

ATAS DAS I JORNADAS LUSÓFONAS DE CIÊNCIAS E TECNOLOGIAS DE INFORMAÇÃO GEOGRÁFICA

Editores:

José Gomes dos Santos

Cidália Fonte

Rui Ferreira de Figueiredo

Alberto Cardoso

Gil Gonçalves

José Paulo Almeida

Sara Baptista



ARTIGO 48

ALGUMAS TÉCNICAS EM AMBIENTE SIG ÚTEIS À OBTENÇÃO DE ÁREAS DE SERVIÇO DE CONJUNTOS DE PONTOS

TRALHAO, Lino¹; COUTINHO-RODRIGUES, João²; SOUSA, Nuno³

¹ Instituto de Engenharia de Sistemas e Computadores (INESC-Coimbra); Rua Antero Quental 199, 3000-033 Coimbra, Portugal; Tel: +351 239 851040/9; Fax: +351 239 824692; email: linotralhao@netcabo.pt

² Departamento de Engenharia Civil da Faculdade de Ciências e Tecnologia da Universidade de Coimbra Instituto de Engenharia de Sistemas e Computadores (INESC-Coimbra); Departamento de Engenharia Civil - FCTUC; Rua Luís Reis Santos - Pólo II, 3030-788 Coimbra, Portugal; Tel: +351 239 797145; Fax: +351 239 797123; email: coutinho@dec.uc.pt

³ Departamento de Ciências e Tecnologia da Universidade Aberta; Instituto de Engenharia de Sistemas e Computadores (INESC-Coimbra); Delegação de Coimbra da Universidade Aberta; Rua Alexandre Herculano, n.º 52, 3000-019 Coimbra, Portugal; Tel: +351 300 001590; Fax: +351 300 001599; email: nsousa@uab.pt

RESUMO

Em estudos de acessibilidade, e não só, são muito úteis um tipo de estruturas que se podem obter a partir de uma rede, eventualmente multi-modal e parametrizável: as chamadas “áreas de serviço”, as quais são constituídas por polígonos, cada qual correspondente a uma zona situada entre um certo intervalo de custo, relativamente a uma certa “feature” (ponto, multiponto, etc.). Pretende-se neste estudo obter, a partir de áreas de serviço relativas a um universo de *features*, áreas de serviço relativas a subconjuntos dessas *features*. Estas técnicas envolvem manipulações relativamente complexas de polígonos e podem ser generalizadas para conjuntos de conjuntos e assim sucessivamente. Convém notar que nem sempre se dispõe da rede, podendo dispôr-se das referidas estruturas; eventualmente, no caso de áreas de serviço, sob a forma de imagens (raster) a serem convertidas para formato vectorial.

PALAVRAS-CHAVE

Acessibilidade a conjuntos de pontos, Sistemas de informação geográfica, ArcGIS, Áreas de serviço.

SOME GIS ENVIRONMENT TECHNIQUES USEFUL FOR OBTAINING SERVICE AREAS OF SETS OF POINTS

ABSTRACT

In studies of accessibility, but not only, it is useful to consider, among others, a type of structures that can be obtained from a network, eventually multimodal and parameterizable: the so-called “service areas”, which are made out of polygons, each corresponding to a zone between a certain impedance interval, with respect to a certain *feature* (point, multipoint, etc.). This study aims at obtaining, from service areas relative to a universe of *features*, service areas relative to subsets of those *features*. These techniques deal with relatively complex polygon manipulations and can be generalized for sets of sets, and so forth. It is worth noting that the network may not always be available, although the aforementioned structures usually are; eventually, for service areas, under the form of (raster) images, to be converted into vector format.

KEYWORDS

Accessibility to sets of points, Geographical information systems, ArcGIS, Service areas.

1. INTRODUÇÃO

O conceito de acessibilidade, aqui visto como respeitante aos benefícios para a população de aceder a certos locais, também ditos oportunidades, usando um determinado sistema de transporte (DE MONTIS & REGIANNI, 2012), tem sido frequentemente usado em estudos científicos e joga um papel reconhecidamente importante na definição das políticas públicas (GEURS & VAN WEE, 2004; BOCAREJO *et al.*, 2012). Tratando-se de um conceito muito abrangente, este conceito é naturalmente dado a diferentes abordagens, o que de facto se verifica na literatura. Do ponto de vista clássico, a acessibilidade foi definida como “o potencial de oportunidades de interação” (HANSEN, 1959), “a facilidade com que qualquer atividade localizada pode ser alcançada a partir de outras localizações, usando um determinado sistema de transporte” (DALVI & MARTIN, 1976) ou

“os benefícios oferecidos por um sistema de ordenamento de território/ transportes” (BEN- AKIVA & LERMAN, 1979). Outras abordagens foram usadas por diferentes autores (e.g. GEURS & VAN ECK, 2001; LITMAN, 2003; HALDEN *et al.*, 2005). Recentemente tem havido um recrudescer do interesse em análises de acessibilidade e estudos afins (DE MONTIS & REGIANNI, 2012), continuando tais análises a ser centrais na pesquisa sobre assuntos regionais e urbanos (PÁEZ *et al.*, 2012). Como reconhecido por IACONO *et al.* (2010), o melhoramento da acessibilidade, vista como a facilidade em alcançar os destinos procurados, dada uma gama de oportunidades disponíveis e impedâncias (“custos”, e.g. tempo, comprimento do caminho mais curto, etc.) associadas aos recursos usados na deslocação entre origens e destinos, emergiu recentemente como um dos objetivos centrais aos decisores ligados ao planeamento urbano.

Neste cenário, os Sistemas de informação geográfica (SIG) (GEERTMAN & VAN ECK, 1995; LONGLEY *et al.*, 2011; HEYWOOD *et al.*, 2011; MAVOA *et al.*, 2012) aparecem como uma ferramenta natural para executar a tarefa. O SIG permite guardar e representar informação georreferenciada de natureza espacial. Permite também analisar, manipular e operar sobre essa informação. Os ambientes SIG dispõem de “ferramentas” (“tools”) para algumas destas operações, mas é frequentemente necessário estender-se as funcionalidades do SIG desenvolvendo ferramentas analíticas adicionais, destinadas a lidar com problemas específicos. A tecnologia SIG tornou-se assim essencial a estudos de acessibilidade, sendo frequentemente útil no planeamento e gestão da cidade e do território.

É neste contexto que se pretende apresentar algumas técnicas úteis ao estudo da acessibilidade a equipamentos cuja representação em termos de múltiplos pontos pode ser de conveniência. P.ex. se um equipamento tiver dimensões suficientemente grandes, a sua representação por um só ponto pode ser grosseira quando a separação espacial (medida como distância, tempo de viagem ou qualquer outra variável de impedância espacial - ver p.ex. POOLER, 1995) ao ponto representativo mais próximo pode ser considerada ao invés.

Na secção 2 deste trabalho (motivação), apresenta-se o tema do trabalho, na secção 3, duas técnicas para obtenção de áreas de serviço (SA) de conjuntos de pontos representativos, e também uma técnica

para a obtenção de um indicador de acessibilidade. Como exemplo de aplicação destas técnicas, é apresentado, na secção 4, um caso de estudo envolvendo as paragens de 14 linhas de autocarro na cidade de Coimbra, Portugal. Na secção 5 são tecidas algumas considerações de conclusão sobre o trabalho.

2. MOTIVAÇÃO

Antes de mais convém notar que este trabalho foi realizado no contexto do ambiente ArcGIS Desktop 10.1. Aqui se considera a impedância a (ou de) um equipamento representado por um conjunto de pontos, como a menor das impedâncias a (ou de) esses pontos. Dado que se lida neste trabalho com áreas de serviço (ESRI, 2007), estas, quando relativas a um conjunto de pontos assentam na “menor impedância” acima referida.

É sabido que, no contexto da extensão *Network Analyst* (NA), pode obter-se áreas de serviço relativas a estruturas de tipo multiponto. No entanto, estas são obtidas relativamente a um dos elementos do multiponto. O NA permite, contudo, obter áreas de serviço relativas a um conjunto de pontos (aqui designados por conjuntos multiponto). Porém, se se tiver um universo de pontos e se quiser áreas de serviço relativas a uma classe de subconjuntos desse universo, essa obtenção terá de ser feita introduzindo, um a um, e correndo o NA para cada desses subconjuntos o que, tendo em conta o facto de em geral tais corridas serem demoradas, pode tornar o processo muito demorado e também bastante maçador (excepto usando um script ou o *Model Builder*). Além disso, usar esta funcionalidade requer ter-se uma rede onde a supra-citada extensão do ArcGIS irá operar e, mais ainda, tê-la programado com parâmetros adequados, nem sempre conhecidos.

Acontece que nem sempre se dispõe ou da rede ou da possibilidade de a usar convenientemente, p.ex. por não se dispor do próprio NA ou porque a programação/configuração do NA, por muito complexa ou envolvendo certos parâmetros, estará disponível noutra local/entidade. É pois conveniente solicitar a essa entidade áreas de serviço relativas aos vários pontos desse universo e, localmente, e por manipulação desses polígonos, obter SA para subconjuntos desse universo, considerando

inclusivamente diferentes partições deste.

Mas mesmo dispondo-se da rede e do NA devidamente programado/configurado, a obtenção de SA para conjuntos de pontos pode não ser possível, ou, pelo menos, ser muito complexa de obter. É o que acontece, por exemplo quando se trata de uma rede multimodal e pretende-se estudar como se encontram situadas uma ou mais zonas de uma cidade, considerando **simultaneamente**, SA agregadas relativamente a diferentes modos de transporte.

Mais ainda, pode de todo não existir rede, pelo que o NA nem sequer chega a ser útil. Neste caso os polígonos relativos a SA são obtidos por outras vias, eventualmente em formato *raster* (a serem convertidos em formato vetorial) ou mesmo já em formato vetorial. Muitos serão os exemplos que poderão ser fornecidos desde zonas de influência de bases navais ou aéreas, até SA demarcadas no terreno, tendo em conta as suas irregularidades. Seja, por exemplo, uma zona florestal com várias entradas possíveis, e, para cada entrada e para cada tipo de veículo, as zonas de penetração desses veículos, correspondentes a diferentes intervalos de tempo; pretende-se determinar qual ou quais as combinações (subconjuntos) de entradas a serem utilizadas num incêndio.

Um outro problema relacionado com áreas de serviço reside em que não raramente se trata de polígonos definidos por muitos vértices, frequentemente da ordem dos milhões. Assim, a sua obtenção usando NA pode originar “*crashes*” computacionais, em geral resultantes de problemas de memória. Uma forma de tornear esta dificuldade consiste em usar-se uma quadrícula (“*mesh*”). Por métodos que aqui se omitem (mas que o algoritmo C sugere) e com precisão relacionada com o lado das células da *mesh*, podem obter-se áreas de serviço a partir de matrizes origem-destino (OD) em que as origens (destinos) são os pontos de , e os destinos (origens) os centróides das células da *mesh*. É sabido que as heurísticas usadas pelo NA na obtenção das impedâncias elementos das matrizes OD são relativamente mais “ligeiras” do que as relativas a áreas de serviço, logo menos propensas a problemas de memória. No entanto, o uso de matrizes OD no *Network Analyst* não permite o tipo de estudo que as áreas de serviço permitem, porque o que acontece é que cada elemento de uma matriz OD relativa a objectos do tipo multiponto

contêm a impedância a um dos pontos da lista em que internamente o multiponto é representado, e não a impedância ao ponto “mais próximo” de cada multiponto.

3. METODOLOGIA UTILIZADA

Vai pois partir-se do princípio de que se dispõe de uma “*pool*” de áreas de serviço relativas a pontos de um certo conjunto, . Dessa *pool* pode, usando algoritmos que aqui se apresentam, obter-se áreas de serviço relativas a subconjuntos de , aqui designados por “conjuntos multiponto”. Assim, pode obter-se conjuntos de áreas de serviço, para diferentes configurações desses subconjuntos, sem necessidade de recurso sistemático ao *Network Analyst*. De notar que essa *pool* pode ser obtida a partir de imagens (*raster*) e depois, por algum dos modos possíveis, convertida ao formato vectorial.

Antes de prosseguir para a apresentação das técnicas, convém esclarecer alguns conceitos. Neste trabalho o conceito de “*feature class*” (FC) (BUTLER, 2008) é usado no sentido de um conjunto de “*features*”, todas elas com o mesmo tipo de geometria e no mesmo sistema de geo-referenciação. Este é um sentido um pouco mais lato do que o do ArcGIS (ESRI, 2014), incluindo conceitos como p.ex. *shapefile*, “*feature layer*”, “*feature set*”. Apesar de algumas ferramentas requererem um tipo específico de dados de entrada, no geral não é esse o caso, além de que há métodos adequados para a conversão entre tipos de dados. Por outro lado, cada FC tem associada uma tabela (no sentido da Teoria das Bases de dados - um SIG engloba como sua peça chave uma ou mais Base de Dados onde todos os seus objectos se encontram codificados) onde um conjunto de atributos (“*fields*” ou campos) pertinentes aos seus objectos (*features*) são especificados, bem como os valores que cada *feature* possui em cada atributo. Cada coluna da tabela corresponde a um atributo, cada linha corresponde a uma *feature*, encontrando-se nessa linha os valores desse objecto nos vários atributos. Temos pois uma quarta característica de uma FC - os seus objectos estão associados a um conjunto comum de atributos. Ao conjunto de atributos de uma tabela chamamos “*schema*”. Alguns desses atributos são criados pelo próprio ambiente, outros são da

responsabilidade do utilizador o qual dispõe de ferramentas para criar ou eliminar atributos deste segundo tipo.

Outro conceito pertinente é o de junção (Join) que aqui se representa por J_m^t , onde os significados de t e m serão esclarecidos no seguimento. Existem fundamentalmente dois tipos de junções: as de atributo ($t = a$) e as espaciais ($t = s$).

As junções de atributo dizem respeito essencialmente às tabelas associadas às FC operandas e correspondem às operações, com o mesmo nome, da álgebra relacional da Teoria das Bases de Dados. Na junção $x_1 J_m^a x_2$, em que x_1 e x_2 são FC (operandos) e m um campo (ou tuplo de campos) chave (aqui designado por “chave estrangeira”), as *features* do resultado são, tal como acontece na junção espacial, as *features* de x_1 (1º operando); porém a tabela tem por schema a união dos schemas dos operandos, vindo primeiro (colunas da esquerda) os atributos de x_1 e a seguir os de x_2 . Tratando-se de assunto bem conhecido da Álgebra Relacional - Teoria das Bases de Dados relacionais - não será aqui mais desenvolvido. Porém convém referir que o *ArcGis* somente respeita o especificado na Álgebra Relacional para relações do tipo m:1. Para relações m:n, e para cada valor da chave estrangeira escolhe somente a 1ª linha do 2º operando que satisfaz a junção, não procedendo a qualquer replicação de linhas/*features* do 1º operando. Assim, estas junções podem ser “desfeitas”, ficando o 1º operando com a sua tabela “separada” da do 2º. Como o *ArcGis* só permite um campo como chave estrangeira, para junções em que esta constitui um tuplo de campos deve criar-se um campo auxiliar que agregue os valores relativos ao tuplo. P.ex. numa junção (multi-atributo) $m = (keyG, keyQM)$, cria-se em ambas as tabelas dos operandos um campo de tipo string, $keyG_keyQM$, preenchendo-o (*Calculate Field*) usando a expressão $[keyG] \& _ \& [keyQM]$ (caso se use VBA); este campo será a chave estrangeira da junção.

Na junção espacial ($t = s$) (ferramenta *Spatial Join*) não estão em causa quaisquer valores de atributos mas relações de tipo geométrico entre as *features* dos operandos (relações tipo “*within*”, “*inside*”, etc.). Por exemplo na junção $x_1 J_{WITHIN}^s x_2$, *features* de x_1 continuam a estar presentes no resultado; existe também o prolongamento da tabela de x_1 referido acima, só que o prolongamento da linha da tabela relativa a uma *feature* de

x_1 acontece se essa *feature* estiver dentro de uma *feature* de x_2 , sendo prolongada pelos valores nos atributos desta *feature* de x_2 . Aqui pode haver replicação de *features* de x_1 ; é o que acontece quando a relação geométrica em causa é verdadeira para mais do que uma *features* de x_2 (e, além disso, é escolhida uma opção, “*join_one_to_many*”). O resultado é colocado numa 3ª FC. Em anexo apresentam-se as ações executadas por ferramentas relevantes aos algoritmos apresentados.

Dada a sua complexidade, serão aqui explicitadas mais detalhadamente três ferramentas. A 1ª é a ferramenta *Union*, aqui representada pelo símbolo *U*. Para mostrar a sua acção vamos recorrer à Fig 1. Tal como em outras figuras, as “*features*” aparecem desniveladas e em perspectiva somente para melhor explicitação. De facto todas encontram-se no mesmo plano. Usa-se a expressão “em perspectiva”, quando tal desnivelamento é dado.

A figura 1 apresenta, do lado esquerdo, duas FC de polígonos: a FC *galaxies* e a FC *MasterDiss100*. As *features* aparecem separadas em perspectiva. Dando transparência às *features* de modo a só se ver as respectivas fronteiras, temos, do lado esquerdo da figura 1, o mesmo conjunto de *features* visto de cima. Aqui se pode ver que, dado situarem-se no mesmo plano, as *features* das duas FC se retalham reciprocamente de modo a termos um conjunto de “pedaços” (divisões). A ferramenta *Union* tem por resultado uma FC constituída por essas divisões. O *schema* do resultado é essencialmente a união dos *schemas* dos operandos (até ao sistema 9); a partir do sistema 10 existe uma opção relativa ao *schema*, a opção que mais interessa aqui é “*No FiD*”, que corresponde à referida união de “*schemas*”. Vamos supor que é criado, em cada das FC um campo que tem por função constituir uma chave candidata (um chave candidata é um campo cujos valores se distribuem sem ambiguidade pelas *features* de uma FC ou pelas linhas de uma tabela, podendo assim identificá-las univocamente). Chamemos a esse campo *keyG* e *GNr*, respectivamente na FC *galaxies* e na FC *MasterDiss100* (vide figura 1). Cada divisão sobrepõe-se a uma ou mais *features* das FC operandas. Ora, no resultado vão aparecer os campos *keyG* e *GNr*.

Os valores atribuídos a esses campos indicam a *feature* de *galaxies* e a *feature* de *MasterDiss100* a que a divisão se sobrepõe. Se uma divisão se

sobrepuer a mais do que duas *features* do mesmo operando, então essa divisão na realidade corresponde a vários polígonos iguais e sobrepostos, ou seja, o respectivo polígono encontra-se replicado. Assim (vide figura 2), uma divisão que se sobreponha às *features* 1 e 2 de *galaxies* e às *features* 2 e 3 de *MasterDiss100*, corresponde no resultado a 4 polígonos iguais e sobrepostos: $(keyG, GNr) = (1,2), (1,3), (2,2), (2,3)$. Se uma divisão se sobrepuer somente a *features* de um dos operandos, surgirá no resultado tantas vezes quantas essas *features*, mas com valor 0 (ou Null) na chave correspondente ao outro operando. P.ex. uma divisão que se sobreponha às *features* 1 e 3 de mas a nenhuma de *MasterDiss100*, terá duas réplicas: $(keyG, GNr) = (1,0), (3,0)$. Isto significa que podemos encarar essas “sobreposições” como “proveniências”, pelo que do resultado de uma *Union* se podem reconstruir os operandos através do uso da ferramenta *Dissolve* sobre a chave respeitante a esse operando. Na ferramenta *Intersect*, que aqui se representa pelo símbolo \cap , tudo se passa similarmente à *Union*, mas com uma diferença: somente surgem (replicadas ou não) as divisões que se sobrepoem a *features* de ambos os operandos, portanto, nos tuplos acima não haverá zeros.

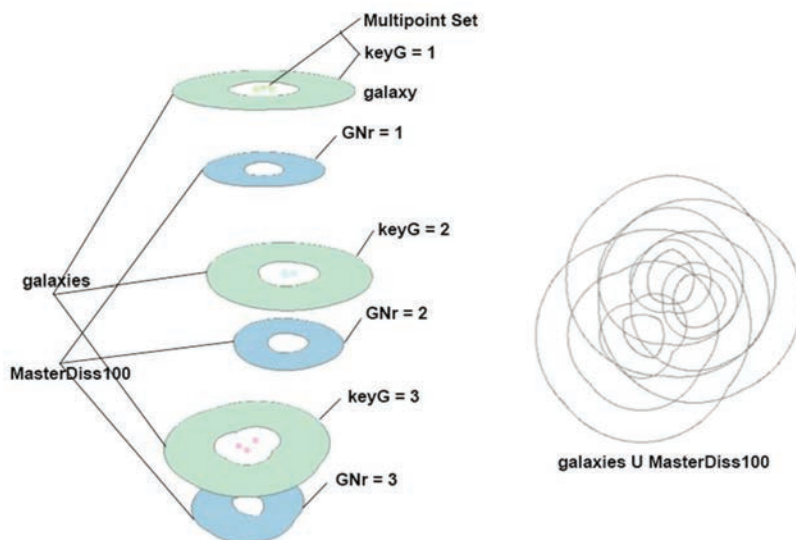


Figura 1 - *Union*

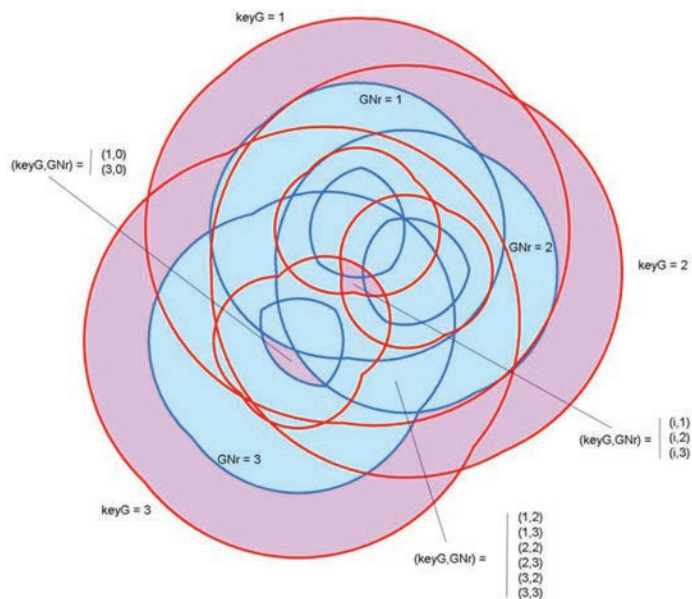


Figura 2 - *Symmetrical Difference e Intersect*

Na ferramenta *Symmetrical Difference*, que aqui se representa pelo símbolo Δ , surgem as divisões da *Union* que não pertencem ao resultado da ferramenta *Intersect*, portanto as divisões que se sobrepõem a *features* de somente um dos operandos. Portanto para todo o par - $(keyG, GNr)$ - um dos componentes é zero. Na figura 2, encontram-se representadas a intersecção e a diferença simétrica. De notar que as *features* da “intersecção” juntamente com as da “diferença simétrica” constituem as *features* da “união”.

Os resultados das três ferramentas acima referidas podem apresentar *features multipart*, i.e. *features* contendo mais do que um elemento geométrico (p. ex. dois polígonos). Na aplicação dos algoritmos abaixo apresentados, tal não constitui problema de maior; porém pode usar-se a ferramenta *Multipart to Singlepart* para decompor essas *features*. É, porém, grave o caso de surgirem divisões com “geometria nula” (polígonos de área nula), ou então polígonos que se auto-intersectam (imagine-se,

p.ex. uma ferradura que é “apertada” de modo à suas pontas se penetra-rem mutuamente, mas mantendo os respectivos bordos). O *ArcGis* não lida bem com objetos destes tipos que considera como contendo erros de geometria. É pois aconselhável o uso da ferramenta *Repair Geometry* após qualquer daquelas operações a qual elimina os objetos de geometria nula e decompõe os que se auto-intersectam.

Uma outra operação que surge nos algoritmos é a atribuição, representada pelo símbolo \leftarrow . Esta operação pode corresponder a uma sequência de operações se, porventura atribuir o resultado de uma operação (p.ex. \cap) a um dos seus operandos. O *ArcGis* não permite tal, salvo algumas exceções. Deverá primeiro colocar-se o resultado da operação (lado direito da atribuição) numa FC auxiliar e depois copiar esta para o destino final (lado esquerdo). Isto pode envolver ainda mais ações, mas um utilizador minimamente familiarizado com este ambiente saberá como proceder.

3.1. Áreas de serviço a um subconjunto de features - algoritmo A

Vamos supor um conjunto P de *features*, as quais, sem perda de generalização, consideramos como sendo pontos. Vamos também supor que temos um conjunto de sequências de k SA, cada qual relativa a um ponto, e que, sem perda de generalização, se trata de áreas de serviço de tipo “ring”, sendo que cada “ring” (ou “anel”) corresponde a um intervalo de impedância (medido na rede, caso esta exista). Temos pois uma sequência $\{I_1, \dots, I_k\}$ de intervalos de impedância adjacentes mas disjuntos. Assim, a cada ponto, j , corresponde uma sequência $\{ring_{I_1}^j, \dots, ring_{I_k}^j\}$ de áreas de serviço. A cada intervalo associamos um número identificador (chave), $keyQM_i$.

Seja *MasterSA* uma FC onde se encontram todas as áreas de serviço correspondentes a todos os pontos de P . Portanto em *MasterSA* estará $Merge(ring_{I_1}^j, \dots, ring_{I_k}^j, \dots, ring_{I_1}^j, \dots, ring_{I_k}^j, \dots)$, referindo a ferramenta *Merge*. Um dos campos da sua tabela associada será o campo $keyQM$. Terá também mais campos: um que identifica o ponto a que cada área de serviço (*ring*) corresponde; dois outros, $minImp(maxIMP)$ que indicam o extremo inferior (superior) do intervalo de impedância correspondente a um *ring*; e eventualmente outros campos com demais informação.

Consideremos agora subconjuntos (conjuntos multiponto) de P , disjuntos entre si, representados por S_{keyG} . O índice *keyG* identifica (chave) o conjunto multiponto, e corresponde em *MasterSA* ao campo *keyG* que indica, para cada SA, qual o conjunto multiponto a que o ponto correspondente a essa SA pertence. Os conjuntos multiponto podem não ser exaustivos: podem corresponder somente a uma parte de P .

Na figura 1 aparecem 9 pontos agrupados em subconjuntos de três. Na figura 3 apresenta-se em perspectiva um desses conjuntos multiponto e áreas de serviço (tomadas como circulares) relativas aos seus pontos.

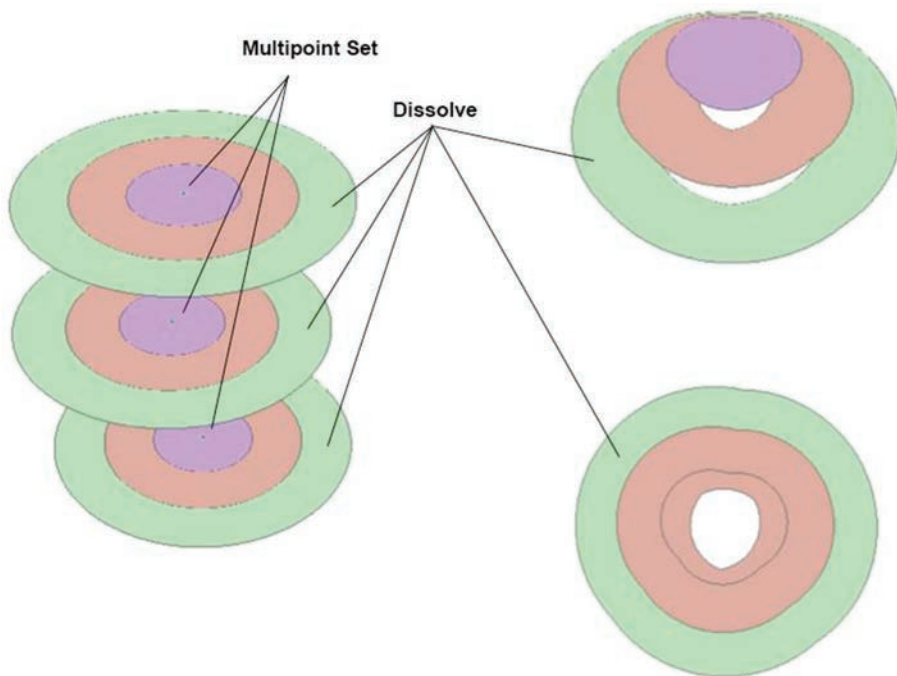


Figura 3 - SA de um subconjunto multiponto

Nesta figura aparece também o resultado da ferramenta *Dissolve* sobre *keyQM*, ou seja as uniões geométricas (“fusões”) das SA referentes ao mesmo intervalo (representadas na mesma cor). Pode verificar-se que os anéis se sobrepõem. Na 3ª parte da figura, agora sem desnivelamentos, mas com a SA referente ao intervalo inferior, transparente mas de modo a ver-se a sua fronteira, tal sobreposição está bem patente. Cada desses

rings corresponde, no algoritmo A, a $ring_{I_i}^{keyG}$ obtido no passo A2.a; no algoritmo B corresponde a um das *features de MasterDiss*.

Posto isto, o algoritmo A permite obter, a partir de *MasterSA*, as áreas de serviço correspondentes aos diferentes multipontos.

Para cada subconjunto multiponto (S_{keyG}):

A1. Determinar, a partir de *MasterSA* e usando a ferramenta *Dissolve*, a união geométrica, *galaxy*, de todas as SA com respeito a todos os pontos de *keyG*; (Nota: a FC *galaxy* é uma FC variável neste algoritmo.)

A2. Para $i = 1$ a $k - 1$:

a. Determinar, a partir de *MasterSA* e usando a ferramenta *Dissolve*, a fusão, $ring_{I_i}^{keyG}$, das SA com respeito aos pontos de *keyG* correspondentes ao intervalo I_i ;

b. Usando a ferramenta *Intersect*, fazer $ring_{I_i}^{keyG} \leftarrow galaxy \cap ring_{I_i}^{keyG}$;

c. Com a ferramenta *Symmetrical Difference*, fazer $galaxy \leftarrow galaxy \Delta ring_{I_i}^{keyG}$;

A3. $ring_{I_k}^{keyG} \leftarrow galaxy$;

Para cada subconjunto multiponto S_{keyG} , as SA são $\{ring_{I_1}^{keyG}, \dots, ring_{I_k}^{keyG}\}$
 (ferramentas *Merge* ou *Append*)

Algoritmo A

Todas as SA obtidas, para todos os intervalos de impedância, e para todos os conjuntos multiponto, podem ser colocadas numa FC resultado, *sa*, no fim do ciclo exterior (usando a ferramenta *Merge*), ou por actualizações sucessivas de (ferramenta *Append*) - vide algoritmo B, passos B5 e B7.

Neste algoritmo, tal como nos seguintes, existem ou podem existir passos intermédios; ou seja, cada “instrução” pode corresponder a mais do que uma operação em ambiente SIG. Em geral trata-se da necessidade de lidar com resultados/passos intermédios, de “arranjos” dos esquemas “*Schemas*” (conjunto de campos - “*field map*”) durante/após a execução de um passo, e também da já referida reparação de geometria.

Na figura 4 apresentam-se os passos A2 a, b e c, para $keyG = 1$, $i = 2$.

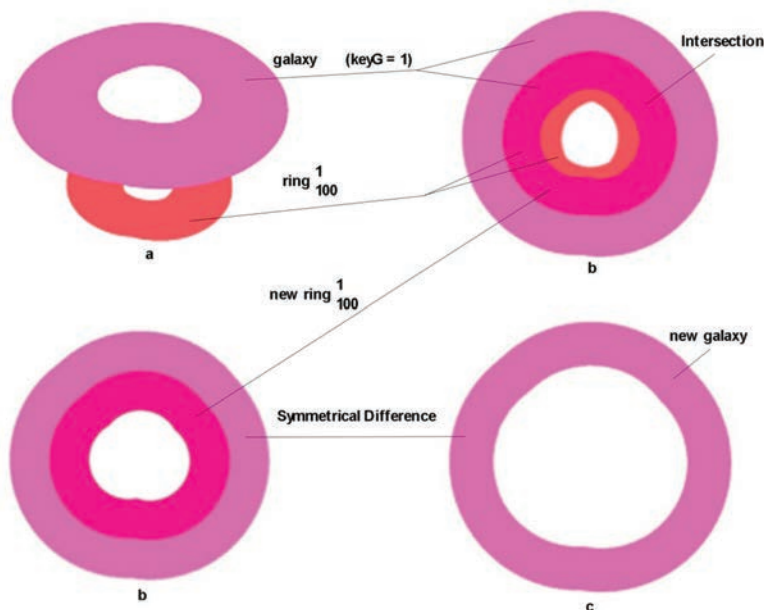


Figura 4 - Passos A2 a, b e c do algoritmo A

3.2. Áreas de serviço de um subconjunto de *features* - algoritmo B

Apresenta-se abaixo um outro algoritmo, **B**, para o mesmo efeito do anterior que, apesar de mais complexo, trata os conjuntos multiponto em simultâneo. No algoritmo B, *galaxies* é uma FC (variável) e *MasterDiss* uma FC em que cada *feature* é a fusão (ferramenta *Dissolve*) das SA das estruturas do estudo, associadas a um subconjunto e a um intervalo de impedância. Inicialmente, uma “*galaxy*” fusão (ferramenta *Dissolve*) de todas as SA correspondentes a um conjunto multiponto e a FC *galaxies* é a coleção dessas fusões. A FC (variável) *rings* corresponde à parte de *MasterDiss* relativa a um intervalo de impedância.

A ideia do algoritmo B é semelhante à do algoritmo A: cada *galaxy* é recortada intervalo a intervalo, removendo-se dela a SA correspondente a esse intervalo (de impedância). Só que isso é feito simultaneamente com todas as “galáxias” e *rings* (relativos a um intervalo de impedância). Temos pois um só ciclo, em vez de dois imbricados.

Seja $I = \{I_1, \dots, I_k\}$ o conjunto de intervalos de impedância, a cada qual corresponde um valor da chave *keyQM*.

B1. Retirar de *MasterSA* as *features* extrínsecas ao estudo.

B2. Usando a ferramenta *Dissolve* sobre *keyQM* e *keyG*, criar a FC *MasterDiss* a partir de *MasterSA*. (Figura 5)

B3. De *MasterDiss* criar, usando a ferramenta *Dissolve* sobre o campo *keyG*, a FC *galaxies*. (Figura 5)

B4. Na FC *MasterDiss*, dar um novo nome ao campo *keyG*, p.ex. *GNr* (tal pode ser feito criando o novo campo, copiando para este os valores do anterior e eliminar *keyB*).

B5. $sa \leftarrow \emptyset$ (ferramentas *Create Feature Class* e/ou *Delete Features*). Na FC *sa* ficará o resultado deste algoritmo.

B6. Para $i = 1, \dots, k - 1$:

- a. Por seleção, colocar os polígonos de *masterDiss* correspondentes a *keyQM_i* em *rings*. [1]
- b. $rings \leftarrow rings \cap galaxies$. [1]
- c. Por seleção, deixar em *rings* apenas os polígonos tais que $GNr = keyG$. Assim, em *rings* ficam as SA (dos vários conjuntos multiponto) correspondentes ao intervalo I_i . Eliminar de *rings* o campo *keyG* (útil mas não necessário). [2] (Figura 6c)
- d. $S_1 \leftarrow rings \cup galaxies$. [3] (Figura 6d)
- e. Por seleção, separar os polígonos de S_1 com $GNr = 0$ (que são por sua vez colocados numa FC, u_0) dos com $GNr > 0$ (que se colocam numa FC, u_1). [4] (Figura 6e)
- f. Usando a ferramenta *Delete Field*, eliminar de u_1 o campo *GNr*. [5]
- g. $inside \leftarrow u_1 \text{ } J_{\text{within}}^s \text{ } rings$ (escolher a opção *join_one_to_many*, e desmarcar a opção *Keep Target Features* – “junção interna”). [6] (Figura 6h)
- h. Remover de u_1 os polígonos correspondentes a *features* de *inside* tais que $keyG = GNr$ e colocar os polígonos restantes numa FC, u_2 . [7] (Figura 6h)
- i. Usando as ferramentas *Append* (ou *Merge*) adicionar à FC *sa* o conjunto de *rings* obtidos no passo B6.c. Simplificar o esquema de *sa* – o campo *keyQM* e um relativo à galáxia (*GNr* ou, preferivelmente, *keyG*) são obrigatórios.
- j. Juntar as *features* u_0 e u_2 (ferramentas *Merge* ou *Append*), fundir (ferramenta *Dissolve*) segundo *keyG* e colocar o resultado em *galaxies*.

B7. No final da iteração $k - 1$, na FC *galaxies* estarão os polígonos correspondentes ao último “ring”, i.e. os polígonos respeitantes ao intervalo I_k . Esta FC deve ser apenas a *sa* (ferramentas *Merge* ou *Append*), mas de modo a que o campo *GNr* passe a chamar-se *keyB*, e o campo *keyQM* seja preenchido na FC *galaxies*. (Figura 7)

B8. Fazer *Dissolve* (com a opção *Multi_Part* ativada) de *sa* sobre os campos relativos ao intervalo de impedância, *keyQM*, e à galáxia, *keyG*. A FC *sa* contém finalmente todas as áreas de serviço correspondentes a todos os conjuntos multiponto e intervalos de impedância. Para obter as SA de qualquer conjunto multiponto, basta usar o *Query Builder* para o selecionar. (Figura 8)

Explicação do Algoritmo:

Ao longo do algoritmo deverá proceder-se à eliminação de eventuais campos espúrios e (não necessário) à transformação Multipart Singlepart. São obrigatórios os campos *keyQM* (em *sa* e *rings*), *GNr* (em *rings*) e *keyG* (em *galaxies* e *sa*).

[0] Na figura 5 mostram-se em perspectiva *galaxies* e *MasterDiss* e respectivas tabelas, juntamente com os conjuntos multiponto. As Figs. 1, 2, 5 e 6 ilustram o algoritmo, e a iteração $i = 2$ do ciclo.

[1], [2] Relembrando a explicação da ferramenta *Intersect* vem que a figura 2 esclarece o passo B6.b. Quanto ao passo B6.c, as *features* para as quais $GNr = keyG$ correspondem a divisões relativas à mesma galáxia (e conjunto multiponto) e portanto temos o equivalente a A2.b do algoritmo A, mas com as (três) intersecções obtidas em simultâneo. (Figura 6c incluindo a tabela de *rings*.)

[3] De notar que neste momento, as *features* de *rings* estão dentro das respectivas galáxias porque as partes que não as intersectavam foram-lhes retiradas.

[4] As divisões de u_0 não se sobrepõem a qualquer das *features* de *rings*, pelo que são agora parte da nova FC *galaxies* (a que será usada na próxima iteração). Quanto aos polígonos de u_1 , alguns farão também parte, outros não. Os últimos devem ser eliminados.

[5] Este passo é recomendado, mas não obrigatório. Destina-se apenas a evitar o aparecimento de um campo, *GNr_1*, no próximo passo.

[6] e [7] É importante salientar que u_2 resulta de *features* de u_1 e a seleção segundo $keyG = GNr$ **diz respeito** a *inside*. Este passo pode ser executado de várias formas, vamos apresentar uma (Fig 6h):

1. Colocar uma chave candidata (*keyy*) e um campo “marcador” (*remove*) em u_1 .
2. Colocar a zeros (ou *false*) todo a coluna relativa a *remove*.
3. Executar B6.g. Em *inside* vão ficar as divisões de u_1 que se encontram

dentro de algum “ring”; em virtude da “completa” repartição produzida por *Union*, não existem em u_1 divisões que estejam parte dentro, parte fora de um *ring* e todas as divisões de *rings* estão replicadas em u_1 .

4. Selecionar em as *features* tais que $keyG = GNr$ (ou seja, “provenientes” de galáxia e de anel relativos o mesmo conjunto multiponto); colocar o resultado em *inside1*.

5. Fazer a junção de atributo $u_1 \overset{a}{j}_{keyy} inside1$.

6. Se se optar por junção interna, colocar a 1 (ou *true*) o campo (exige editor activo); se se optar por junção externa, fazer o mesmo mas só nas linhas com correspondência (sem valores *Null*) nos campos da junção correspondentes a *inside1*.

7. Desfazer a junção e por seleção sobre u_1 colocar em u_2 as *features* de u_1 em que *remove* = 0.

No contexto de um *script* estes sete (sub-)passos são muito mais fáceis de implementar (usando cursores), não sendo necessária a junção de atributo, mas antes um “varrimento” conjunto de u_1 e , ambos ordenados segundo *keyy*.

Imagine-se uma divisão que, independentemente do que se passa quanto a outras galáxias e *rings*, se sobreponha à galáxia 1 e aos *rings* 1 e 2. Esta divisão aparece (pelo menos) duas vezes em u_1 . Sejam 10 e 11 os valores de *keyy* nessas duas réplicas. Então nestas temos $(keyG, GNr, keyy) = (1,1,10), (1,2,11)$. Eliminado o campo *GNr* ficamos com $(keyG, keyy) = (1,10),(1,11)$. Ambas encontram réplicas iguais nos *rings* 1 e 2, pelo que após a junção espacial, essas duas réplicas vão originar em *inside*, quatro réplicas, a saber: $(keyG, GNr, keyy) = (1,1,10), (1,2,10), (1,1,11), (1,2,11)$. Com a seleção $keyG = GNr$ ficam somente duas: $(1,1,10), (1,1,11)$. Repare-se que os dois valores de *keyy*, 10 e 11, aparecem, pelo que, após a junção de atributo ambas as réplicas existentes em u_1 vão ficar “marcadas” (*remove* = 1) para eliminação.

Com raciocínio similar, mas agora com $G_{Nr} \neq 1(keyG \neq G_{Nr})$ nos tripletos de u_1 acima, poderá concluir-se que as réplicas não ficam marcadas para eliminação, ou seja, são transferidas para u_2 . Assim, todas as divisões de u_1 provenientes da galáxia 1, que intersecta *ring* 1 são eliminadas, as que não intersectam *ring* 1 vão para u_2 . Em u_2 e em u_0 estão as divisões da “diferença simétrica” entre a galáxia 1 e o *ring* 1.

Facilmente se verifica que o raciocínio acima é perfeitamente generalizável qualquer que seja o tipo de sobreposição, uma ou mais galáxias um ou mais *rings*. Aplicando o devido formalismo matemático, serve como demonstração.

Por último, convém notar que, tanto o algoritmo A como o B, podem aplicar-se para o caso de *MasterSA* ser constituída não só por SA de pontos isolados mas também (e conjuntamente), ele próprio, por SA de conjuntos multiponto. Para tal basta considerar as SA de cada conjunto multiponto como as SA de um ponto isolado. Imagine-se que se tem já o resultado, , para um certo número de conjuntos multiponto, mas pretende-se acrescentar a um ou mais desses conjuntos mais alguns pontos. Em vez de usar-se novamente o NA (caso seja possível...) para os novos conjuntos multiponto (introduzidos um a um), pode aplicar-se qualquer dos algoritmos para actualizar . No caso da adição de novos pontos dizer respeito a somente um conjunto multiponto, o algoritmo A é claramente mais vantajoso, não só pela sua simplicidade como também porque, neste caso, o ciclo exterior é constituído por uma só iteração. De facto, ao resultado, , basta retirar as SA relativas a esse conjunto multiponto, usar-se o algoritmo para actualizar as SA desse conjunto e actualizar , pela adição das novas SA do conjunto multiponto (ferramentas *Merge* ou *Append*), em substituição das antigas.

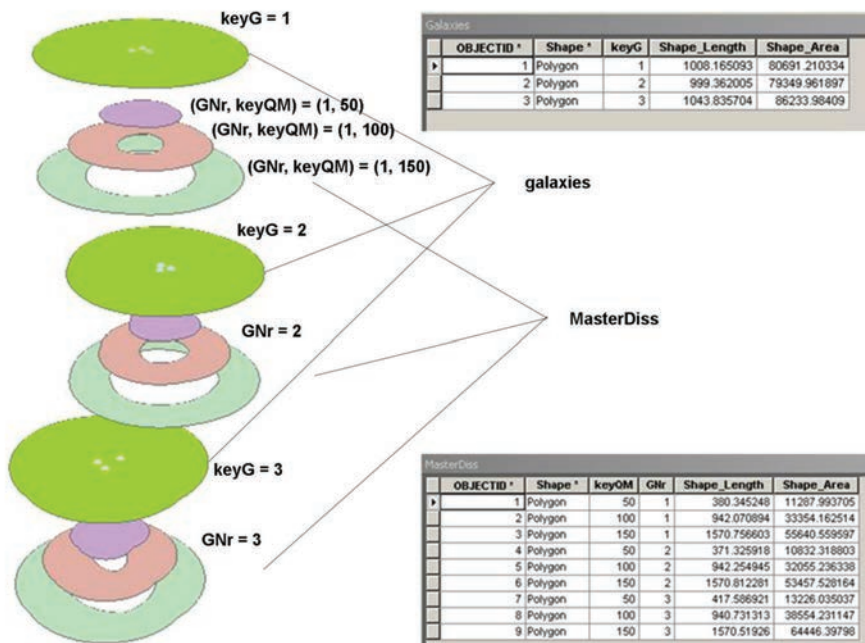


Figura 5 - FC *galaxies* e *MasterDiss* e tabelas associadas

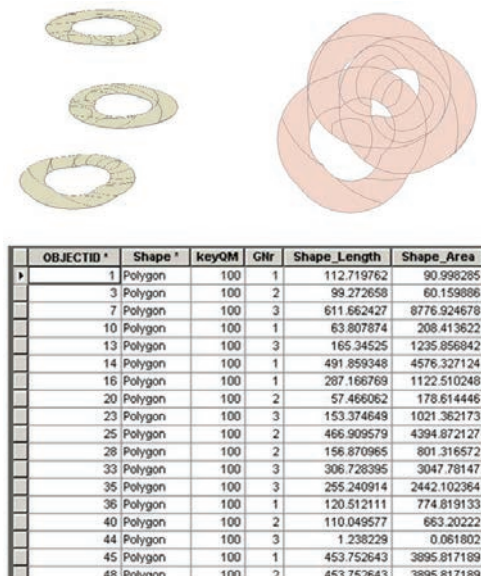


Figura 6c - FC *rings* e tabela associada, após B6.c

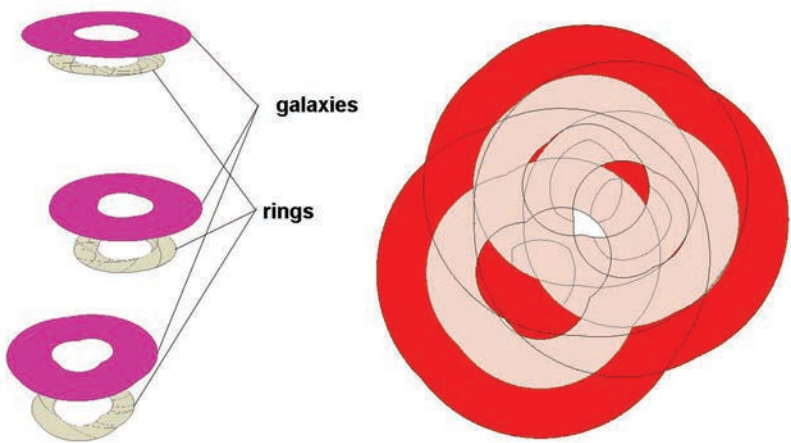


Figura 6d - *galaxies*, *rings* e S_1

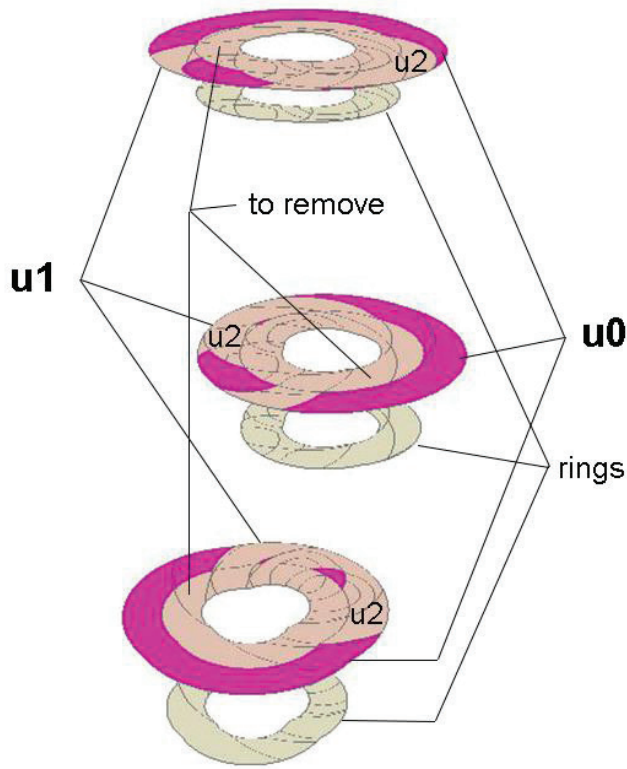


Figura 6e - u_0 , u_1 e *rings*

OBJECTID	Shape *	keyOM	keyC	Shape_Length	Shape_Area	keyy	remove
1 Polygon	100	1	112.719318	90.996207	1	0	
2 Polygon	100	2	99.272659	401.159861	2	0	
3 Polygon	100	3	611.663008	8776.524458	3	0	
4 Polygon	100	1	63.807067	208.413264	4	0	
5 Polygon	100	3	165.543496	1235.691611	5	0	
6 Polygon	100	2	57.466062	176.614446	6	0	
7 Polygon	100	3	153.375112	1021.36246	7	0	
8 Polygon	100	1	120.512111	774.619133	8	0	
9 Polygon	100	2	110.049577	663.20222	9	0	
10 Polygon	100	3	1.236229	0.061002	10	0	
11 Polygon	100	1	491.85935	4576.32713	11	0	
12 Polygon	100	2	491.85935	4576.32713	12	0	
13 Polygon	100	1	207.165963	1122.509579	13	0	
14 Polygon	100	3	207.165963	1122.509579	14	0	
15 Polygon	100	1	466.909579	4394.672127	15	0	
16 Polygon	100	2	466.909579	4394.672127	16	0	
17 Polygon	100	2	156.870665	801.316552	17	0	
18 Polygon	100	3	156.870665	801.316552	18	0	
19 Polygon	100	1	306.726395	3047.76147	19	0	
20 Polygon	100	3	306.726395	3047.76147	20	0	
21 Polygon	100	2	255.240914	2442.102364	21	0	
22 Polygon	100	3	255.240914	2442.102364	22	0	
23 Polygon	100	1	246.415721	1796.072912	23	0	
24 Polygon	100	3	246.415721	1796.072912	24	0	
25 Polygon	100	2	263.691572	1858.041005	25	0	
26 Polygon	100	3	263.691572	1858.041005	26	0	
27 Polygon	100	1	127.174421	826.143767	27	0	
28 Polygon	100	2	127.174421	826.143767	28	0	
29 Polygon	100	2	62.762209	157.419562	29	0	
30 Polygon	100	0	62.762209	157.419562	30	0	
31 Polygon	100	1	112.486703	660.542939	31	0	
32 Polygon	100	2	112.486703	660.542939	32	0	
33 Polygon	100	1	453.752643	3695.617189	33	0	
33 Interiors	100	1	453.752643	3695.617189	33	0	

u₁

inside

OBJECTID	Shape *	keyOM	keyC	keyy	remove	GN	Shape_Length	Shape_Area
31 Polygon	100	1	31	0	2	112.486703	660.542939	
32 Polygon	100	2	32	0	2	112.486703	660.542939	
33 Polygon	100	1	33	0	1	453.752643	3695.617189	
34 Polygon	100	1	33	0	2	453.752643	3695.617189	
35 Polygon	100	2	34	0	1	453.752643	3695.617189	
36 Polygon	100	2	34	0	2	453.752643	3695.617189	
37 Polygon	100	1	36	0	1	453.752643	3695.617189	
38 Polygon	100	1	36	0	2	453.752643	3695.617189	
39 Polygon	100	2	38	0	1	196.000218	1590.97138	
40 Polygon	100	2	38	0	2	196.000218	1590.97138	
41 Polygon	100	1	37	0	1	196.000218	1590.97138	
42 Polygon	100	1	37	0	2	196.000218	1590.97138	
43 Polygon	100	1	38	0	1	196.000218	1590.97138	
44 Polygon	100	1	38	0	2	196.000218	1590.97138	
45 Polygon	100	1	39	0	1	196.000218	1590.97138	
46 Polygon	100	1	39	0	2	196.000218	1590.97138	
47 Polygon	100	3	40	0	1	196.000218	1590.97138	
48 Polygon	100	3	40	0	2	196.000218	1590.97138	
49 Polygon	100	1	41	0	1	321.4407	3946.649621	
50 Polygon	100	2	42	0	1	321.4407	3946.649621	
51 Polygon	100	3	43	0	1	321.4407	3946.649621	
52 Polygon	100	2	44	0	2	168.120704	1511.722066	
53 Polygon	100	2	44	0	3	168.120704	1511.722066	
54 Interiors	100	1	45	0	1	168.120704	1511.722066	

u₁ / a_{key}

inside1

OBJECTID	Shape *	keyOM	keyC	keyy	remove	GN	Shape_Length	Shape_Area			
1 Polygon	100	1	112.719318	90.996207	1	1	100	1	0	1	
4 Polygon	100	1	63.807067	208.413264	4	1	100	1	4	0	1
8 Polygon	100	1	120.552111	774.619133	8	1	100	1	8	0	1
11 Polygon	100	1	491.85935	4576.32713	11	1	100	1	11	0	1
13 Polygon	100	1	207.165963	1122.509579	13	1	100	1	13	0	1
23 Polygon	100	1	246.415721	1796.072912	23	1	100	1	23	0	1
27 Polygon	100	1	127.174421	826.143767	27	1	100	1	27	0	1
33 Polygon	100	1	453.752643	3695.617189	33	1	100	1	33	0	1
37 Polygon	100	1	196.000218	1590.97138	37	1	100	1	37	0	1
41 Polygon	100	1	196.000218	1590.97138	41	1	100	1	41	0	1
49 Polygon	100	1	321.4407	3946.649621	49	1	100	1	49	0	1
59 Polygon	100	1	260.36272	1506.610979	59	1	100	1	59	0	1
69 Polygon	100	1	149.270319	938.934562	69	1	100	1	69	0	1
79 Polygon	100	1	149.270319	938.934562	79	1	100	1	79	0	1
89 Polygon	100	1	418.125151	2500.363038	89	1	100	1	89	0	1
99 Polygon	100	1	418.125151	2500.363038	99	1	100	1	99	0	1
109 Polygon	100	1	215.445616	2539.297786	109	1	100	1	109	0	1
119 Polygon	100	1	215.445616	2539.297786	119	1	100	1	119	0	1
129 Polygon	100	1	194.514515	456.386441	129	1	100	1	129	0	1
139 Polygon	100	1	194.514515	456.386441	139	1	100	1	139	0	1
149 Polygon	100	1	194.514515	456.386441	149	1	100	1	149	0	1
159 Polygon	100	1	99.272659	401.159861	159	1	100	1	159	0	1

Figura 6h - Sequência de tabelas dos (sub-)passos de 6g e 6h (ver explicação)

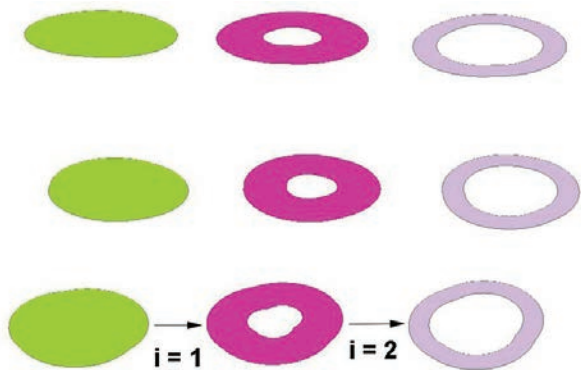


Figura 7 - Evolução da FC *galaxies*

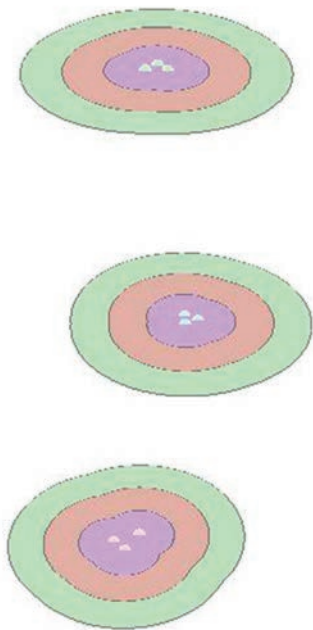


Figura 8 - FC *sa* final

3.3. Um indicador de contorno

Vamos agora supor que temos um conjunto de pontos (“procura” - *demand*), por exemplo os centróides de uma *mesh*. Caso exista, a *mesh* é colocada na FC *MeshPol*. Supõe-se a procura na FC *MeshPt*. Ambas estas FC têm chave candidata *keyB*.

Abaixo apresenta-se um algoritmo para determinar um indicador de contorno: o número de conjuntos multiponto que cada ponto de procura pode alcançar, até um dado valor máximo da impedância, *imp*. Convém que *imp* seja extremo de um dos intervalos de impedância relativos às áreas de serviço. Porém, associando-se a cada área de serviço uma impedância, *keyQM*, o algoritmo também pode ser usado para o mesmo efeito, só que agora com erro dependente das amplitudes dos intervalos de impedância.

-
- C1. Na tabela associada de *MasterSA*, criar e preencher um campo com os valores correspondentes de *keyQM*. Este passo não é estritamente necessário, mas facilita os restantes.
 - C2. Por seleção, ou através do *Query Builder*, retirar de *MasterSA* polígonos correspondentes a impedâncias acima de *imp*.
 - C3. Por junção espacial, criar a FC $S_1 = MeshPt \Join_{within}^s MasterSA$, usando a opção *one-to-many*. Em S_1 cada ponto de procura é então representado por uma pilha de pontos, um para cada polígono (SA) que o seu centróide (ou ponto de procura) intersecta.
 - C4. Usando a ferramenta *Summary Statistics* sobre S_1 , criar uma tabela T_1 , na qual para cada par (*keyG*, *keyB*) está presente o mínimo de *keyQM* (designa-se este campo por *Min_keyQM*).
 - C5. $S_2 \leftarrow S_1 \Join_{(keyG, keyB)}^a T_1$.
 - C6. Remover de S_2 todos os pontos para os quais *keyQM* \neq *Min_keyQM* e colocar o resultado em S_3 .
 - C7. Colocar um campo chave candidata (*key*) em S_3 .
 - C8. Usando *Summary Statistics* criar a tabela T_3 em que para cada valor do par (*keyG*, *keyB*) somente o primeiro valor de *key* aparece (no campo *FIRST_key*).
 - C9. Usando procedimento similar a C5 e C6, agora usando *key* e *FIRST_key* criar a FC S_4 em que, para cada valor do par (*keyG*, *keyB*) somente a primeira *feature* de S_3 aparece. Ficamos pois em S_4 , com a impedância de cada centróide da *mesh* ao ponto “mais próximo” de cada conjunto multiponto.
 - C10. Aplicando a ferramenta *Summary Statistics* (ou *Frequency*) sobre S_4 , criar a tabela T_4 , onde para cada *keyB*, o número de ocorrências de *keyG* é contado e colocado num campo. Cada um desses números corresponde ao número de subconjuntos multiponto à impedância *imp* ou menor do ponto *keyB*.
 - C11. O resultado final, sob a forma de uma FC de polígonos da *mesh* é obtido pela junção *meshPol* $\Join_{keyB}^a T_4$.
-

Algoritmo C

Explicação do Algoritmo:

O tuplo ($keyG$, $keyB$) identifica um par (conjunto multiponto, ponto da *mesh*). C3 replica cada ponto da *mesh* (centróide ou ponto de procura) de forma a ficarem em cada “pilha” tantos pontos quantas as SA o centróide intersecta. C4, C5 e C6 permitem determinar a menor impedância de cada centróide a cada conjunto multiponto, eliminando de S_2 os pontos de cada “pilha” a impedância superior a esse valor mínimo. Para prevenir o caso de um centróide estar à impedância mínima de mais do que um ponto de um conjunto multiponto, os passos C8 e C9 permitem escolher somente um dos pontos da pilha correspondente ao centróide, por sinal o primeiro, e eliminar os outros. Agora trata-se somente de contar quantos pontos restaram em cada “pilha” e obtêm-se o resultado final.

É possível usar uma parte de *MasterSA* em vez de toda essa FC, dependendo do que se pretende. Em particular, pode ser interessante considerar somente alguns subconjuntos de pontos. Note-se que se as áreas de serviço forem obtidas subconjunto a subconjunto, os passos C4 a C8 são desnecessários.

Uma variante curiosa deste algoritmo consiste em, num passo C6' selecionar de S_3 os valores de $keyQM_i$ correspondentes a um intervalo de impedância, ou ao intervalo união de vários intervalos adjacentes. Neste caso teríamos como resultado, e para cada célula da *mesh* o número de conjuntos multipontos situados nesse intervalo (união) de impedâncias.

4. CASO DE ESTUDO

Como aplicação dos algoritmos, apresenta-se um caso de estudo relativo à acessibilidade a algumas (14) das linhas dos Serviços Municipais de Transportes (rodoviários) Urbanos de Coimbra (SMTUC).

Aqui, P é o conjunto das paragens das diversas linhas; a distância temporal pedonal a cada das paragens desempenha o papel de impedância. Como são estas os únicos pontos de acesso às linhas, torna-se claro que uma área de serviço (correspondente a um intervalo de tempo)

relativa a uma linha é a área de serviço do conjunto multiponto constituído pelas suas paragens. Assim, neste caso *keyG* identifica uma linha e *keyQM* um intervalo de tempo. Como cada linha tem várias variantes, consideram-se, para cada linha as paragens de todas as suas variantes. Por aplicação do algoritmo B (ou A), obteve-se SA relativas às diferentes linhas em estudo, destacando-se (*Query Builder*) o resultado para a linha 7, apresentado na figura 9. Nesta figura, os pontos de várias cores correspondem às paragens: uma cor por cada linha (as paragens da linha 7 encontram-se a branco). *keyG* é o código da linha. As zonas sombreadas a cores indicam áreas de serviço para acesso pedonal à paragem mais próxima da Linha 7 (a branco).

Pode parecer estranho considerarem-se valores tão elevados de impedância como 45min. Acontece, porém que a velocidade comercial média apresentada pelos SMTUC para os seus veículos é de aproximadamente 4 vezes o valor mais usual para a velocidade pedonal média. Assim, e para as SA relativas a impedâncias elevadas, o mapa dá também uma ideia do tempo de autocarro até à linha em causa, a qual é, notoriamente, de natureza circular.

Na figura 10 apresenta-se um resultado obtido pelo algoritmo C: o número de linhas de autocarros SMTUC a menos de 5 minutos a pé de cada quadrícula de uma *mesh* com quadrículas de 50m de lado.

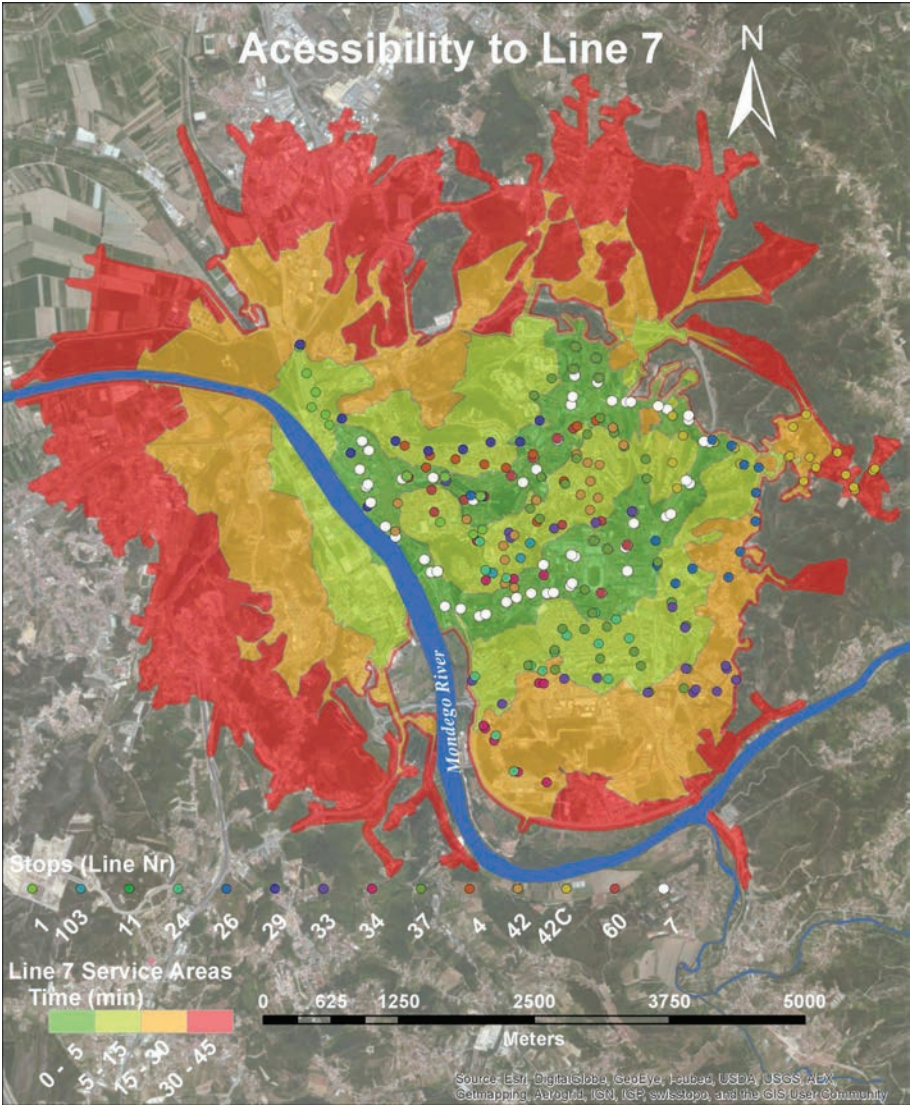


Figura 9 - Acessibilidade pedonal à Linha 7 dos SMTUC

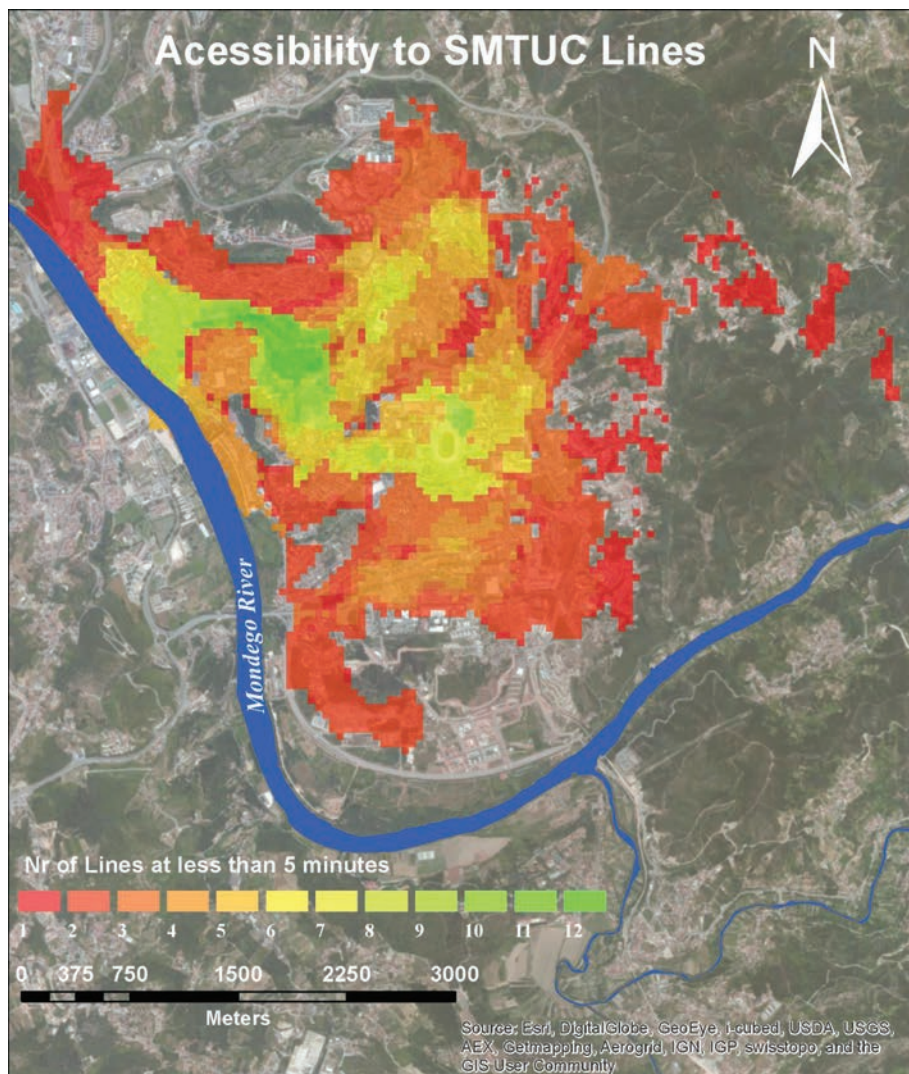


Figura 10 - Número de linhas de autocarros SMTUC a menos de 5 minutos a pé

5. CONCLUSÕES

Neste trabalho foi analisada a questão da acessibilidade a conjuntos de pontos. Apresentaram-se algumas técnicas SIG úteis à abordagem desse tema, baseadas na obtenção de áreas de serviço e manipulação de polígonos. Estas técnicas foram testadas no contexto de um caso de estudo.

O caso de estudo debruçou-se sobre algumas linhas (14) de transportes públicos rodoviários da cidade de Coimbra, Portugal, mas a abordagem é completamente geral e extensível a outros tipos de equipamentos urbanos públicos, comerciais ou de serviços, vistos tanto isolada como agregadamente.

Os resultados permitem identificar deficiências e desequilíbrios oferta/procura, fornecendo ao decisor elementos para o planeamento e gestão da cidade.

AGRADECIMENTOS

Os autores agradecem à Fundação Portuguesa para a Ciência e a Tecnologia” (FCT), que financiou parcialmente este trabalho sob as referências “PEst-OE/ EEI/UI308/2014” e a iniciativa *Energy for Sustainability* da Universidade de Coimbra, suportada pelo projeto *Energy and Mobility for Sustainable Regions* (EMSURE) - CENTRO-07-0224-FEDER-002004.

BIBLIOGRAFIA

- BEN-AKIVA, M. & LERMAN, S.R. (1979) - “Disaggregate Travel and Mobility-Choice Models and Measures of Accessibility.” *in* HENSHER, D.A. & STOPHER, P.R. (eds.): Behavioural Travel Modeling, London: Croom-Helm, 654-679.
- BOCAREJO, S. *et al.* (2012) - “Transport accessibility and social inequities: a tool for identification of mobility needs and evaluation of transport investments.” *Journal of Transport Geography*, Vol. 24, 142-154.
- BUTLER, J.A. (2008) - Designing Geodatabases for Transportation. ESRI Press, New York. 480 p.
- DALVI, M.Q. & MARTIN, K.M. (1976) - “The measurement of accessibility: some preliminary results.” *Transportation*, Vol. 5, 17-42.
- DE MONTIS, A. & REGGIANI, A. (2012) - “JTG Special Section on Accessibility and Socio- Economic Activities: Methodological and Empirical Aspects.” Guest editorial, *Journal of Transport Geography*, Vol. 25, 95-97.
- ESRI (2007) - Working with ArcGIS network analyst. Environmental Systems Research Institute, EUA.
- ESRI (2014) - ArcGIS 2008. Environmental Systems Research Institute, EUA. Disponível online no endereço url: <http://resources.arcgis.com/content/>

- web-based-help (acedido em 30 de março 2014).
- GEERTMAN, S. & VAN ECK, J. (1995) - "GIS and models of accessibility potential: an application in planning." *International Journal of Geographical Information Systems*, Vol. 9, Nº 1, 67-80.
- GEURS, K.T. & VAN ECK, J.T.R. (2001) - Accessibility Measures: review and applications. RIVM Report 408505/006. Utrecht University, Países Baixos.
- GEURS, K.T. & VAN WEE, B. (2004) - "Accessibility evaluation of land-use and transport strategies: review and research directions." *Journal of Transport Geography*, Vol. 12, Nº 2, 127-140.
- HALDEN, D. *et al.* (2005) - "Accessibility analysis literature review." Working paper. Disponível online no endereço url: (acedido em 30 de março 2014) http://home.wmin.ac.uk/transport/download/SAMP_WP3_Accessibility_Modelling.pdf
- HANSEN, W. (1959) - "How Accessibility Shapes Land Use." *Journal of the American Institute of Planners*, Vol. 25, 73-76.
- HEYWOOD, I. *et al.* (2011) - An Introduction to Geographical Information Systems, 4ª Ed. Pearson Education Limited Ed., Harlow, Reino Unido.
- IACONO, M. *et al.* (2010) - "Measuring non-motorized accessibility: issues, alternatives, and execution." *Journal of Transport Geography*, Vol. 18, Nº 1, 133-140.
- LITMAN, T. (2003) - Accessibility: Defining, Evaluating and Improving Accessibility. Victoria Transport Policy Institute. Disponível online no endereço url: www.vtpi.org (acedido em 30 de março 2014).
- LONGLEY, P.A. *et al.* (2011) - *Geographic Information Systems & Science*. Wiley, Hoboken (NJ), EUA.
- MAVOA, S. *et al.* (2012) - "GIS based destination accessibility via public transit and walking in Auckland, New Zealand." *Journal of Transport Geography*, Vol. 20, 15- 22.
- PÁEZ, A. *et al.* (2012) - "Measuring accessibility: positive and normative implementations of various accessibility indicators." *Journal of Transport Geography*, Vol. 25, 141- 153.
- POOLER, J.A. (1995) - "The use of spatial separation in the measurement of transport accessibility." *Transportation Research A*, Vol. 29, Nº 6, 421-427

Série Documentos

Imprensa da Universidade de Coimbra

Coimbra University Press

2015

