve_shp2, dissolve_buffer_shp__2_, "-1 Meters", "FULL", "ROUND", "NONE", "")

   print('stap4: spatial join')
   arcpy.SpatialJoin_analysis(statistische_sectoren_CopyFe, dissolve_buffer_shp__2_, join_shp, "JOIN_ONE_TO_MANY", "KEEP_ALL",
"", "INTERSECT", "", "")

   print('stap 4: bereken aantal inwoners')
   arcpy.Statistics_analysis(join_shp, sum_inwoners, "inwoners SUM", "JOIN_FID")
   arcpy.JoinField_management(join_shp, "JOIN_FID", sum_inwoners, "JOIN_FID")

   print('bereken stap4: selectie indien groter dan drempelwaarde')
   arcpy.CalculateField_management(join_shp, "stap4", "stap4( !stap3!, !SUM_inwone!)", "PYTHON_9.3", "def stap4(stap3,sum):\\n
if (stap3 == 1 and sum>="+aantal_inw+"):\\n  return 1\\n else:\\n  return 0")

   # Stap 5: randstedelijke zones selecteren als aangrenzend bij stedelijke cluster: indien 1 van de buren = stedelijk (MAX_nbr_st = 1)
--> randstedelijk = 1 ! LATER OOK GECOMBINEERD MET VOLDOENDE RUIMTEBESLAG!
   print('Stap 5: bereken buren')
   arcpy.PolygonNeighbors_analysis(join_shp, test_dbf, "CODSEC;stap4", "NO_AREA_OVERLAP", "BOTH_SIDES", "", "METERS",
"SQUARE_METERS")
   arcpy.Statistics_analysis(test_dbf, test2_dbf, "nbr_stap4 MAX", "src_CODSEC")
   arcpy.JoinField_management(join_shp, "CODSEC", test2_dbf, "src_CODSEC")

   # velden toevoegen waarin resultaten van randstedelijk worden toegevoegd
   arcpy.AddField_management(join_shp, "randsted", "SHORT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
   arcpy.AddField_management(join_shp, "rand2", "SHORT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
   arcpy.AddField_management(join_shp, "rand3", "SHORT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
   arcpy.AddField_management(join_shp, "rand4", "SHORT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")

   print('Stap 5: indien VOLDOENDE RUIMTEBESLAG EN 1 buur = stedelijk --> randstedelijk')
   arcpy.CalculateField_management(join_shp, "randsted", "calc( !ruimtebesl!, !MAX_nbr_st!)", "PYTHON_9.3", "def calc(ruimtebesl,
MAX_nbr_st):\\n if ruimtebesl>"+ruimtebeslag+" and MAX_nbr_st==1:\\n  return 1\\n else:\\n  return 0")

   # Stap 6: extra stap: indien ruimtebeslag > drempelwaarde en aangrenzend aan randstedelijk --> ook randstedelijk maken
   print('Stap 6: bereken buren')
   arcpy.PolygonNeighbors_analysis(join_shp, test_dbf, "CODSEC;stap1;randsted", "NO_AREA_OVERLAP", "BOTH_SIDES", "",
"METERS", "SQUARE_METERS")
   arcpy.Statistics_analysis(test_dbf, test2_dbf, "nbr_randst MAX;nbr_stap1 MAX", "src_CODSEC")
   arcpy.JoinField_management(join_shp, "CODSEC", test2_dbf, "src_CODSEC")

   print('Stap 6: indien ruimtebeslag = hoog en aangrenzend: randstedelijk maken')
   arcpy.CalculateField_management(join_shp, "rand2", "select( !randsted! , !MAX_nbr_ra!, !stap1! )", "PYTHON_9.3", "def
select(rand1, rand, rb):\\n if ( rand1 == 1 or (rand >= 1 and rb ==1)):\\n  return 1\\n else:\\n  return 0")

   #indien alle clusters van aaneengesloten sectoren met hoog ruimtebeslag moeten toegevoegd worden: onderstaande blok code
ook uitvoeren
   #indien enkel aangrenzende sectoren met hoog ruimtebeslag moeten toegevoegd worden: deze blok code verwijderen, en in stap
7 opnieuw overstappen op 'join_shp' ipv 'resultaatcluster' en op 'rand2' ipv 'rand3'
   print('Stap 6: clusters aangrenzend met hoog ruimtebeslag toevoegen')
   arcpy.Select_analysis(join_shp, selectie, '"stap1" = 1') #stap 1 = 1 => rb > drempelwaarde
   arcpy.Dissolve_management(selectie, dissolve_shp, "stap1", "", "SINGLE_PART", "DISSOLVE_LINES")
   arcpy.Buffer_analysis(dissolve_shp, dissolve_buffer_shp__2_, "-1 Meters", "FULL", "ROUND", "NONE", "")
   arcpy.MakeFeatureLayer_management(dissolve_buffer_shp__2_, 'selectielaag')
   arcpy.MakeFeatureLayer_management(join_shp, 'selectielaag2')
```

```
   arcpy.SelectLayerByAttribute_management('selectielaag2', "NEW_SELECTION", '"rand2" = 1') #rand2=1 => randstedelijk (want RB hoog en aangrenzend
   arcpy.SelectLayerByLocation_management('selectielaag', 'intersect', 'selectielaag2')
   arcpy.SelectLayerByAttribute_management('selectielaag2', "CLEAR_SELECTION")
   arcpy.SelectLayerByLocation_management('selectielaag2', 'intersect', 'selectielaag')
   arcpy.CalculateField_management('selectielaag2', "rand3", "1", "VB")
   arcpy.SelectLayerByAttribute_management('selectielaag2', "CLEAR_SELECTION")

   arcpy.CopyFeatures_management('selectielaag2', resultaatclusters)
   arcpy.CalculateField_management(resultaatclusters, "rand3", "rand3( !rand2!, !rand3!)", "PYTHON_9.3", "def rand3(stap2, stap3):\\n if (stap2 == 1 or stap3 == 1):\\n  return 1\\n else:\\n  return 0")


   # Stap 7: gaten in randstedelijk gebied opvullen
   print('Stap 7: bereken buren')
   arcpy.PolygonNeighbors_analysis(resultaatclusters, test_dbf, "CODSEC;rand3", "NO_AREA_OVERLAP", "BOTH_SIDES", "", "METERS", "SQUARE_METERS")
   arcpy.Statistics_analysis(test_dbf, test2_dbf, "nbr_rand3 SUM", "src_CODSEC")
   arcpy.JoinField_management(resultaatclusters, "CODSEC", test2_dbf, "src_CODSEC")

   print('Stap 7: gaten vullen')
   arcpy.CalculateField_management(resultaatclusters, "rand4", "select( !FREQUENC_4!, !SUM_nbr_ra!, !rand3! )", "PYTHON_9.3", "def select(freq,sum,rand):\\n if (rand == 1 or freq ==sum):\\n  return 1\\n else:\\n  return 0")

   # resultaat wegschrijven naar nieuwe output-laag. Resultaat zit in kolom 'totaal': 0 indien landelijk, 1 indien stedelijk, 2 indien randstedelijk
   print('(tussen)Resultaat wegschrijven')
   arcpy.AddField_management(resultaatclusters, "totaal", "SHORT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
   arcpy.CalculateField_management(resultaatclusters, "totaal", "select( !stap4!, !rand4!)", "PYTHON_9.3", "def select(sted, rand):\\n if (sted == 1):\\n  return 1\\n elif (rand == 1):\\n  return 2\\n else:\\n  return 0")
   arcpy.CopyFeatures_management(resultaatclusters, tussenresultaat, "", "0", "0", "0")

   # stap 8: dichtplamuren van gaten van meerdere statistische sectoren omgeven door enkel stedelijk
   print('stap8')
   arcpy.Dissolve_management(tussenresultaat, temp, "totaal", "", "SINGLE_PART", "DISSOLVE_LINES")
   arcpy.PolygonNeighbors_analysis(temp, temp_PolygonNeighbors1, "OBJECTID;totaal", "NO_AREA_OVERLAP", "BOTH_SIDES", "", "METERS", "SQUARE_METERS")
   arcpy.Statistics_analysis(temp_PolygonNeighbors1, temp_stat, "nbr_totaal SUM", "src_OBJECTID")
   arcpy.JoinField_management(temp, "OBJECTID", temp_stat, "SRC_OBJECTID", "FREQUENCY;SUM_NBR_TOTAAL")
   print('stap8: spatial join')
   arcpy.SpatialJoin_analysis(tussenresultaat, temp, temp4_shp, "JOIN_ONE_TO_ONE", "KEEP_ALL", "", "WITHIN", "", "")
   arcpy.AddField_management(temp4_shp, "stap8", "SHORT", "", "", "", "", "NULLABLE", "NON_REQUIRED", "")
   # resultaat wegschrijven naar nieuwe output-laag
   arcpy.CalculateField_management(temp4_shp, "stap8", "select( !FREQUENC_5!, !SUM_nbr_to! )", "PYTHON_9.3", "def select(FREQUENCY_,SUM_nbr_to):\\n if (FREQUENCY_ == 1 and SUM_nbr_to ==1):\\n  return 1\\n else:\\n  return 0")
   print('resultaat wegschrijven')
   arcpy.CalculateField_management(temp4_shp, "totaal", "select( !stap4!, !rand4!, !stap8!)", "PYTHON_9.3", "def select(sted, rand, stap8):\\n if (sted == 1 or stap8==1):\\n  return 1\\n elif (rand == 1):\\n  return 2\\n else:\\n  return 0")
   arcpy.CopyFeatures_management(temp4_shp, resultaat, "", "0", "0", "0")
```

