South African datazones: A technical report about the development of a new statistical geography for the analysis of deprivation in South Africa at a small area level

David Avenell, Michael Noble and Gemma Wright

Working Paper No 8

Centre for the Analysis of South African Social Policy
University of Oxford

Working Paper No 8

South African datazones: A technical report about the development of a new statistical geography for the analysis of deprivation in South Africa at a small area level

By David Avenell, Michael Noble and Gemma Wright

Centre for the Analysis of South African Social Policy
Department of Social Policy and Social Work
University of Oxford, England

May 2009

About the authors

David Avenell is GIS Consultant for the Centre for the Analysis of South African Social Policy (CASASP) at the Department of Social Policy and Social Work (DSPSW) at the University of Oxford.
Professor Michael Noble is Professor of Social Policy and Director of CASASP at the University of Oxford. He is also an Honorary Fellow of the Human Sciences Research Council and Adjunct Professor at the University of Fort Hare.
Dr Gemma Wright is a Senior Research Fellow and Deputy Director of CASASP at the University of Oxford.

Recommended citation

Avenell, D., Noble, M. and Wright, G. (2009) *South African datazones: A technical report about the development of a new statistical geography for the analysis of deprivation in South Africa at a small area level,* CASASP Working Paper No. 8, Oxford: Centre for the Analysis of South African Social Policy, University of Oxford.

Acknowledgements

The datazone development work was funded by the Republic of South Africa's National Department of Social Development. The resources were made available by the UK Department for International Development Southern Africa as part of its SACED Programme (Strengthening Analytical Capacity for Evidence-Based Decision-Making). The other members of the research team are thanked (in alphabetical order) for their contributions to the datazone development work: Helen Barnes, Prof Andrew Dawes, David McLennan, Benjamin Roberts (HSRC) and Dr Adam Whitworth.

Disclaimer

1 Introduction

Datazones are geographical units which were developed by the authors to enable deprivation in South Africa to be analysed at a small area level. Prior to their development the research team had produced *ward* level Provincial Indices of Multiple Deprivation (PIMD) for each province in South Africa (Noble *et al.,* 2006a, 2006b, 2009a). Wards were the best sub-municipality level geographical unit available at the time. However, wards vary greatly by population size and so are not an optimal unit to use for this purpose (see Noble *et al.,* 2006a pp.53-54).

Having produced the PIMD, the research team sought to develop a new statistical geography – the datazones – in order to produce a geographical unit with a tighter population size range. This enables deprivation at an area level to be compared across the whole of South Africa. This report provides a technical account of how the datazones were produced. Though the datazones are in essence no more than 'empty shells' it was essential that they were carefully constructed to meet the purposes that had been identified for them. The techniques used were complex and build on work undertaken internationally to create small area level statistical geographical units. Similar geographical units have been developed elsewhere for the measurement of small area level deprivation (e.g. Noble *et al.,* 2006).

In terms of their application, the datazones have been used to develop a datazone-level South African Index of Multiple Deprivation (SAIMD) (Noble *et al.,* 2009b) based on data from the 2001 Census. The SAIMD provides a small area level profile of deprivation experienced by the total population (children and adults of all ages).

The datazones have also been used to produce a datazone-level South African Index of Multiple Deprivation for Children (SAIMDC) (Wright *et al.,* 2009), also based on data from the 2001 Census, but relating only to children aged 0-17 inclusive. The SAIMDC takes forward municipality-level research undertaken in relation to child poverty and deprivation in South Africa (Barnes *et al.,* 2007 and 2009).

2 Background

The lowest level census geography for the 2001 South African census is the enumeration area (EA). To maintain confidentiality, census data has not been publicly released at this level by Statistics South Africa. Instead, a new higher level geography - the Small Area Level (SAL) geography - was created for census data dissemination. SALs are created from aggregations of EAs and have a minimum population of 500. EAs were merged only on the basis of geographic proximity and population size – and not neighbourhood and population type.

EAs and SALs were both considered as possible 'building blocks' for the datazones, i.e. the units from which datazones could be constructed. This section outlines the overarching aims for the new datazone geography, and the process of selecting which building blocks to use.

Datazone requirements

The intentions were that the datazones would provide a new statistical (not political) geography to better delineate pockets of deprivation and that they should also contain similar numbers of people so that each area could be compared alongside all other areas in the country. The aims for the new datazone geography were that they should fulfil the following requirements -

- The datazones should have a similar population size, of between 1000 and 3000 and target size of 2000
- The datazones should delineate pockets of deprivation by maximising datazone social homogeneity and population density homogeneity
- The datazones should be a manageable geography, placing controls on datazone size and shape – to prevent creation of very large datazones and to minimise complex shapes
- The datazones should nest within existing census municipality and province geographies
- The datazones should form a continuous geography, constructed from contiguous lower level building blocks

Selection of the building block

The two geographies, of sufficient granularity, that were available as building blocks for the creation of the datazones were EAs and SAL polygons.

Of the 80,787 EA boundaries in South Africa, 50.4% are identical to SAL boundaries. Since SAL boundaries are only created where EA populations fall below 500, the majority of EAs with a population greater than 500 translate directly to a SAL boundary.

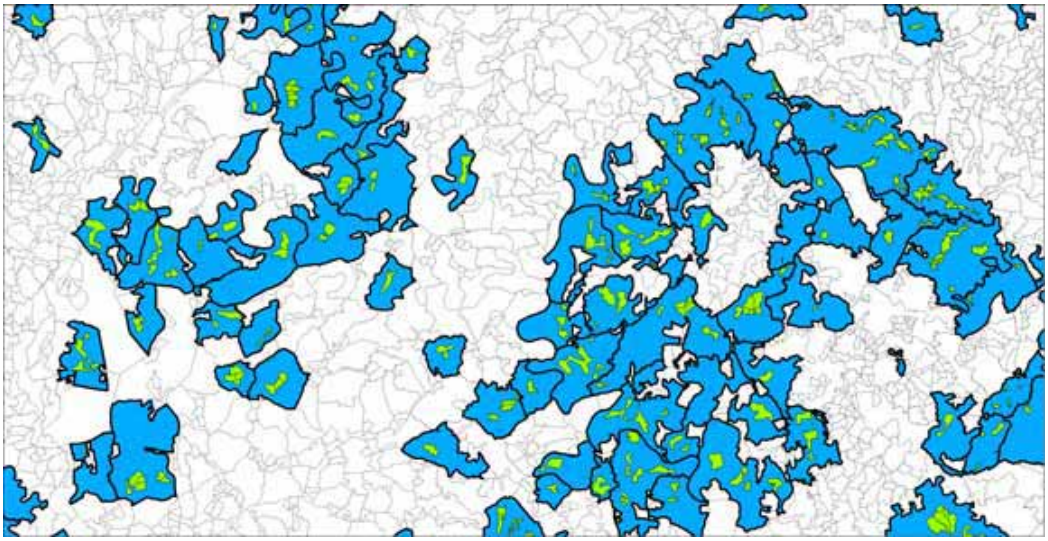
After close scrutiny of the EA and SAL boundaries the following observations were made:

(1) Fragmented EAs and SALs: A single EA may consist of several geographically separated sections. Datazones constructed from these types of EAs may end up as more than one geographically separated polygon if other EAs within the same datazone are unable to fill the gaps. This is a limitation of using EAs, but not a reason to exclude their use in the light of their advantages and the limited availability of alternative geographies. In most cases the SAL geography inherits this feature.

(2) Non-contiguous geographic structure: EAs may have a geographic structure that is not truly contiguous. EAs can exist within the greater boundary of other EAs. This creates the effect of 'Island EAs' (those that are entirely inside other EAs) and 'Sea EAs' (those that completely surround an Island EA). In some cases a Sea EA (SEA) can contain more than one Island EA (IEA).

Occurrence of SEAs and IEAs within the EA (and SAL) geography have the following implication in the rule based datazone growing process -

In a situation where a SEA contains an island (or many islands) with a population less than the minimum datazone threshold of 1000 - an attempt has to be made to join the island(s) to the sea. However, there is a possibility that the SEA and IEA will have quite different population densities and types (an IEA is more likely to represent a village/urban area and a SEA more likely to represent a surrounding sparsely populated rural area) and therefore rule based factors may exist to de-prioritise the joining of SEAs and IEA (EAs are prevented from merging if difference in population density is too great). This may create a geography with stranded sub-minimum population single EA datazones (i.e. a single IEA) surrounded by vast swathes of merged SEAs (from forced merger of many low population SEAs and other more rural EAs to meet the minimum population thresholds, and prevention of incorporation of IEAs due to differences in population density or population type). The figure below provides an example of IEAs located within SEAs.



SEAs are blue; IEAs are green, darker outline shows datazone boundary after merging linked SEAs and IEAs. South Africa has 1413 SEAs, 6919 IEAs and 10 cases where EAs are both SEAs and IEAs

(3) Literal islands: A small number of EAs are literally islands in the ocean that do not have contiguous EAs, This may result in datazones with population below the minimum threshold.

(4) Small municipalities: A small number of municipalities have a population less than the minimum allowable for a datazone – and therefore datazones created within these zones will also have a population less than the minimum allowed.

(5) Additional SAL issues:Of the 56,255 SAL boundaries in South Africa, 72.4% are identical to EA boundaries. An average of 3.31 EAs were assigned to each SAL boundary. SAL geographies inherit geographic problems associated with EAs, such as geographically separated sections, SEAs and IEAs. Some SALs were created from EAs that are not contiguous and hence SALs have a non-contiguous structure which is problematic for datazone creation. Additionally, population and geographic location were taken into account when SALs were constructed from EAs and not neighbourhood type – so the ability to delineate pockets of deprivation is inevitably reduced using a SAL level building block. Since a number of SAL boundaries were created from more than one EA boundary – the SAL geography has a higher average population per area than EAs and has a coarser geographic resolution. SAL geography does however have the advantage of limited census data availability (without interpolation), whereas census data has not been released at EA level, so variables assigned to EAs would need to be derived from higher geographic level publicly released SAL census data.
.

Eventually, EAs were selected as the datazone building block for the following key reasons –

- EA boundaries are available at a finer resolution than SALs.
- EAs have smaller populations and areas (where they do not share same boundaries as SALs) – providing greater flexibility for creation of a higher level geography. A larger amount of geographic information is available at a finer resolution (area, population) – potentially providing a greater number of ways to construct the new geography within the constraints of the rule base. This should result in a more optimised geography
- EAs are less geographically complex than SALs
- A significant number of EAs share the same geographic extent as SAL boundaries - allowing attribution of SAL level census data directly to a large number of EAs and interpolation to remaining EAs that form part of SALs. This enables examination of population size, population density and population type at EA level in the datazone creation process – and generation of population classifications using cluster analysis.

3 Summary of the Methodology

Introduction

This section summarises the methodology that was applied to construct datazones. Details about the precise steps (the 'procedures') that were undertaken are provided in the following section.

The techniques used for creation of South Africa datazones have drawn upon automated zoning procedures (AZP) introduced by Stan Openshaw, and adapted for the 2001 census of England and Wales by David Martin (please see additional references at end of paper). Due to the complex geography of South African Enumeration Areas (to be used as building blocks) and stringent requirements of datazones for identifying pockets of deprivation – the creation of South African datazones uses a very loose form of AZP and combines this with use of a complex iterative rule base. The software packages MapBasic and STATA were used.

The broad concept used was the same as used within AZP: given a set of contiguous areas, create a smaller set of contiguous areas that meet certain general criteria; then optimise this initial aggregation based upon an objective function (derived from statistics related to the newly created geography).

The same broad structure was followed for constructing the South African datazones. An initial aggregation was created and this was then optimised. However, the use of 'random' moves was removed and replaced by complex multi-layered logic - due to the large numbers of rules that had to be satisfied. The major datazone creation process occured during the initial allocation of EAs to datazones, and the optimisation process was reduced to a small number of rule-based iterations

At the heart of this process were a series of fixed rules that had to be obeyed, for all but exceptional circumstances, and a series of optimising rules that were applied within the boundaries of the fixed rules, wherever possible.

Datazone creation: fixed rules

- Population greater than 1000 and less than 3000
- Datazones must contain contiguous EAs
- Datazones must nest into municipality and province boundaries
- EAs without population become empty datazones. These cannot become part of other datazones (except for creation of linked SEA/IEA datazones to reduce creation of very large datazones)

Within constraints of the fixed rules attempts were made to optimise datazone design using a series of optimisation rules:

Datazone creation: optimising rules

- Population optimised towards 2000
- Homogeneity of population type classification within datazones to be maximised
- Homogeneity of population density within datazones to be maximised
- Enhanced delineation of urban and rural settlements – using population density indicators
- Datazones to maintain efficient shape – using a compactness ratio
- Datazones to be restricted from growing too large (geographically)

Fixed rules were always obeyed. Optimising rules were applied strictly at onset and then progressively loosened through iterative rounds (in both development of initial aggregation and optimisation stages) to provide increasing flexibility for construction of a continuous coverage of datazones.

The initial aggregation stage

Each municipality (split by province) was considered in turn as the 'universe' for creation of datazones. That is, datazones were created for one municipality at a time as datazones nest within municipality boundaries.

The initial aggregation process allocated each EA in South Africa to a datazone – through application of the fixed and optimising rules. The stage consisted of two core procedures – datazone growing (Procedure 2, below) and assignment of remaining EAs to existing datazones (Procedure 3, below). These two procedures were alternated and implemented iteratively – through a series of rule rounds – where each successive rule round relaxed the rule base slightly – allowing greater freedom for datazones to grow, though less optimally. Specific routines were then applied to ensure continuous datazone coverage for South Africa, and to manipulate existing underpopulated/ overpopulated datazones (created as the final step to ensure continuous datazone coverage) to bring these back within population tolerances.

Manual adjustment

In the small number of cases where, after initial aggregation, datazones fell below minimum population thresholds manual adjustment was applied to alter local datazone geographies in such a way as to minimise this effect.

The optimisation stage

This stage took the initial aggregation of EAs and attempted to further optimise through a series of sweeps through all EAs in South Africa – looking to implement movements of EAs between adjacent datazones that result in increased datazone optimisation. This used the same rule base as the initial aggregation procedure. EAs at the edge of every datazone were evaluated. At

completion of an optimisation round, and given a change in the datazone landscape, a further optimisation round was applied – allowing propagation of change.

Some datazone statistics

- 85% (19,503) of datazones were created by initial aggregation growing phase (procedure2 and 3) using the very tightest cluster and population density rules
- 167 non-DMA datazones exceed maximum population thresholds (of which 120 contain single EAs with population greater than maximum)
- 568 non-DMA datazones fall below minimum population thresholds (of which 562 have 0 population and 2 are Islands)
- 11 DMA datazones exceed maximum population thresholds
- 13 DMA datazones fall below minimum population thresholds
- Average datazone population = 1962 (STD = 648)
- Average EAs per datazone = 3.54 (STD = 2.49)
- Maximum EAs per datazone = 34

The rest of this report provides details about the eleven procedures that were undertaken in order to produce the datazones. The procedures are written-out descriptions of the code steps that were taken, which is referred to by practitioners as ‘pseudo-code’.

4 The Detailed Methodology: The Procedures

This section provides details about the eleven procedures that were undertaken to produce datazones in South Africa.

Procedure 1: Pre-processing

EA populations
EA populations were not available for this project so these were derived from SAL population data. Even population distribution across each SAL was assumed. An EA boundary with the same extents as a SAL boundary is given the same population as the SAL. In the case of a SAL boundary made up of more than one EA, the SAL population was distributed to EAs weighted by geographic area. The Table Mountain area of Cape Town was given special treatment. Here, EAs void of population (identified through aerial imagery) were excluded from the geographic area weighted population re-allocation - ensuring re-allocation to populated EAs only.
Linking of polygons containing smaller polygons (i.e. SEAs and IEAs)
Geographic techniques were used to identify Island EAs (IEAs) and Sea EAs (SEAs). Connected SEAs and IEAs were given identifiers to allow particular SEA/IEA combinations to be managed as single units within datazone growing and optimisation procedures. This helped maintain datazone compactness and prevent creation of large numbers of 'stranded' IEAs within very large datazones dominated by SEAs.
EA adjacency matrix
A list of EAs and adjacent EAs was created for all of South Africa using topological relationships existing within the Enumeration Area geography - using ESRI ArcGIS.
Assigning EA population cluster types
EAs are assigned to one of seven cluster types. A small number of EAs are not assigned a cluster type - typically empty or low population EAs. Cluster types are required within the rule base for creation of datazones. Cluster analysis was performed for a series of variables at SAL geography. EAs inherited the cluster type of the SAL to which they belonged. Cluster types were aggregated to cluster groups and then used within the rule-base. Cluster analysis was implemented on a Province basis and **cluster rules are province specific**. The cluster analysis was undertaken in STATA using variables available from the SAL-level 2001 Census.
Deriving EA population density rules
An analysis of EA population density across the country revealed population density ranges for typical urban and rural communities which varied for different parts of the country. Therefore, municipalities were grouped into three classes (former homelands, metropolitan areas and the remainder of South Africa) and each class assigned a particular set of population density thresholds for use within the rule base. These were built into the rule base to restrict the ability of datazones to cross distinct urban/rural boundaries during the datazone growing stages where the full restrictive rule base was applied. Three population density ranges were created for each municipality according

to which class it fell within – the lower range defining more rural EAs, the mid range defining mixed urban/rural and the upper range defining more urban EAs.

Creation of special datazones

A number of datazones were created as pre-processing steps as special datazones –

- EAs with 0 population
- EAs with a population greater than 3000
- EAs without adjacent EAs (e.g. islands)
- Linked chains of SEAs and IEAs with a population between 1000 and 3000 (and incorporating any EAs that are part of the chain that have 0 population)
- Municipalities with contiguous EAs and total population < 1000

District Management Area datazones

Municipalities that are District Management Areas are allocated as a single datazone. The population of DMA datazones may fall outside of thresholds – and populated and non-populated EAs are allowed to merge in these regions. 25 DMAs are converted to 26 datazone DMAs. 2 DMA datazones are created to cover Kruger National Park – one for the part of the DMA covered by each province.

Procedure 2: Datazone growth from seed EA

Operating within restrictions imposed by the *fixed rules* the following steps are implemented to grow a datazone that meets criteria, from a single seed EA -

2a All EAs unallocated to a datazone within the current municipality 'universe' and untested as a 'seed' polygon for datazone growth are ordered by SEA/IEA and then population density

2b Select first EA from list created in 2a, if none then **END**

2c If selected EA is an IEA or SEA and the population of all Eas within the linked SEA/IEA chain is less than minimum population and EAs in linked chain are not currently allocated to a datazone then allocate all EAs to growing datazone – as long as the new growing datazone population is less than the maximum population (IEA/SEA in a linked chain with population within tolerances would already have been assigned to a datazone as a pre-processing step for this municipality)

2d If selected EA is an IEA or SEA and the population of all EAs within the linked SEA/IEA chain is greater than maximum population, or if one or more EAs in the linked SEA/IEA chain are already allocated to a datazone then treat this IE/SEA as a normal EA

2e Obtain a set of EAs adjacent to the current growing datazone that have not been allocated to another datazone and fall within the same municipality.

Simplified illustration - the growing datazone (the 'seed' EA) is blue and adjacent EAs for consideration are yellow.

2f Deselect adjacent EAs that, if added to the growing datazone, compromise fixed rules or the current set of datazone growing rules at this stage – relative to the initial seed EA where applicable (i.e. unacceptable population cluster group or population density, would take population of datazone over maximum allowed, would take population of growing datazone further from target population if at the same time the current datazone population is within tolerances)

2g Adjacent EAs are also deselected if when added to the growing datazone, the population of growing datazone > minimum and the growing datazone area > 50 square km. This measure is an attempt to minimise creation of geographically large datazones

2h If valid adjacent EAs still remain, then calculate compactness ratio (explained later) for each EA (for the polygon created by adding this EA to the growing datazone). Sort valid adjacent EAs by compactness ratio (1 to 0) and select EA with highest compactness ratio indicator.

2i If an EA is selected in 2h and it is a SEA or IE - then attempt to add all linked EAs to growing datazone – only if the growing datazone population remains within tolerances and the linked EAs are not allocated to other datazones. Only datazone population rules are considered when looking to absorb entire SEA/IEA linked chains into the growing datazone – **ELSE** – treat this SEA/IEA as a normal EA for evaluation.

Simplified illustration - the growing datazone is blue, valid adjacent EAs for consideration are yellow, deselected adjacent EAs are darker yellow.

2j If an additional EA (or set of linked SEA/IEAs) was added to the growing datazone then **GOTO 2e – else GOTO 2k.** The figure below illustrates in more detail the new enlarged datazone – the blue areas are existing datazones that limit the ability for growth of new datazones. A new set of adjacent datazones to the newly enlarged datazone is shown, with a new set of valid adjacent datazones.

The growing datazone is shown (light blue) with valid and invalid adjacent EAs (yellow and dark yellow) with surrounding datazones (blue) that act as a barrier to growth

2k If there are no further valid adjacent EAs and the population of the growing datazone falls within tolerances then the growing datazone becomes a formal datazone and is accepted into the datazone geography. Else, the growing

datazone is rejected and associated EAs released to act as seeds for growth of future datazones or for consumption by other growing datazones.

Confirmed new datazone (light blue with a red boundary), no further valid adjacent EAs (dark yellow) and existing datazones (blue)

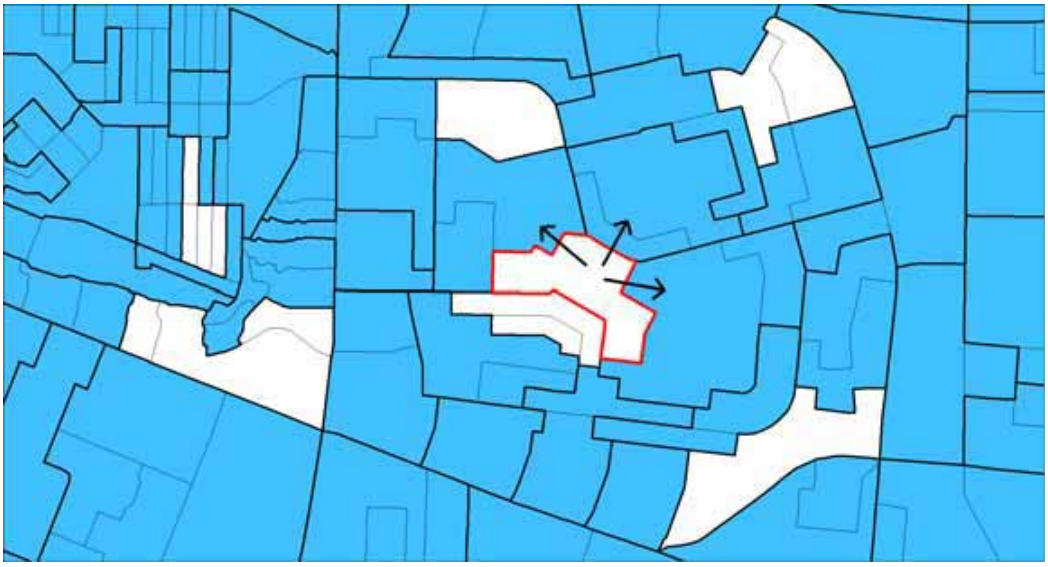
2l GOTO 2a and attempt to select a new seed EA for datazone growth.

The datazone landscape now incorporates the newly added datazone. The next EA to seed growth of a new datazone is circled

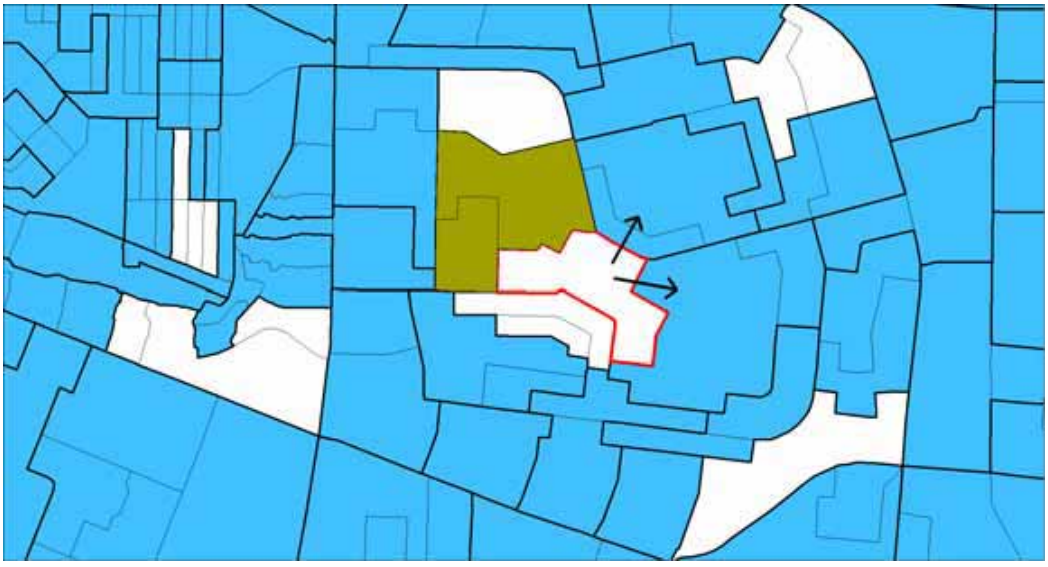
Procedure 3: Assignment of unallocated EAs to adjacent datazones

Unallocated single EAs are identified and an attempt is made to add to existing adjacent datazones – within the constraints of the current rule base. Each unallocated EA is considered once only in this procedure – and is compared to the current dynamic datazone geography (this changes as the

procedure progresses as increasing numbers of unallocated EAs become allocated)



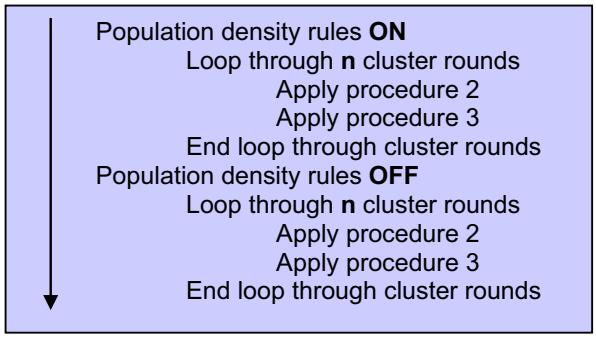
Datazones are blue, unallocated EAs are white. Red outline shows unallocated EA and potential consuming adjacent datazones



Joining the unallocated EA to the dark yellow datazone would break the current set of rules (cluster group, population count, population density), so the EA now has choice of joining 2 out of the original 3 datazones. It will be consumed by the datazone with the most similar population density (having satisfied the other rules). Datazone rules are relaxed in this procedure and datazone geographic area and compactness are not evaluated.

Procedure 4: Application of rule base – process flow logic

A rigorous procedure is applied to combined *procedure2* and *procedure3* and the rule base in an attempt to allocate all EAs to a datazone (not all EAs may be allocated in this procedure). Starting with the tightest rule base an attempt is made to create continuous datazone coverage for South Africa using *Procedure 2*. An attempt is then made to assign any unallocated EAs (again using the tightest rule base) using *Procedure3*. If unallocated EAs remain then the rule base is relaxed slightly (allowing more cluster types to merge) and *procedure2* and *procedure3* are applied again in sequence. This is performed a pre-defined number of times for each Province – resulting in a series of ever more flexible cluster group 'rule rounds', which vary by province. The entire procedure detailed above is then repeated but with population density rules removed - until finally – in the last cluster round with population density rules removed – only population tolerances are considered when attempting to assign unallocated EAs to datazones.



Process flow: initial aggregation

Procedure 5: Forceful allocation of all remaining EAs to create continuous datazone coverage, potentially containing underpopulated/overpopulated datazones

Procedure4 may not provide a datazone allocation solution for all EAs - despite ultimately relaxing all rules except for population tolerances and contiguity. *Procedure5* additionally removes population tolerances, allowing datazones to be created with populations lower than the minimum and greater than the maximum. This forces creation of continuous (but potentially problematic) datazone coverage. This is an intermediate step, allowing deviation from the acceptable – to be re-addressed in p*rocedure6* and *procedure7* – returning overpopulated datazones to within tolerances.

Step 1 - An attempt is made to allocate remaining unallocated EAs to the adjacent datazone with the lowest population (if one exists and its population is greater than 0). Unallocated EAs are tested in turn and allocated to adjacent datazones accordingly. If, within a complete sweep of all unallocated EAs, one or more EAs are successfully allocated to a datazone then another complete sweep takes place and this process continues until no further EAs can be assigned to datazones. This allows datazones to spread out to

consume EAs that are not originally adjacent. All rules are ignored except contiguity, inability to join to empty EAs and to cross province/municipality boundaries. **This step will introduce datazones with a population greater than the maximum allowed**

Step 2 – Remaining unallocated EAs are now sent to the datazone growing routine (**procedure2**) – but all rules are removed (except for contiguity, inability to join to empty EAs and to cross province/municipality boundaries). Creation of datazones with a population below 1000 is now possible. Datazones created by this Step will most likely be created from groups of EAs with a population < minimum that are prevented from growing into larger datazones (or being consumed by adjacent datazones) because they are entrapped by 0 population datazones, municipality boundaries or have a limited number of adjacent EAs (adjacency to the sea, lakes, province or municipality boundaries). **This step will introduce datazones with a population lower than the minimum allowed**

Step 3 – A check is performed to search for any unallocated EAs. All EAs should have been allocated before this step, regardless of their circumstance.

Procedure 6: Reducing numbers of overpopulated datazones

This procedure is designed to reduce the number of overpopulated datazones within the continuous datazone coverage (created in **procedure5**) using additional techniques now available, such as heuristic search. This attempts to draw a balance between obtaining the optimal result and required computational effort. Splitting overpopulated datazones takes place as a series of progressive steps – starting with 'quick wins' and moving towards computationally expensive complex problem solving heuristic routines.

Step 1: Splitting a datazone into one optimised and one or more additional datazones

Treating an overpopulated datazone as the *universe* ensures analysis is self-contained preventing knock on effects within the datazone landscape. This step breaks a datazone into 2 or more smaller datazones using a rigorous but computationally quick routine – creating one optimal datazone and one or more residual datazones - that fall within population tolerances but are not optimised.

1a: **Procedure2** is applied - growing a single datazone using full rules (contiguity, population threshold, population density, geographic area and cluster group). Cluster rule round 1 is used – hence a datazone is only allowed to contain the same cluster groups.
1b: **Procedure3** is then applied - assigning additional unallocated EAs within the universe to the growing datazone attempting to optimise towards the target population – using the same rule base as **step1a** (above)

If **steps 1a** and **1b** (above) create a datazone meeting population criteria (within tolerances and optimised to the target population) then the **steps 1c** and **1d** are applied to the remaining unallocated EAs within the *universe* (i.e. the remaining EAs in the original overpopulated datazone that were not allocated to the optimised *interna*l datazone created in **steps 1a** and **1b**). These steps are an attempt to create a second datazone by joining together the remaining EAs – and to assist this process – all rules, except population thresholds and contiguity, are removed

1c: Procedure2 is applied to remaining unallocated EAs to grow one or more additional datazones – with all rules removed except population thresholds and contiguity.
1d: Procedure 5 Step 1 is applied, with all rules removed except contiguity, to assign remaining unallocated EAs (in this universe) to an adjacent datazone with the lowest population

If *step1* results in one or more datazones containing populations within tolerances and all EAs within the *universe* are allocated to a datazone - then the new datazone configuration is accepted and the original non-split overpopulated datazone is removed.

Step 2: Splitting a datazone into *two* smaller datazones – based on population size and contiguity and optimised for compactness

More demanding overpopulated datazones are now split using a faster, less rigorous routine that preserves contiguity and population tolerances, but ignores cluster group and population density.
EAs within the overpopulated datazone again become the *universe*. These are sorted by latitude and longitude and the seed EA for initiating a growing routine is selected as *most likely* being in the south-west corner. EAs are appended to the growing datazone if remaining EAs (not in the growing datazone but within the *universe*) remain contiguous. Attempts are made to select EAs to add that will maximise datazone compactness. If the population of the growing datazone, and the population of all the remaining EAs fall within tolerances (and the remaining EAs are all contiguous) then the original datazone is removed and replaced with the two new lower population split datazones.

Step 3: Splitting a datazone into *two* smaller datazones – based on population size and contiguity - using heuristic breadth first exhaustive search

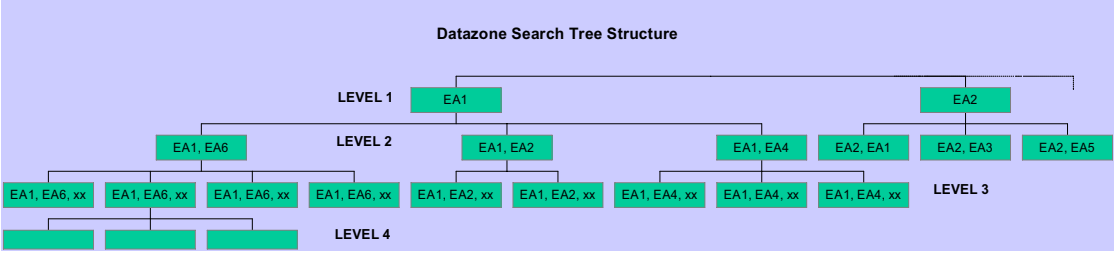
EAs within the overpopulated datazone become the *universe* and are assigned to one of two groups. **All** possible combinations of assignments of EAs to these two groups are tested – to find a configuration in which each group has internally contiguous EAs and a population size within thresholds. When a successful configuration is found the search will stop and the initial datazone is replaced by the two new smaller datazones. For computational reasons this routine is only applied to datazones with fewer than 6 EAs.

The routine is implemented using a 'list' structure and a tree-based search algorithm.

3a EAs within datazone are placed into a list. This is the top level of a search tree structure. For example a datazone may have four EAs - (EA1, EA2, EA3 and EA4). The list representation of this is shown below –

1
EA1
1
EA2
1
EA3
1
EA4
…

The list is structured so that the first entry ('1') defines the number of EAs that follow that are to be considered as a 'datazone' – in this case a single EA (EA1) is considered as the first datazone and all other EAs are considered as the second datazone – and both datazones are tested against the current set of rules. If the test is unsuccessful then the next configuration in the list is considered (EA2+all remaining EAs) and so on. Each position in the search list is represented on the search tree (below) and is referred to as a 'node'



Tree structure representation of list structure. Part of the tree structure for a single datazone. Nodes are shown as green boxes

3b The search starts at the top of the list and works down (within the tree structure this is equivalent to moving across the 'breadth' of the tree before moving down to the next level). The first test (node) considers two datazones – the single EA (EA1) - and - all remaining EAs within the datazone *universe*
3c Both sets of EAs are tested for internal contiguity (a single EA will always pass) and population size (fails if not within thresholds). If both sets of EAs meet these criteria then the search is successful – the original datazone is split into two smaller datazones.
3d If the two datazones ('EA1' and 'all the remaining EAs') are not valid then EAs adjacent to EA1 are selected. Each combination of EA1 and an adjacent EA are added to the bottom of the list (equivalent to adding a new branch with additional nodes to the search tree) for testing at a later stage and the search resumes at the current position higher up the tree. The process continues until criteria have been met or the list is exhausted.

For example – the developing list representing the tree structure above is shown below (horizontally, and incomplete)

1,EA1,**1**,EA2,**1**,EA3,**1**,EA4,**2**,EA1,EA6,**2**,EA1,EA2,**2**,EA1,EA4,**2**,EA2,EA1,**2**,EA2,EA3,**2**,EA2,EA5 …

Step 4: Splitting a datazone into *two* smaller datazones – based on population size and contiguity - using heuristic depth first search

A non-exhaustive version of depth first search was implemented for datazones containing six or more EAs – attempting an efficient traverse through the search tree structure to locate a problem solving node. Searches are restricted to those branches that are more likely to produce a favourable outcome. The procedure moves across the tree branches at the top level, searching down through each branch in turn before shifting to the next branch. This is implemented using a search list (**step3a** and **3c**), however, modified forms of procedure **3b** and **3d** are applied, attempting to more efficiently solve complex problems by introducing heuristic techniques.

A search progresses down through the nodes/sub-branches of a branch until either –

- The top level branch is exhausted
- A depth is reached within the branch where the number of nodes to be considered exceeds a pre-defined threshold
- The problem is solved

If a branch is exhausted without solving the problem – the search will resume from the top of the next branch – and work down through it's branch/node structure

If a tested node does not meet required criteria (both datazones within population thresholds and both contain contiguous EAs) and this is not at the bottom of the branch then rules are applied that determine which sub-branch is followed from this point to continue the downward search path (i.e. the adjacent EA to add to the collection of EAs representing the first datazone – and to be removed from the EA collection representing the second datazone).

A new sub-branch is created if both of the following conditions are true -

- one or both datazones at this node fall outside of population tolerances
- both datazones contain contiguous EAs

The EA selected to represent the new branch below the current node (i.e. to add to existing first datazone node set of EAs to create a new node and new entry on the search list) is the adjacent EA that best optimises the population of both resultant datazones and ensures both contain contiguous EAs. This

ensures the route traversed depth first through the search tree is optimised towards bringing the population of both datazones within tolerances and ensuring datazone contiguity. This considers other adjacent EA combinations (i.e. other branches from this node) as sub-optimal and forces their exclusion from the search process. Only the optimal search path is followed.

Procedure 7: Swapping EAs between adjacent datazones to solve overpopulated datazones

This attempts to solve any remaining problematic datazones by swapping EAs between adjacent datazones. The routine is positioned towards the end of the initial aggregation procedure because the datazone is not considered as the *universe* and analysis is not self-contained. Changes are made to adjacent datazones and propagation can occur through the datazone landscape. Placing this procedure here is an attempt to minimise knock on effects to other datazones.

EAs are shifted from overpopulated datazones to adjacent datazones, ensuring that the following criteria are satisfied –

Adjacent datazone population remains within tolerances
Reduction in deviation of overpopulated datazone population from maximum allowable
Overpopulated datazone retains EA contiguity after losing an EA

Priority is given to shifting an EA to an adjacent datazone containing an adjacent EA of the same cluster type – though this rule is relaxed if this is not possible

For each overpopulated datatzone –

7a Identify list of EAs that if removed would ensure the overpopulated datazone population remains above the minimum allowable threshold and falls within or more closely approaches the maximum threshold
7b Order the list of EAs selected in **7a** by EAs that, if removed, would leave the overpopulated datazone within tolerances with minimal deviation from the maximum allowable population, followed by EAs that if removed would leave the datazone population with least exceedance of of the maximum threshold. This ordering ensures that the amount of population transferred to an adjacent datazone is minimised – to reduce negative impact on the adjacent datazones optimised population.
7c Select first or next EA from the list generated in **7b** – the 'selected EA'
7d Additional rules.

- A selected EA that is a member of a sea/island chain will lead to the consideration of the entire sea/island chain as a single unit for shifting – to maintain sea/island integrity (see *Procedure1*).
- EAs can only shift if contiguity of donor and recipient datazones are maintained. This also applies if an entire sea/island chain is shifted – the

original sea/island chain may have attached additional EAs to create a datazone – so removing an entire island/sea chain must maintain contiguity of the bits of the original datazone that remain.
- A selected EA is prevented from shifting into datazones with population of 0

7e Select EAs adjacent to the selected EA - that belong to a different datazone. Compile list of adjacent datazones where population would remain within tolerances with the addition of the 'selected EA'
7f Attempt to shift the selected EA (or sea/island chain) into an adjacent datazone identified in **7e** – with priority given to (1) selecting an adjacent datazone identified by an EA with the same cluster type as the selected EA. (2) Secondly by minimising adjacent datazone deviation from target population
7h If an EA has NOT shifted in this round and the list generated in **7b** is not exhausted then **goto 7c**
7i If an EA has shifted in this round and the overpopulated datazone is still above the maximum population threshold then **goto 7a**
7j If an EA has shifted in this round and population of the overpopulated datazone is within population thresholds then **Stop – datazone solved, save changes**
7j Stop – datazone not solved, possibly further optimised, save changes

Procedure 8: Re-application of existing procedures in a changed datazone landscape

If *procedure7* causes a change to the datazone landscape and datazones remain unresolved then previous algorithms are repeated – for these may now provide resolution as the initial datazone configuration will have altered. Datazone splitting routines are repeated in the order listed below –

- *Procedure 6, step1*
- *Procedure 6, step2*
- *Procedure 6, step3*
- *Procedure 6, step4*

Procedure 9: Enhanced delineation of small urban areas surrounded by rural areas

An attempt is made to improve delineation between urban and rural areas – targeting datazones that contain both IEAs and SEAs. A datazone containing both types of EA indicates likelihood of a 'bleed' from an urban area into a rural area – assuming SEAs are most likely to be large predominantly rural EAs. In many cases datazones containing both SEAs and IEAs are acceptable – to satisfy population criteria and prevent datazones with large geographic areas. This routine examines existing datazones constructed from both SEAs and IEAs and attempts to further optimise - creating 2 or 3 smaller datazones that prevent mixing of SEAs and IEAs. It also considers datazones

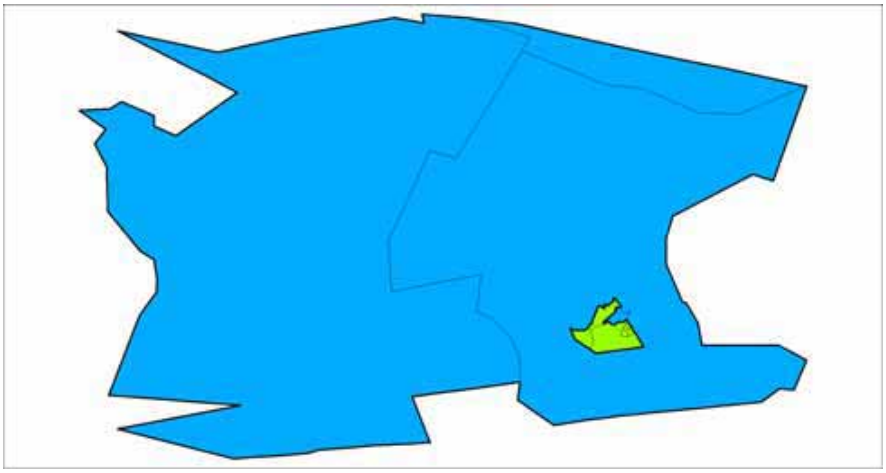
that exist within the outer boundary of the original datazone (i.e. datazones completely surrounded by other datazones/datazones within datazones) and also making these EAs available – providing greater flexibility for restructuring datazone geography beneath the original datazone (by removal of mixed SEAs/IEAs datazones). Two techniques are applied – merging of EAs of the same type (i.e. IEA or SEA) and a combination of merging SEAs and splitting IEAs (using search algorithms defined in *procedure6 step3* and *step4*)

For datazones that contain both SEAs and IEAs –

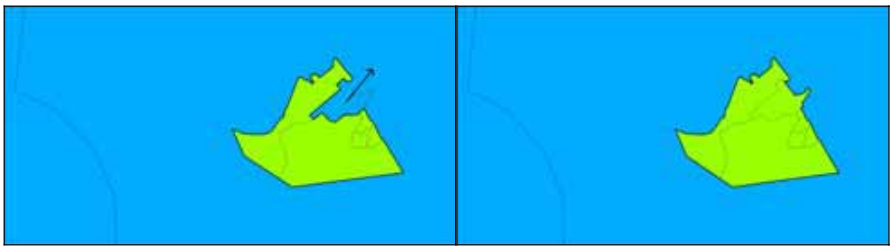
9a Merge EA polygons into a single polygon – and infill holes created by internal datazones
9b Select EAs falling within the polygon created in **9a** – using point in polygon (algorithms used ensure polygon centroids fall within polygon boundaries). The selection may include entire sets of EAs that form datazones falling entirely within the datazone being examined.
9c Attempt to allocate all EAs selected in **9b** to one of **2 datazones** – (1) merger of all SEAs and (2) merger of all IEAs. If the population of each datazone falls within tolerances and both datazone EAs are contiguous then accept the new datazones and remove the original datazone. **END**
9d Attempt to allocate all EAs selected in **9b** to one of **3 datazones.** If all SEAs are contiguous and their population falls within tolerances then merge to form the first datazone. If the remaining IEAs are contiguous (the population will exceed the maximum, else 8c would have resolved this) then search routine (**Procedure 6 step 3 or step 4**) is applied - attempting to split and create two additional datazones - with contiguous IEAs and populations within tolerances. **END**
9e Procedure not successful for this datazone. **END**

This procedure may result in –

(1) The original datazone splitting into two smaller datazones
(2) The original datazone splitting into two smaller datazones incorporating entire sets of EAs from one or more internal datazones
(3) The original datazone splitting into three smaller datazones incorporating entire sets of EAs from one or more internal datazones
(4) No change



Large datazone (blue) contains: two standard EAs (left and top), one SEA (right), one IEA(small blue polygon next to the green) and one interior datazone (green) made up of IEAs



The internal datazone (green) expands to consume the IEA that originally was allocated to the large EA (blue). The population of both datazones will remain within tolerances, but potentially less optimised – as priority has been given to refinement of the likely urban/rural boundary

Procedure 10: Manual enhancement (prior to optimisation)

Procedures 1 to 9 create a continuous datazone landscape covering South Africa. This geography is not perfect – but is highly optimised to the rule base applied. Datazones can occur with population less than the minimum allowable, for the following reasons –

- The datazone is a single EA with 0 population i.e. an empty EA
- The datazone consists of one or many EAs that are constrained by either a physical or administrative boundary (i.e. literally an island in the ocean or a municipality boundary) and the maximum population that can be achieved by assigning all EAs to a single datazone is lower than the minimum threshold permitted in the rule base.
- The datazone consists of one or many EAs that are prevented from merging with surrounding EAs due to rules governing mixing of empty and populated EAs and contiguity rules

Manual procedures are implemented to remove low population datazones, where possible, as follows –

- Low population datazones, separated (and prevented from merging) by empty (0 population) datazones are allowed to merge with empty datazones to create a single enlarged datazone satisfying minimum population criteria.
- Table Mountain is especially prominent within the Cape Town landscape and an obvious misallocation of SAL populations to EAs allocated population to a large empty EA. This population was re-allocated back to relevant urban EAs, preventing 'bleeding' of population onto the unpopulated part of the mountain.

Datazone numbering
Datazones are numbered using the following structure
Municipality code + "_" + sequential number e.g. 171_9
Datazones representing DMAs are labelled DMA1, DMA2 etc

Procedure 11 – Datazone Optimisation

Procedures 1 to 10 create an 'initial aggregation' of EAs to datazones. This, however, is not an 'initial random aggregation' as commonly referred to in Automated Zoning Procedures (AZP) literature. It is a highly optimised allocation of EAs to datazones achieved through geoprocessing and manipulation of a complex EA and SAL geography. Datazone optimisation, implemented in this procedure, is therefore a fine tuning process rather than the core component of an AZP based approach. AZP is not suited to a complex rule base and low level geography.

The partially optimised datazone geography is further optimised using the following parameters –

- target population size
- compactness
- population density homogeneity
- social homogeneity (via cluster type homogeneity)

ensuring the following fixed rules are maintained –

- minimum and maximum population threshold
- contiguity
- geographic area size

A number of sweeps through the datazone landscape take place. During each sweep each EA in South Africa is considered as a candidate for shifting into an adjacent datazone. A datazone receiving an EA may return an existing EA to the donating datazone – if this provides balancing and further optimisation. Only those moves that improve local optimisation (i.e. of the two evaluated datazones) are allowed. EAs at the edge of each datazone are considered as candidates for shifting into all datazones adjacent to that EA. EAs at the edge of datazones receiving an EA are also considered for shifting back to the donating datazone – creating a *swapping* effect.

EAs with 0 population or single EA datazones (e.g. islands in the sea, EAs with population greater than the maximum allowed) are not allowed to move between datazones. EAs are preventing from joining datazones with 0 population.

For each datazone in South Africa with a population > 0 and containing more than one EA –

11a calculate datazone compactness ratio.

Compactness Ratio
The compactness ratio is the ratio of an area, AreaA to the area of a circle, CircleA, that has the same circumference as the perimeter of AreaA. It is a measure of circularity – of shape efficiency. Put simply – we take a polygon of interest, calculate its perimeter, create a circle with the same perimeter and look at the relationship between the area of the polygon and area of the circle to provide a statement on compactness of the polygon.

The Compactness Ratio

Given the area of a circle, a, is defined as follows -
*a = pi * r * r*
and the circumference, c, of a circle is defined as follows -
*c = 2 * pi * r*
the radius, r, of a circle can then be defined as -
*r = c / (2 * pi)*
so the area of a circle can therefore also be defined as -
*A = pi * ((c / (2*pi)) * (c/(2*pi)))*

To obtain the area of a circle with the same perimeter as the datazone of interest, the datazone perimeter is substituted for the circle circumference in the equation above, to give the following -

A = pi * ((datazone_perimeter / (2 * pi)) * (datazone_perimeter/(2*pi)))

The compactness ratio is defined as -

compactness ratio = sqr(area of datazone/area of circle having same circumference as perimeter of datazone)

resulting in the following equation for use –

compactness ratio = sqr(datazone_area / A)

A circle has a compactness ratio of 1 – and is the target for datazone creation.

11b calculate datazone population density homogeneity
This is calculated as the average deviation from the mean population density – and is different to the standard deviation.
11c A list is created of all cluster types within the datazone
11d A list is generated of EAs at datazone edge that are adjacent to at least one datazone with a population > 0 within the same municipality.

11e – for each EA in the list created in **11d, record a list of valid single moves** from *donor* to all *adjacent* datazones ensuring the following criteria are met –

- EAs within both *donor* and *adjacent* datazone remain internally contiguous
- Remaining *donor* and *adjacent* datazones must contain at least one EA i.e. datazones cannot be destroyed
- Population of *donor* and *adjacent* datazones must remain within tolerances. Underpopulated or overpopulated datazones are given further opportunity here to move to within tolerances.
- Cluster group of EA leaving *donor* datazone must exist within *adjacent* recipient datazone
- *Donor* EA area must not deviate by more than 50% from the area of adjacent EA in *adjacent* datazone
- Total area of *adjacent* recipient datazone must not increase by more than 25% if already above the maximum area size, 50 sq km. This allows for a slight increase in geographic area to assist optimising other variables

11f - for each EA in the list created in **11d, record a list of valid two way moves** between *donor* and all *adjacent* datazones. A *donor* datazone provides an EA to the *adjacent* datazone and the *adjacent* datazone returns an EA to the newly constructed *donor* datazone (minus the EA it has just donated). All edge EAs in all *adjacent* datazones to EAs created in **10d** are considered, in turn. Valid moves meet the following criteria –

- EAs within both *donor* and *adjacent* datazone remain internally contiguous
- Remaining *donor* and *adjacent* datazones must contain at least one EA i.e. datazones cannot be destroyed
- Population of *donor* and *adjacent* datazones must remain within tolerances. Underpopulated or overpopulated datazones are given further opportunity here to move to within tolerances.
- Cluster group of EA leaving *donor* datazone exists within *adjacent* recipient datazone. Cluster group of EA returning from *adjacent* datazone to *donor* datazone must exist in *donor* datazone.
- Area of both donor and adjacent EA designated for moving between datazones must not deviate by more than 50% of the area of adjacent EA in *adjacent* datazone to which it is moving

- Total area of *donor* and *adjacent* recipient datazone must not increase by more than 25% if already above the maximum area size, 50 sq km. This allows for a slight increase in geographic area to assist optimising other variables

11g – for each valid single direction move **(11e)** and two way move **(11f)** the following are calculated for the two original and the two amended datazones –

- Population density homogeneity
- Deviation from target population
- Compactness ratio

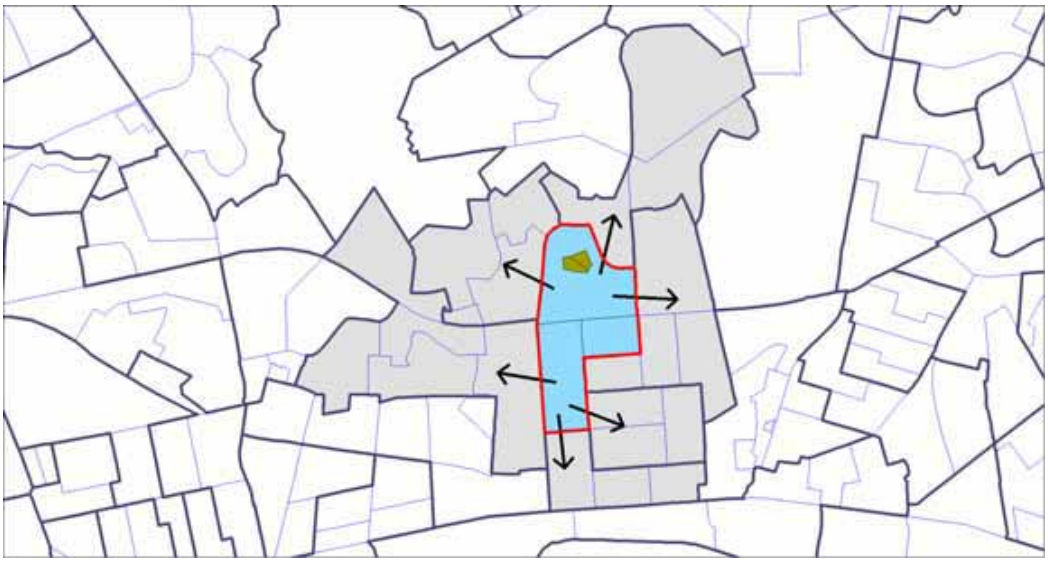
11h – The list of the valid moves from **11e** and **11f** is restricted to those that also meet the following datazone change criteria –

- Increase in population density homogeneity for **each** datazone
- Reduction in **net** population deviation from mean for **both** datazones
- Decrease of circularity of no more than 0.2 (20%) for **each** datazone

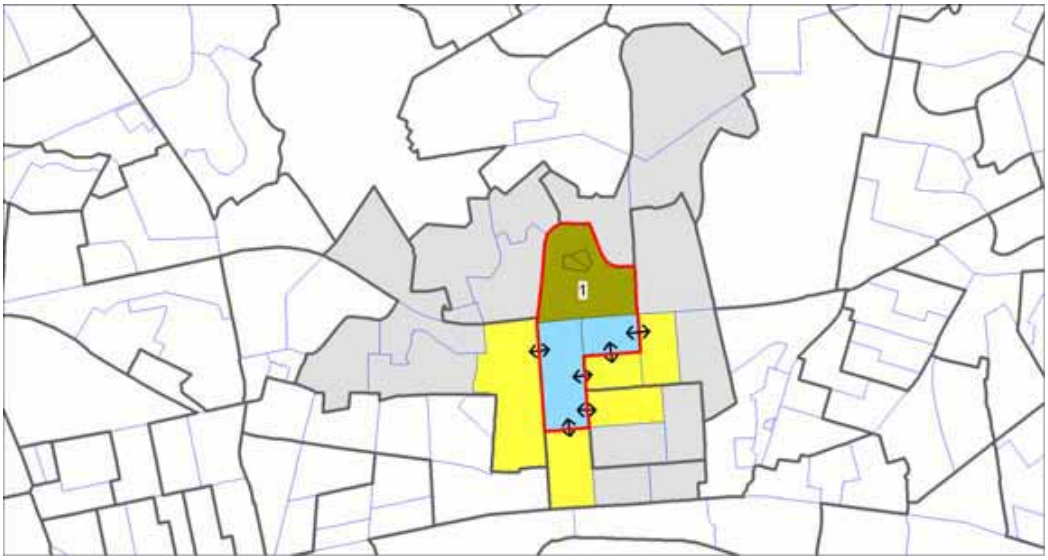
11i- The *single* or *two-way move* meeting criteria of **11h** and with the greatest improvement in net circularity is selected, and implemented.

11j **Repeat 11a to 11i until all datazones have been evaluated once as a donor within this sweep of South Africa**. However, a datazone may be considered many times as a recipient within a single sweep.

11k **Repeat 11a to 11j to further optimise –** using the endpoint of the previous sweep of South Africa as the start point for a new sweep.



Donor datazone outlined red – containing edge EAs (blue) and internal non-valid EAs (dark yellow). Recipient datazones adjacent to edge EAs are highlighted grey.



Dark yellow EA (labeled 1) is deselected as a valid EA to move into an adjacent datazone since it will leave the donor datazone with non-contiguous EAs (the two small EAs above the label). All other donor datazone edge EAs remain valid – as removing these (individually) would maintain donor datazone EA contiguity. Donor datazone population (after losing an EA) is not a consideration at this point as a returning EA from an adjacent datazone may offset population loss. EAs in adjacent datazones that are adjacent to valid EAs in the donor datazone are highlighted yellow. At this stage all EAs marked blue can move into an adjacent datazone and all EAs marked yellow can return to the donor datazone

The following options are evaluated for each valid EA within the donor datazone –

(1) movement into an *adjacent* datazone (all valid *adjacent* datazones are considered) – **a one way move**
(2) movement into an *adjacent* datazone and the return of an EA from the newly constructed *adjacent* datazone back to the *donor* datazone (all valid EAs in *adjacent* datazone are considered, except the EA that has just been received) – **a two way move**

A single or two way move is valid if the population of both datazones remain within tolerances and both datazones maintain internal EA contiguity. Datazones cannot exceed a certain area size threshold and EAs can only move into datazones that contain at least one EA of the same cluster group and also contain EAs of similar size geographic area. Not all the moves (single and two way) identified in the diagram above may be valid for these reasons. When a datazone receives an EA its outer boundary will change, creating a new set of *adjacent* datazone edge EAs to be considered as 'return' EAs in a two way move. This set will exclude the EA that has been absorbed. A datazone may lose an EA bringing its population below the minimum threshold, but receive an EA in return that brings the population back in line with datazone requirements – as long as other criteria are satisfied.

All combinations of moves are evaluated for the *donor* datazone – and valid moves placed into an *optimisation table*. A list of *optimal moves* (single or two way) is generated from this table.

Definition of an optimal move

- Improvement of population density homogeneity in both datazones
- Net reduction in population deviation from the mean for both datazones
- Circularity not decreased by more than 0.2 for each datazone

The move selected is one that meets these criteria and results in the most compact datazone

Optimisation applied

Two iterations were applied across South Africa. The second iteration produced few changes. Optimisation was designed for fine tuning of a thorough initial aggregation process.

- Number of datazones after initial aggregation: 22 846
- Number of datazones after 2 optimisation rounds: 22 846
- EAs shifted during 2 optimisation rounds: 1 366

References

Barnes, H., Noble, M., Wright, G. and Dawes, A. (2009) 'A geographical profile of child deprivation in South Africa', *Child Indicators Research*, 2 (2): 181-199. doi:10.1007/s12187-008-9026-2

Barnes, H., Wright, G., Noble, M. and Dawes, A. (2007) *The South African Index of Multiple Deprivation for Children 2001,* Cape Town: Human Sciences Research Council Press.

Noble, M., Babita, M., Barnes, H., Dibben, C., Magasela, W., Noble, S., Ntshongwana, P., Phillips, H., Rama, S., Roberts, B., Wright, G. and Zungu, S. (2006a) *The Provincial Indices of Multiple Deprivation for South Africa 2001*, Oxford: University of Oxford, UK.

Noble, M., Babita, M., Barnes, H., Dibben, C., Magasela, W., Noble, S., Ntshongwana, P., Phillips, H., Rama, S., Roberts, B., Wright, G. and Zungu, S. (2006b) *The Provincial Indices of Multiple Deprivation for South Africa 2001: Technical Report*, Oxford: University of Oxford, UK.

Noble, M., Barnes, H., Wright, G., McLennan, D., Avenell, D., Whitworth, A., and Roberts, B. (2009b) *The South African Index of Multiple Deprivation 2001 at Datazone Level,* Pretoria: Department of Social Development.

Noble, M., Barnes, H., Wright, G. and Roberts, B. (2009) 'Small area indices of multiple deprivation in South Africa', *Social Indicators Research*. doi: 10.1007/s11205-009-9460-7.

Noble, M., Wright, G., Smith, G.A.N and Dibben, C. (2006), 'Measuring multiple deprivation at the small-area level', *Environment and Planning A* 38 (1): 169-185.

Wright, G., Barnes, H., Noble, M., and Dawes, A. (2009) *The South African Index of Multiple Deprivation for Children 2001 at Datazone Level,* Pretoria: Department of Social Development.

Additional reading

Flowerdew, R., Graham, E. and Feng, Z. (2004) *The Production of an Updated Set of Data Zones to Incorporate 2001 Census Geography and Data,* Report to the Scottish Executive. School of Geography and Geosciences, University of St Andrews, Scotland.

Grobbelaar, N. (2005) 'The development of a Small Area Spatial Layer to serve as the most detailed geographical entity for the dissemination of Census 2001 data', Africa GIS conference, Pretoria, 31 October- 4 November 2005.

Martin, D. (2002) 'Geography for the 2001 Census in England and Wales', *Population Trends,* 108: 7-15.

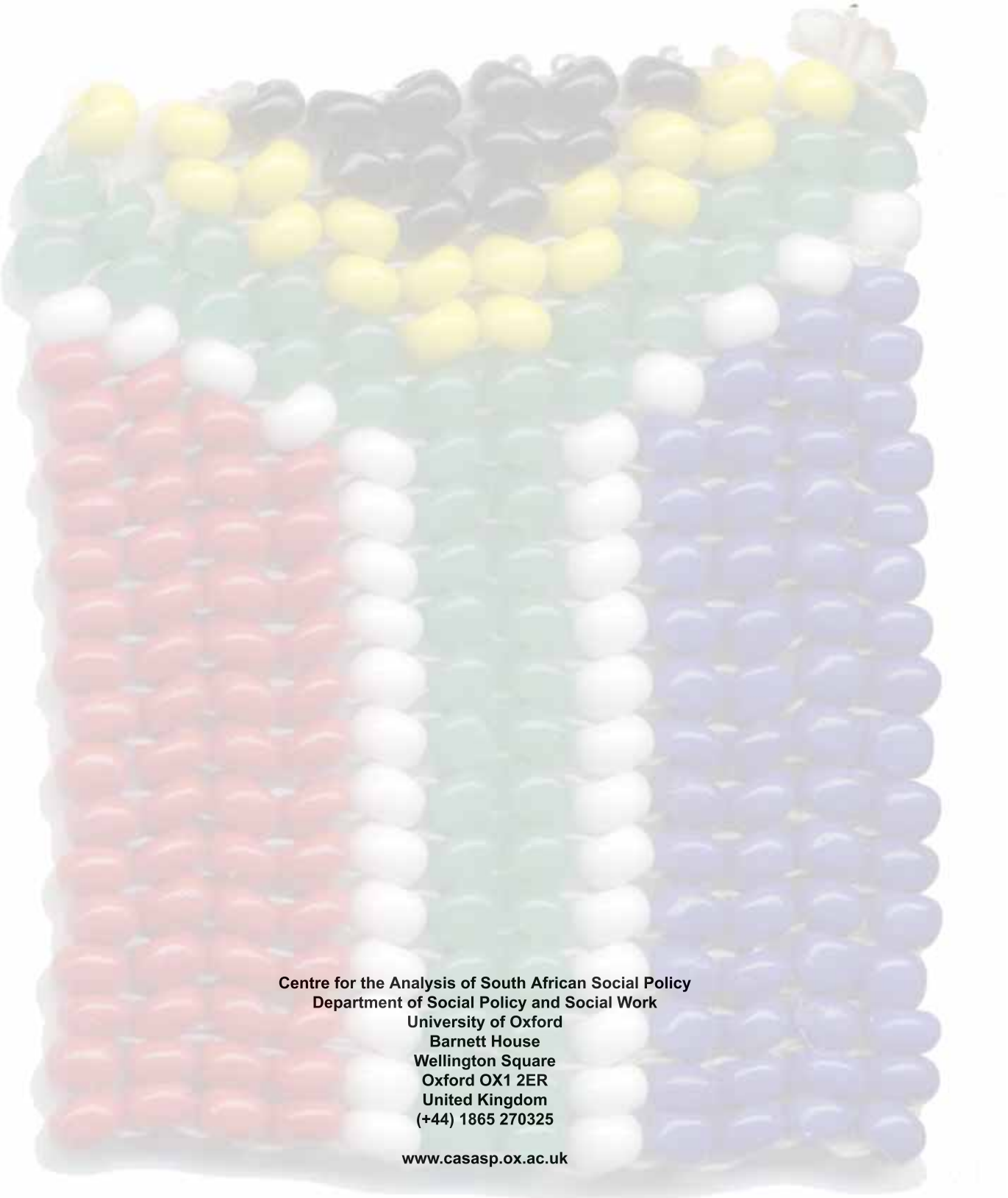
Martin, D., Nolan, A. and Tranmer, N. (2001) 'The application of zone-design methodology in the 2001 UK Census', *Environment and Planning*, 33: 1949-1962.

Martin, D. (2001) 'Developing the automated zoning procedure to reconcile incompatible zoning systems', Proceedings of the 6th International Conference on Geocomputation, University of Queensland, Brisbane, 24-26 September 2001.

Openshaw, S. (1977) 'Algorithm 3: A procedure to generate pseudo-random aggregations of N zones into M zones where M is less than N', *Environment and Planning*, 9: 1423-1428.

Openshaw, S. (1977) 'A geographical solution to scale and aggregation problems in region-building, partioning and spatial modelling', *Transactions of the Institute of British Geographers*, New Series, 2: 459-472.

Openshaw, S. (1995) 'Re-engineering 1991 census geography: serial and parallel algorithms for unconstrained zone design', University of Leeds Geography Department Research Paper, available from: http://www.geog.leeds.ac.uk/papers/95-3/.



Centre for the Analysis of South African Social Policy
Department of Social Policy and Social Work
University of Oxford
Barnett House
Wellington Square
Oxford OX1 2ER
United Kingdom
(+44) 1865 270325

www.casasp.ox.ac.uk